IBM

IBM VSE/Advanced Functions

Handbook

Program Number 5666-301

Order Number LY33-9121-0
File No. S370/4300-40

First Edition (July 1985)

This edition applies to Version 2 Release 1 of IBM Virtual Storage Extended/Advanced Functions, Program Number 5666-301, and to all subsequent versions and releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System /370 and 4300 Processor Bibliography, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in in which IBM operates. Any reference to an IBM program product in this document is not intended to state or imply that only IBM's program products may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the address given below; requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comment is provided at the back of this publication. If the form has been removed, comments may be addressed either to:

IBM Corporation
Dept. 6R1B
180 Kost Road
Mechanicsburg, PA 17055, USA

or to:

WT-DP/CE Technical Operations
Dept. 7944
Pascalstr. 100
D-7000 Stuttgart 80, Federal Republic of Germany

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

PREFACE

This is a major revision of and obsoletes LY33-9101-1. The manual is provided as a VSE/Advanced Function, Version 2 Release 1, serviceability aid and is a summary of other VSE/Advanced Functions Version 2 Release 1 documentation.

Note: For reasons of brevity, the product name "VSE/Advanced Functions" is referred to in this publication as VSE/AF.

The volume contains the following information:

Chapter 1: VSE/SP General Information
2: VSE/AF General Information
3: VSE/AF IOCS (General)
4: VSE/AF Supervisor Control Blocks and Areas
5: VSE/AF Service Aids
Appendix

If there is any discrepancy between the information in this manual and the Diagnosis Reference Manuals for the product, the latter is assumed to be correct.

Separate handbooks are available for related program products as follows:

VSE/Power handbook: LY33-9094-3
VSE/ICCF handbook: LY33-9122-0

A handbook sized binder, FE Part Number 453 559, may be purchased from IBM. Customers may order it by their IBM marketing representative. IBM personnel should order it as an FE part from Mechanicsburg.

CONTENTS

CHAPTER 1. VSE/SP GENERAL INFORMATION

GENERAL OBJECTIVES

Package Description

The VSE System IPO/E was used to deliver VSE systems as a package. VSE/SP will be used to deliver VSE Advanced Functions 2.1.x and associated products. It is the successor to VSE System IPO/E 1.4.x and previous VSE/System IPO/E packages. The VSE/SP package is provided to simplify the Installation, Use and Servicing of these products.

VSE System IPO/E 1.3.1

VSE System IPO/E 1.3.1 was delivered as sets of production and service libraries. All Relocatable and Source code delivered in the products was contained in the service libraries while only the subset of that code which was needed by the customer for day to day use was in the production libraries. Service application was accomplished by applying service to the service libraries and then recreating the production libraries from the updated code.

VSE System IPO/E 1.4.x

VSE System IPO/E 1.4.x is delivered as either a Production system or a Generation system. The Generation system is similar to VSE System IPO/E 1.3.1 in both installation and service.

The Production system is also similar to VSE System IPO/E 1.3.1 except that there are no service libraries. Service which involves relocatable and source code not normally required by the customer for day to day use is assembled and link edited by IBM. Service is then delivered to the customer for direct application to his Production libraries.

VSE/SP 2.1.x

VSE/SP 2.1.x is delivered only as a Production system. There is an optional Generation feature which contains source code for VSE Advanced Functions Supervisor and CICS control programs only. All service is delivered in the form normally required by the customer for day to day use. Service which affects code in the Generation feature will be delivered as two PTFs a Production and a Generation Feature PTF. The Generation Feature PTF will only be applied by MSHP if the feature has previously been installed.

Service Process Overview

The service process is activated when a user of IBM code encounters a problem. The user contacts the IBM Support Center with a description and the symptoms of the problem. The Support Center representatives will use this information to perform a search of the data bases containing descriptions and symptoms of known problems.

Known Problems

When a match is found in one of the data bases, the problem is discussed with the user to determine if it is the same problem. When a Support Center Representative and the user agree that this is the same problem and a fix is available, the fix will be made available to the user. The fix may be in the form of a PTF or as an update to an existing Phase, Module, or Macro in the user's system.

When the problem is known but a fix is not available, the customer may request to be added as an interested party to the existing APAR. When a fix is available for the APAR, all interested parties will be notified that the fix is available and it will be delivered to them, upon request, in the correct form for their environment.

New Problems

When no match is found in any of the data bases at the Support Center, an APAR (Authorized Program Analysis Report) is created and sent to the Change Team responsible for the affected component. The Change Team will analyze and correct the error as appropriate and develop a fix. The fix will be available to the user who encountered the problem and to other interested parties, if requested. The problem description and symptoms will then be available in the Support Center data bases to assist other users who encounter the problem.

Note: When a severe problem is encountered for which there is no fix available the Support Center Representative will develop a bypass or circumvention, if requested by the customer.

Service Delivery

Service to resolve a given problem must be tailored by IBM to correspond to each of the environments in which the error exists. For example, 4 PTFs may be required to resolve 1 APAR in CICS 1.6.0., since this product is delivered as part of 3 VSE packages and also as a separately installable product. The format and content of service for each of these environments is different and requires that separate PTFs be provided for each. THE ENVIRONMENT IS DEFINED IN THE SYSTEM HISTORY FILE AND CAN BE OBTAINED BY USING THE MSHP FUNCTION RETRACE PRODUCT. THE SUPPORT CENTER REPRESENTATIVE AND THE USER SHOULD CAREFULLY DEFINE THE ENVIRONMENT IN WHICH THE ERROR OCCURRED IN ORDER TO ENSURE THAT THE CORRECT FIX WILL BE PROVIDED.

Non-Production Format Service

PTFs are applied to the lowest form of the code (Phase, Module, or Macro) in the target system. VSE System IPO/E 1.4.x Generation and previous System IPO/E systems contained modules and macros which could be assembled or link-edited to replace the executable code. PTFs replaced modules or macros which required assembly or link-editing to activate the updated code.

APAR fixes and problem circumventions were also provided as updates to modules or macros with assemblies and/or link edits required to activate the fix.

Production Format Service

VSE System IPO/E 1.4.x Production and VSE/SP 2.1.x systems do not provide modules or macros for service purposes. Service is applied to the level of code which is used for the day to day operation of the system. PTFs for this environment are created by applying the fix and performing assemblies and link-edits as required. The updated Phase, Module, or Macro is then extracted to build a PTF which corresponds to the form of the code in the user system. Whenever possible, the PTF will contain an updated phase with no assembly or link-edit required to activate the fix.

APAR fixes and problem circumventions are also provided as updates to the level of code in the user system. Updates to modules and macros which are available in the IBM Support Center data bases will be retro-fitted by the Component Specialists as needed to correspond to a given system.

Corrections to Phases

In VSE/SP 2.1.x, PDZAP has been superseded by the MSHP CORRECT function. VSE System IPO/E 1.4.x Production and VSE/SP 2.1.x systems consist primarily of executable phases. APAR fixes and bypasses will be provided as corrections to phases when appropriate, to provide relief from severe problems. It is important that information concerning application of the fix is recorded in the system history file to prevent the accidental loss of the fix. MSHP CORRECT provides all the capability of PDZAP with the following additional benefits:

- Automatic recording of the fix.
- System controlled 'Verify' processing.

The VSE/SP 2.1.x Installation Manual also provides examples of using MSHP CORRECT to apply corrections to user programs. This provides the user with a method to control and track all modifications to the system.

Product Refreshes

VSE SP/2.1.x (including the Generation Feature and Optional Programs) is planned to be refreshed periodically at PID and EPL. The refreshed system will include service and Small Programming Enhancements (SPEs) which have been made available since the previous refresh. These refreshes of the system are indicated by the modification level. Each planned refresh will be announced and can be ordered through normal channels.

The primary intent of the refresh is to provide a high level of service for the new VSE/SP 2.1.x user and to provide new function via SPEs, when included in the refresh, to existing users. Refreshes are not intended to be used for corrective service.

PTFs are to be ordered and applied to correct problems. However, there may be exception situations when a customer will desire to bring an installed VSE/SP 2.1.x system to a higher service level. Several things must be reviewed with the customer which could impact the success of the refresh installation.

> There will normally be service available beyond the refresh level. Some of this service could be present (and have to be reinstalled) on the customer's system or required in addition to the refresh. Whenever a refresh is considered, research must be done by the IBM Support Center Representative with the customer to identify this service.
>
> Customer changes to the system can prevent the Interactive Interface produced job-stream for installing a refresh from working correctly. These changes include but are not limited to:
>
> > Changes to the VSE/SP Library structure.
> >
> > Modifications to IBM supplied programming.
> >
> > Additions and/or changes to IJSYSRS.SYSLIB which are not included in PRD2.SAVE.
> >
> > Additions and/or changes to PRD1.BASE.
> >
> > Changes to DASD layouts and/or Volume IDs.
> >
> > Relocation of System History File.
> >
> > Relocation of Job Manager File.

Problem Determination (PD)

Problem Determination is that set of tasks necessary to assess whether a suspected problem is located in hardware or software. Problem Determination is complete when an examination of the available symptoms has isolated the problem to hardware or software.

Problem Source Identification (PSI)

Problem Source Identification begins when Problem Determination has been completed and the problem is suspected to be in software. PSI is an assessment of where the problem resides in the customer's software. That assessment may include examining the available symptoms by using all appropriate expertise and resources (e.g., EWS, documentation, specialists, etc.).

PSI is complete when one of the following has been identified as being the source of the problem:

- IBM System Control Programming,
- IBM Licensed Program,
- Other IBM Program,

- Customer or Non-IBM Program,

  OR

the examination reveals that there was:

- A system operator error,
- An error in the use of the IBM programming or program,
- An error in the use of other programs,
- No trouble found.

Problem Verification, Diagnosis And Resolution

Problem Verification, Diagnosis and Resolution begins when PSI has been completed. It includes the necessary additional effort to decide what further action is required, and taking that action to resolve the problem. The corrective action may include the application of an available IBM fix

Support Structure Tasks

The following sections outline the responsibilities of each level of the support structure that is involved in the service and support of VSE. The intent is not to define all the duties and tasks of each function, but only those directly related to service and support of VSE.

IBM Support Center (Level 1)

Level 1 is the first point of contact for the customer for all problems and service related questions.

Level 1 responsibilities include:

- Checking and verification of customer authorization.
- Opening a Problem Record in RETAIN Problem Management.
- Identifying the customer environment, i.e. SIPO/E 1.3.1, SIPO/E 1.4.x Production, SIPO/E 1.4.x Generation, VSE/SP 2.1.x, or other.
- Assisting the customer with PSI.
- Identifying customer's service level.
- Identifying and gathering problem symptoms.
- Searching the RETAIN Data Base for known problems.
- Providing an existing fix to the customer, i.e. APAR fix or PTF, if available.
- Dispatching Local Support (PSR) if required.
- Passing unresolved problems to the appropriate (Level 2 or Component Specialist) Location

IBM Support Center (Level 2 or Component Specialist)

Handles those problems that are not resolved by Level 1.

Responsibilities include:

- Performing additional RETAIN Data Base searches if needed.
- Performing component level diagnosis to isolate the source of the problem.
- Identifying and recommending application of service which will resolve the problem, if available.

- Providing a fix or circumvention when available, or retro-fitting an existing fix according to the customer's needs and environment.
- Developing a bypass or circumvention in high severity situations, for new problems, if requested by the customer.
- Dispatching Local Support (PSR) if required.
- Assisting the customer in APAR preparation and submission.

Product Change Team

The Product Change Teams provide third level support.

Product Change Team responsibilities include:

- Resolving APARS.
- Providing a fix or PTF.
- Providing assistance to Level 2 when required.

Local Support

Local support is provided by the Program Support Representative.

PSR responsibilities include:

- Assisting the Hardware Customer Engineer (CE) with Problem Determination (PD), upon request.
- Assisting the customer with Problem Source Identification (PSI) and Problem Diagnosis and Resolution, upon request.
- Assisting in the installation of APAR fixes, PTFs and circumventions, upon request.
- Providing assistance to the customer in gathering problem documentation and APAR preparation.

PTF Control (PTFC)

PTF Control is the central collection point for service for all VSE products.

PTF Control responsibilities include:

- Performing a complete visual and automated syntax check of each PTF.
- Testing PTFs for installability and applicability.
- Verifying that post-install steps function correctly.
- Returning PTFs to Product Change Teams if the PTFs fail any of the above testing.
- Shipping PTFs to Programming Service Support (PSS) Mahwah and decided APSGs.

Central Build Group (CBG)

Central Build Group is a new group established for VSE System IPO/E 1.4.x and VSE/SP 2.1.x to create and verify Production Library PTFs from Generation Library PTFs, and to provide support for synchronization of the Production and Generation Library systems. Equivalent Production Library PTFs will be built for each Generation Library PTF.

Central Build Group responsibilities include:

- Utilizing PTFs shipped through PTFC to update the master Production Library system. (Note: If a Product Change Team builds Production Library PTFs, then these PTFs may be shipped directly to PTF Control without CBG intervention.)
- Creating Production Library PTFs.
- Creating Generation Library PTFs for the VSE/SP 2.1.x Generation Feature.
- Performing installation testing of Production Library PTFs.
- Acting as a resource for Level 2 and Product Change Teams in case of emergency fixes.
- Shipping Production Library PTFs to PTFC for entry into the normal PTF distribution process.
- Resolving Production Library PTF build errors.
- Maintaining Production and Generation Library systems for Refresh purposes.

Programming Service Support (PSS) Mahwah

PSS Mahwah receives PTFs from PTFC and integrates them into its PTF Data Base. This Data Base is the source for corrective PTFs in the U.S.

PSS Mahwah responsibilities include building and shipping corrective PTF requests to customers when ordered.

Area Programming Support Group (APSG)

The APSG receives PTFs from PTFC and integrates them into their PTF Data Base. This Data Base is the source for corrective PTFs in EMEA.

APSG responsibilities include building and shipping corrective PTF requests to customers when ordered.

Local Marketing Team

The Marketing Team is generally responsible for ordering IBM products, planning and recommending customer education, assisting the customer in understanding the functions of IBM products, and jointly with the CE representative assisting the customer with installation planning. If the customer requires assistance in these areas the Marketing Team should respond to that need. For example, Refreshes must be ordered by the marketing office when desired by the customer.

VSE/SP Components

The list shows the individual product names, product and component IDs, and in which VSE library each one resides.

Product	PROD ID	COMP ID (CLC)	SUB-LIBRARY
VSE/SP Unique Code 2.1	316A41	5666-31602-A41	IJSYSRS.SYSLIB
VSE/Adv.Func.2.1	301A42	5666-30102-A42 to 108-A42	IJSYSRS.SYSLIB
		5745-SCASM-A42	IJSYSRS.SYSLIB
VSE/Adv. Func.Gen.Feat.	301A43	5666-30106-A43	PRD2.GEN1
VSE/SP NLS f. English 2.1	316A39	5666-31602-A39	IJSYSRS.SYSLIB
DSF1.7	DS2149	5745-SCDSF-149	IJSYSRS.SYSLIB
EREP 3.1	260167	5656-26001-167	PRD1.BASE
OLTEP 1.1	092923	5656-09201-923	PRD.BASE
POWER2.2	273A45	5666-27301-A45	IJSYSRS.SYSLIB
VSAM 1.3	AM2I93	5745-SCAMS-I93 5745-SCVSM-I93 5745-SCVCM-I93	IJSYSRS.SYSLIB
VSAM SPACE MGMT.	AM2I79	5745-SCAMS-I78 5745-SCVSM-I78	IJSYSRS.SYSLIB
VSAM B/R 1.2	AM2I78	5745-SCAMS-I79	IJSYSRS.SYSLIB
ACF/VTAM 2.1	280E27	5666-28001-E27	PRD1.BASE
BTAM-ES 1.1	CG1I08	5745-SCBTM-I08	PRD1.BASE
CICS 1.6	CC3A46	5746-XX300-A46	PRD1.BASE
CICS Gen. Feat.	CC3A47	5746-XX300-A47	PRD2.GEN2
ICCF 2.1	302H02	5666-30201-H02	IJSYSRS.SYSLIB
DITTO 1.1	917I27	5668-91701-I27	PRD1.BASE
FASTCOPY 1.2	AM4F98	5745-AM400-F98	PRD.1BASE

VSE Optional Programs

Certain program products are designated as VSE/SP optional programs, and are packaged and supported for simplified installation. The list is a preliminary list of optional programs. Check with the IBM sales representative for optional programs announced after the appearance of this book.

PRODUCT	PROD ID	COMP ID (CLC)	SUB-LIBRARY
PL/I OPT COMPILER	PL3N74	5736-PL161-N74	PRD2.PROD
PL/I RES LIBRARY 1.6	PL3N72	5736-LM461-N72	PRD2.PROD
PL/I TRANS LIBRARY 1.6	PL3N73	5736-1m561-N73	PRD2.PROD
PL/I Complete (PL3) 1.6	PL3N74	5736-PL161-N74	PRD2.PROD
	PL3N73	5736-LM561-N73	
	PL3N72	5736-LM461-N72	
COBOL 1.3	CB1E44	5746-CB100-E44	PRD2.PROD
	LM4E45	5746-LM400-E45	
RPG II 1.3	RG1042	5746-RG100-042	PRD2.PROD
SORT/MERGE II 2.5	SM2F46	5746-SM200-F46	PRD2.PROD
DMS/CICS/VS 1.4	XC4H31	5746-XC400-H31	PRD2.DBASE
DMS/CICS - IAG FEAT 1.4	XC4H32	5746-XC400-H32	PRD2.DBASE
SQL/DS 1.2	SD1F80	5748-SD1Y0-F80	PRD2.DBASE
DB/DC DATA DICT. 1.4	XXCE43	5746-XXC00-E43	PRD2.DBASE
NCCF 1.2	XX6G44	5745-XX600-G44	PRD2.COMM
NPDA 3.1	295H08	5666-29501-H08	PRD2.COMM
ACF/NCP/vs 3705 2.1	NC2E81	5748-NC216-E81	PRD2.COMM
EP Feat of NCP 3705 2.1	CH1554	5735-SC100-554	PRD2.COMM
SSP 2.1.1	SP2E78	5745-SP215-E78	PRD2.COMM
ACF/NCP/VS 3725 2.1	NC2E77	5748-NC215-E77	PRD2.COMM
	NC2H77	5748-NC215-H77	
EP 3725 1.1	EP1E02	5748-EP115-E02	PRD2.COMM2
	NC2H77	5748-NC215-H77	
DL/I 1.7	XX1H57	5746-XX100-H57	PRD2.DBASE
INFO/SYSTEM 1.1	OZ1H29	5745-OZ135-H29	PRD2.DBASE
SDF/CICS 1.4	XXTF92	5746-XXT00-F92	PRD2.PROD
FTP 2.2	932F86	5668-93201-F86	PRD2.COMM
OCCF 1.2	XC5H03	5746-XC500-H03	PRD2.COMM
Access Control 1.2	XE7H06	5746-XE700-H06	PRD2.COMM

Distribution Tape Formats

Base Tape One

- Device Support Facilities (DSF)
- VSE Standalone utilities
- MSHP History File
- IJSYSR1.SYSLIB (SYSRES library backup)
- Null file
- EOB record

Base Tape Two

- DTSFILE header label
- DTSFILE BACKUP in DTSUIL format
- EOF trailer for DTSFILE
- Null file

- ID and History File
- PRD1.BASE library backup
- Null file
- EOB record
- IESMSG online message file
- Null file

Generation Feature Tape

- HEADER file
- ID and History File
- PRD 2.GEN1 Generation Feature library backup
- Null file
- EOB record

Optional Products Tape (STACKED)

- Null
- Start of stacked tape record
- Null
- Product Header file (1)
- ID and History file (1)
- Libraries (1)
- Null (1)
- EOB record (1)
- Null (1)
- Null
- End of stacked tape record
- Null

Note: (1) These 6 files are repeated for each stacked product.

Disk Volume Requirement for Initial Installation

Disk Type	DOSRES	SYSWK1	SYSWK2	SYSWK3	SYSWK4	SYSWK5	SYSWK6
3310	X	X	X	X	X	X*	—
3330***	X	X	X	X	X	X*	—
3340****	X	X	X	X	X	X*	X**
3350	X	X	—	—	—	—	—
3370	X	X	—	—	—	—	—
3375	X	X	—	—	—	—	—
3380	X	X	—	—	—	—	—

*) Needed only when generation library is used.
**) Needed for ACF/NCP/VS installation.
***) Models 1 and 11. Model 11 is treated as Model 1.
****) Models 70 or 70F or 3344 where one 3344 equals four 3340s.

VSE Libraries

System components are stored in two libraries, IJSYSRS and PRD1. A third library, PRD2, has six predefined sublibraries which are created during initial installation and can be extended dynamically, as it is a VSAM file, to contain more sublibraries with your optional IBM programs and your own application programs.

Library.Sublibrary	Space Managmt.	Created by
IJSYSRS.SYSLIB	non-VSAM	Stand Alone RESTORE
PRD1. BASE	non VSAM	Initial Installation
PRD.2 VSAM	VSAM	Initial Installation
SAVE		" "
PROD		" "
DBASE		" "
COMM		" "
COMM2		" "
GEN1		Interface dialog
any other sublibrary		Interface dialog

The first two libraries contain VSE/SP components in object code format plus modules and source books required for system operation.

IJSYSRS.SYSLIB

The components in this library give hardware and IPL Support.

- VSE/Advanced Functions
- Device Support Facilities
- VSE/POWER
- VSE/VSAM
- VSE/VSAM Space Managements Feature for SAM
- VSE/VSAM Backup Restore Feature
- VSE/ICCF
- VSE/SP 2.1 unique code

PRD1.BASE

The components in this library provide the rest of system functions besides hardware and IPL support.

- ACF/VTAM
- BTAM-ES
- CICS/DOS/VS
- DITTO
- FCOPY
- OLTEP
- EREP

PRD2

PRD2.CONFIG contains user-unique members not required in IJSYSRS:

- Members created during initial installation or by using VSE/SP 2.1 dialogs, for example, CICS tables and VTAM startup books.
- Supervisors and CICS objects generated with the generation feature are also cataloged here. Since PRD2 precedes IJSYSRS in the library search chain, the production system is shielded from any errors from generation.

PRD2.SAVE also contains members unique to your installation, but they are duplicated here from IJSYSRS so that, after installation of a fast service upgrade, they can be copied into IJSYSRS. This duplication allows to apply preventive service without a complete re-installation.

PRD2.PROD,DBASE,COMM,AND COMM2 are default sublibraries for VSE/SP optional programs.

PRD2.GEN1 is the sublibrary for the generation feature. It contains the generation parts of VSE/Advanced Functions and CICS and needs not to reside on disk except during generation tailoring and service application to VSE/Advanced Functions or CICS.

PRD2 is spread over several volumes. The maximum number of library blocks (1K) IBM uses on one PRD2 volume at installation time depends on the disk device according to following list:

3310	23808
3330	23826
3340	23856
3350	112500
3370	113460
3370-2	113460
3375	112500
3380	113925

The rest of the space on each device is empty for additional programs.

You can define or extend non-VSAM libraries with the help of the sample job streams SKLIBDEF or SKLIBEXT in (ICCF) library 59 and VSAM libraries with the dialog "File/Catalog Management".

ICCF (Program Development) Libraries

Library	Contents
1	ICCF administrative library, contents shipped with VSE/ICCF
2	Common library, macros and procedures, VSE/ICCF and VSE/SP 2.1 unique code members.
3–7	Empty
8	Primary library for default operator profile
9	Primary library for default programmer profile
10	VSE/SP Administrative library
11–49	Empty
50–58	Reserved for VSE/SP 2.1
59	VSE/SP Job Streams, Samples, and CICS Tables
60–69	Reserved for VSE/SP 2.1
70–99	Empty

VSE/ICCF libraries are defined as common, public, or private.

Installation

This part of Installation describes the installation tasks for VSE/SP. Installation tasks are varied. Some are performed only once, such as the initial installation of the VSE/SP system. Others may be done periodically. For example, you may decide at a later time to install additional software.

At certain times, it may be necessary to install service. The VSE/SP Interactive Interface provides dialogs to help you with corrective and preventive service. The tasks which apply service to your VSE/SP system are considered installation tasks. They are described in this book.

Initial installation of the VSE/SP system involves many tasks. Some of these are required for all users. Others are done depending on your individual requirements. The tasks which make up VSE/SP initial installation are outlined below.

Initial Installation of the VSE/SP System

This task is required for all users. The subtasks for this installation step are:

- IPL to load Device Support Facilities
- Initialize disks
- IPL to load Standalone Utilities
- Restore SYSRES
- IPL VSE
- Installation job stream processing
- Install VSE/SP 2.1 optional programs
- Complete initial installation

The installation of VSE/SP optional programs during initial installation is optional. They can be installed later using the Interactive Interface.

Change Passwords for VSE/SP User-IDs

VSE/SP ships several user-IDs that are used for various tasks. One special user-ID (POST) and password (BASE) is used for the Complete Initial Installation subtask described above. This ID is defined to do special processing and is reserved for this task only.

The passwords of the VSE/SP user-IDs should be changed. This helps to ensure that unauthorized users do not sign on to the system with these IDs.

Install VSE/SP Generation Feature

This task is optional.

It is done using the Interactive Interface. It helps you install the VSE/SP Generation Feature from tape.

Install Additional Software

This task is optional.

The Interactive Interface provides dialogs to install VSE/SP optional programs and other VSE programs.

The VSE/SP optional programs can also be installed automatically during Initial Installation of the VSE/SP System.

Installation Tailoring

Before your system is fully operational, there are other tasks you may want to do. For example, you might define users to the system or tailor your own IPL procedure.

Sample Disk Layout

DOSRES -------- 3370

Start Block	No of Blocks	File ID
2	55116	VSE.SYSRES.LIBRARY
55118	744	DOS.LABEL.FILEID.AREA1
55862	1488	VSE.POWER.QUEUE.FILE
57350	2232	VSAM.MASTER.CATALOG
59582	62372	VSAM.DATA.SPACE.DOSRES
		%DOS.WORKFILE.SYSLNK
		VSE.CONTROL.FILE
		VSE.TEXT.REPSTORY.FILE
		VSE.MESSAGE.ONLINE
		VSE.ONLINE.PROB.DET.FILE
		%DOS.WORKFILE.SYS001.RECOVER
		CICS.AUTO.STATS.A
		CICS.AUTO.STATS.B
		CICS.TD.INTRA
		DFHTEMP
		CICS.CSD
121954	16864	PAGING.DATA.SET.ONE
138818	63240	VSE.PRD1.LIBRARY
202058	90768	LIBR.DATA.SPACE.DOSRES
		VSE.PRD2.LIBRARY
292826	265112	UNUSED.SPACE
557938	62	VTOC

SYSWK1 -------- 3370

Start Block	No of Blocks	File ID
2	55116	SYS.NEW.RES
55118	3460	WORK.HIST.FILE
58590	136152	LIBR.DATA.SPACE.SYSWK1
194742	37200	VSE.DUMP.LIBRARY
231942	2976	VSESP.USER.CATALOG
234918	67704	VSAM.DATA.SPACE.SYSWK1
		%DOS.WORKFILE.SYS001
		%DOS.WORKFILE.SYS002
		%DOS.WORKFILE.SYS003
		%DOS.WORKFILE.SYS004
		%DOS.WORKFILE.SYS002.RECOVER
		%DOS.WORKFILE.SYS001.SORT
		%WORK.FILE.N11
		TO
		%WORK.FILE.N54
302622	75144	ICCF.LIBRARY
377766	17050	PAGING.DATA.SET.TWO
394816	73656	VSE.POWER.DATA.FILE
468472	2232	VSE.POWER.ACCOUNT.FILE
470704	100	VSESP.JOB.MANAGER.FILE
470804	4994	VSE.HARDCOPY.FILE
475798	752	VSE.RECORDER.FILE
476550	3460	VSE.SYSTEM.HISTORY.FILE
480010	186	INFO.ANALYSIS.DUMP.MGNT.FILE
480196	62	INFO.ANALYSIS.EXT.RTNS.FILE
480258	500	VTAM.TRACE.FILE
480758	1600	CU379X.LOAD.FILE
482358	1000	CU370X.DIAG.FILE
483358	2512	CICS.DUMPA
485870	580	CICS.DUMPB
486450	1000	CICS.AUXTRACE
487450	340	CICS.MSGUSR
487790	70148	UNUSED.SPACE
557938	62	VTOC

DOSRES -------- 3375

Start Track	No of Tracks	File ID
1	1103	VSE.SYSRES.LIBRARY
1104	48	DOS.LABEL.FILEUID.AREA1
1152	24	VSE.POWER.QUEUE.FILE
1176	120	VSAM.MASTER.CATALOG
1296	2664	VSAM.DATA.SPACE.DOSRES
		%DOS.WORKFILE.SYSLNK
		VSE.CONTROL.FILE
		VSE.TEXT.REPSTORY.FILE
		VSE.MESSAGES.ONLINE
		VSE.ONLINE.PROB.DET.FILE
		%DOS.WORKFILE.SYS001.RECOVER
		CICS.AUTO.STATS.A
		CICS.AUTO.STATS.B
		CICS.TD.INTRA
		DFHTEMP
		CICS.CSD
3960	324	PAGING.DATA.SET.ONE
4284	1272	VSE.PRD1.LIBRARY
5556	1800	LIBR.DATA.SPACE.DOSRES
		VSE.PRD2.LIBRARY
7356	4140	UNUSED.SPACE
11496	12	VTOC

SYSWK1 -------- 3375

Start Track	No of Tracks	File ID
1	1103	SYS.NEW.RES
1104	61	WORK.HIST.FILE
1176	2700	LIBR.DATA.SPACE.SYSWK1
3876	744	VSE.DUMP.LIBRARY
4620	168	VSESP.USER.CATALOG
	1104	VSAM.DATA.SPACE.SYSWK1
		%DOS.WORKFILE.SYS001
		%DOS.WORKFILE.SYS003
		%DOS.WORKFILE.SYS004
		%DOS.WORKFILE.SYS002.RECOVER
		%DOS.WORKFILE.SYS001.SORT
		%WORK.FILE.N11
		TO
		%WORK.FILE.N54
5892	2376	ICCF.LIBRARY
8268	336	PAGING.DATA.SET.TWO
860	2316	VSE.POWER.DATA.FILE
1092	72	VSE.POWER.ACCOUNT.FILE
10992	12	VSESP.JOB.MANAGER.FILE
11004	24	VSE.HARDCOPY.FILE
11028	35	VSE.RECORDER.FILE
11036	61	VSE.SYSTEM.HISTORY.FILE
11124	8	INFO.ANALYSIS.DUMP.MGNT.FILE
11132	4	INFO.ANALYSIS.EXT.RTNS.FILE
11136	24	VTAM.TRACE.FILE
11160	48	CU370X.LOAD.FILE
11208	48	CU370X.DIAG.FILE
11256	72	CICS.DUMPA
11328	24	CICS.DUMPB
11352	24	CICS.AUXTRACE
11376	24	CICS.MSGUSR
11400	96	UNUSED.SPACE
11496	12	VTOC

Interactive Interface

This new interface is the base of the VSE/SP 2.1 unique functions. It makes the various functions of the VSE/SP base components as well as additional system services available to all users in a unified and easy to use way. In addition, the interface can be tailored to the personal needs of each individual user. Through single sign on; authorization checking and tailoring of the selection panels, the system can appear to each user as if it were designed especially for him.

The Interactive Interface is a set of function lists, dialogs, selection and data entry panels. It allows the user to utilize the different SP 2.1 component programs (like VSAM/Access Method Services, ICCF and POWER commands etc.), without detailed knowledge of component commands.

You sign on to the Interactive Interface from the 'VSE/SP Online' panel. Enter your user-ID and password. The password is not displayed on the panel.

```
  USER-ID...==>  xxxx        xxxx - user-ID
  PASSWORD...==>  yyyyyy    yyyyyy - password
```

The ID and password are checked by the system. If they are correct, the selection panel or application defined for your user-ID is accessed.

If you get a message informing you that your user-ID or password is incorrect, type in the information again. You may have made a mistake the first time. If it does not work, contact the person responsible for defining user-IDs.

One VSE/SP profile is used for special processing. The other three are predefined to reflect different levels of authorization. The default user-IDs and corresponding passwords are:

User-ID	Password	Function
POST	BASE	Complete initial installation (reserved)
SYSA	SYSA	Default system administrator
PROG	PROG	Default Programmer
OPER	OPER	Default Operator

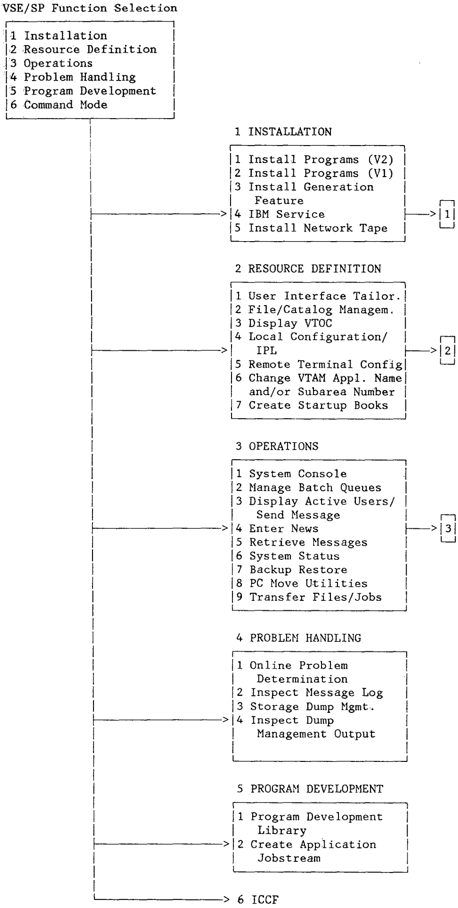
The user-ID 'POST' is a reserved ID. It performs special processing to complete initial VSE/SP installation which is described in 'VSE/SP Installation'. You should only use it for this task. Do not use it to do any other work.

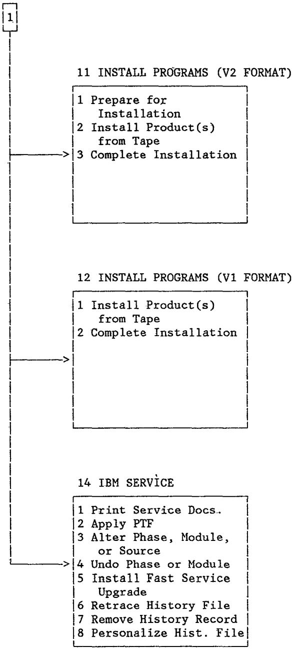
The other three user profiles are default profiles for:

- System administrator
- Programmer
- Operator

You can use them of a day to day basis. You can also use them as models to define other user profiles with specific authorization.

Panel Hierarchy for System Administrator





1

11 INSTALL PROGRAMS (V2 FORMAT)

1 Prepare for Installation
2 Install Product(s) from Tape
3 Complete Installation

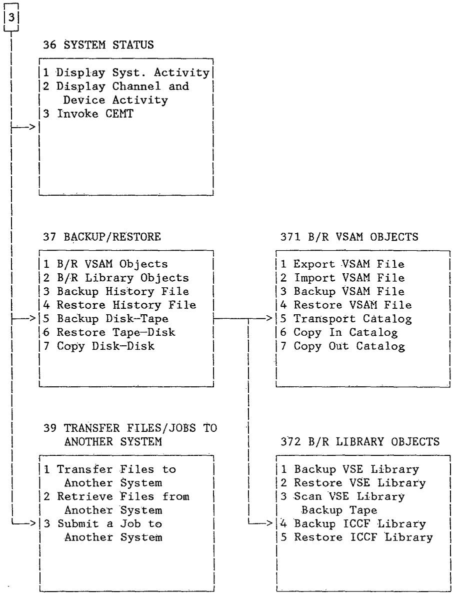
12 INSTALL PROGRAMS (V1 FORMAT)

1 Install Product(s) from Tape
2 Complete Installation

14 IBM SERVICE

1 Print Service Docs.
2 Apply PTF
3 Alter Phase, Module, or Source
4 Undo Phase or Module
5 Install Fast Service Upgrade
6 Retrace History File
7 Remove History Record
8 Personalize Hist. File

```
┌─┐
│2│
└┬┘
 │
 │         21 USER INTERFACE TAILORING
 │        ┌─────────────────────────┐
 │        │1 Maintain User Profiles │
 │        │2 Maintain Selection     │
 │        │   Panels                │
 │        │3 Maintain Application   │
 ├───────>│   Profiles              │
 │        │                         │
 │        │                         │
 │        │                         │
 │        └─────────────────────────┘
 │
 │
 │         24 LOCAL CONFIGURATION AND IPL
 │        ┌─────────────────────────┐
 │        │1 Configure HW Addresses │
 │        │2 Configure Local        │
 │        │   Non-SNA 3270s         │
 │        │3 Configure Local SNA    │
 ├───────>│   3270                  │
 │        │4 Tailor IPL Procedure   │
 │        │5 Maintain Printer FCB   │
 │        │6 Catalog Printer UCB    │
 │        │                         │
 │        └─────────────────────────┘
 │
 │
 │         25 REMOTE TERMINAL CONFIGURATION
 │        ┌─────────────────────────┐
 │        │1 Configure ICA          │
 │        │   connected Terminals   │
 │        │2 Configure NCP          │
 │        │   connected Terminals   │
 └───────>│3 Configure EP           │
          │   connected Terminals   │
          │4 Create Network Tape    │
          │                         │
          │                         │
          └─────────────────────────┘
```



3

36 SYSTEM STATUS

1 Display Syst. Activity
2 Display Channel and Device Activity
3 Invoke CEMT

37 BACKUP/RESTORE

1 B/R VSAM Objects
2 B/R Library Objects
3 Backup History File
4 Restore History File
5 Backup Disk–Tape
6 Restore Tape–Disk
7 Copy Disk–Disk

371 B/R VSAM OBJECTS

1 Export VSAM File
2 Import VSAM File
3 Backup VSAM File
4 Restore VSAM File
5 Transport Catalog
6 Copy In Catalog
7 Copy Out Catalog

39 TRANSFER FILES/JOBS TO ANOTHER SYSTEM

1 Transfer Files to Another System
2 Retrieve Files from Another System
3 Submit a Job to Another System

372 B/R LIBRARY OBJECTS

1 Backup VSE Library
2 Restore VSE Library
3 Scan VSE Library Backup Tape
4 Backup ICCF Library
5 Restore ICCF Library

* Selection Panel

Selection Panel Example: 372 B/R Library Objects

Normal Path:

Select 3 in VSE/SP Function Selection
Select 7 in Operations
Select 2 in Backup/Restore

Fast Path:

Select 372 in VSE/SP Function Selection

Using the System Console

The Interactive Interface has the System Console dialog to help you with console operations. The dialog lets you:

- Display messages currently on the system console.

- Display previous console messages that were written to the VSE/SP hardcopy file.

- Ask for an explanation of VSE/SP messages online. These are the messages which are documented in 'VSE/SP Messages and Codes'. (This facility uses the VSE/SP 2.1 online message file IESMSGS)

- Enter certain console commands.

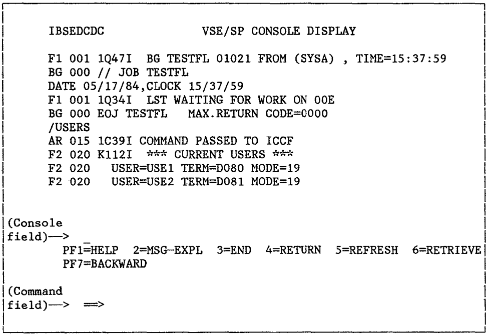
- Reply to console messages.

You can access the dialog as follows:

```
 Administrator         Programmer            Operator
┌──────────────────┐  ┌──────────────────┐  ┌──────────────────┐
│Function Selection│  │Function Selection│  │Function Selection│
│                  │  │                  │  │                  │
│3. Operations     │  │5. Operations     │  │2. System Console │
│                  │  │                  │  │                  │
└──────────────────┘  └──────────────────┘  └──────────────────┘

┌──────────────────┐  ┌──────────────────┐
│Operations        │  │Operations        │
│                  │  │                  │
│1. System Console │  │1. System Console │
│                  │  │                  │
└──────────────────┘  └──────────────────┘
```

The 'VSE/SP Console Display' panel displays current console messages (as displayed at the real system console).

Example Display of System Console Dialog



```
     IBSEDCDC              VSE/SP CONSOLE DISPLAY

     F1 001 1Q47I  BG TESTFL 01021 FROM (SYSA) , TIME=15:37:59
     BG 000 // JOB TESTFL
     DATE 05/17/84,CLOCK 15/37/59
     F1 001 1Q34I  LST WAITING FOR WORK ON 00E
     BG 000 EOJ TESTFL   MAX.RETURN CODE=0000
     /USERS
     AR 015 1C39I COMMAND PASSED TO ICCF
     F2 020 K112I  *** CURRENT USERS ***
     F2 020   USER=USE1 TERM=D080 MODE=19
     F2 020   USER=USE2 TERM=D081 MODE=19


(Console
field)—> _
       PF1=HELP  2=MSG-EXPL  3=END  4=RETURN  5=REFRESH  6=RETRIEVE
       PF7=BACKWARD

(Command
field)—>  ==>
```

Enter Console Commands

You must have authorization to enter console commands. If you have the authorization in your user profile, the console field ('-') is shown at the left of the panel.

Move the cursor to the console field (-), type in a command, and press ENTER. The command is sent to the VSE/SP system. A dialog displays a message which asks you to press PF5 to update the console display.

You can use PF6 to retrieve commands that were previously entered from the dialog.

You cannot use the following commands from the dialog:

```
/CICS  /TC
/DISC  /TTF
HALT   QEND
PEND   Z
```

You cannot use the following commands for the partitions where VSE/ICCF,ACF/VTAM, and VSE/POWER are running:

```
CANCEL
F
PFLUSH
```

The command 'D' to enter or control redisplay mode is sent to the system. However, it is rejected. You can only use it from the real console.

The dialog has information about commands that are frequently used for VSE,VSE/ICCF,VSE/POWER, and ACF/VTAM. You can display this information in one of two ways.

1. Press PF1 for a selection panel. Select one of the options or use PF8 to scroll through the information sequentially.

2. In the command field (==>), type in one of the following, and press PF1. This provides information on the commands for the specific subsystem.

- VSE (For VSE commands)
- ICCF (For VSE/ICCF commands)
- POWER (For VSE/POWER commands)
- VTAM (For ACF/VTAM commands)

Message Explanation

You can request an explanation of a message online from either the VSE/SP Console Display panel or any HELP panel in the dialog. This function accesses the online message file of VSE/SP. There are two ways to request an explanation.

1. Move the cursor to the message line you want explained. Press PF2.

   If a message references another message number, you can display information about this message number. Move the cursor to the message number (in the message line). Press PF2.

2. Type in the message number in the command field (==>). Press PF2.

   You can also display message information about VSE/VSAM return codes. Enter one of the following in the command field (==>). Press PF2.

   ```
   VSAMOPEN
   VSAMCLOSE
   VSAMREQU
   VSAMXXCB
   ```

   Note: If you enter a message number which is incorrect or which cannot be found, the dialog will do one of two things:

   a. Display information about a message number which closely matches the one you entered.

   b. Display a message telling you that the number cannot be found.

   Carefully check the panel display. Make sure that the message number you wanted is the one that the dialog displays.

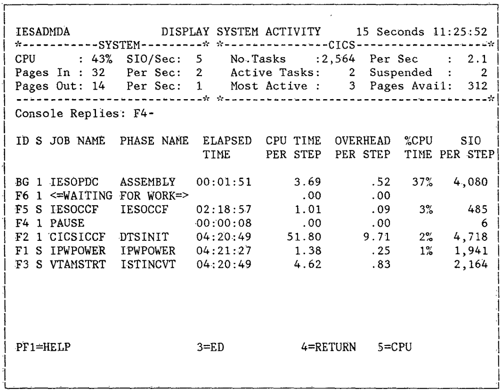
Display System Activity

The 'Display System Activity' dialog displays an interactive picture of the system's activity. You can use the dialog for summary information about the performance of your system. It can highlight areas where performance tools like CICS/PARS or VSE/PT could be used.

You can access the dialog as follows:

Administrator	Programmer	Operator
Function Selection 3. Operations	Function Selection 5. Operations	Function Selection 2. Display System Activity
Operations 6. System Status	Operations 3. Display System Activity	
System Status 1. Display System Activity		

Example Display of System Activity



```
IESADMDA              DISPLAY SYSTEM ACTIVITY     15 Seconds 11:25:52
*-----------SYSTEM---------* *---------------CICS-----------------*
CPU       : 43%  SIO/Sec:  5    No.Tasks     :2,564  Per Sec     :  2.1
Pages In  : 32   Per Sec:  2    Active Tasks:     2  Suspended   :    2
Pages Out: 14    Per Sec:  1    Most Active :     3  Pages Avail:  312
---------------------------* *-------------------------------------*
Console Replies: F4-

ID S JOB NAME  PHASE NAME  ELAPSED   CPU TIME  OVERHEAD  %CPU    SIO
                           TIME      PER STEP  PER STEP  TIME PER STEP

BG 1 IESOPDC   ASSEMBLY   00:01:51      3.69       .52   37%    4,080
F6 1 <=WAITING FOR WORK=>                .00       .00
F5 S IESOCCF   IESOCCF    02:18:57      1.01       .09    3%      485
F4 1 PAUSE                00:00:08       .00       .00              6
F2 1 CICSICCF  DTSINIT    04:20:49     51.80      9.71    2%    4,718
F1 S IPWPOWER  IPWPOWER   04:21:27      1.38       .25    1%    1,941
F3 S VTAMSTRT  ISTINCVT   04:20:49      4.62       .83          2,164




PF1=HELP                  3=ED            4=RETURN    5=CPU
```

SERVICE

Preventive Service - Fast Service Upgrade

The dialog to create the jobstream to install preventive service is invoked by selecting INSTALLATION on the initial system administration panel, IBM SERVICE on the next panel and INSTALL FAST SERVICE UPGRADE on the following panel. The input is required from the user to allow VSE/SP 2.1 to create the jobstream to install the updated system:

- Job submission information
- Tape unit address
- Install updated Generation Feature - Yes/No

Corrective Service

Corrective service for VSE/SP 2.1 is in the form of either PTFs or APAR/Local fixes. They are installed by invoking the appropriate selections on the IBM Service panel.

- Apply PTF
- Alter Phase, Module, or Source
- Undo Phase or module

The following represents the information required to allow the jobstream to be created.

- Job submission information
- Tape unit address
- Backup of affected libraries - Yes/No
- Force indirect service of SYSRES - Yes/No
- PTF numbers to be included or excluded
- Products to be included or excluded
- Components to be included of excluded

PTF Delivery

PTFs for VSE/SP 2.1. are shipped on the new 'SYSIN' format service tape. This tape replaces both, the old PUT tape format (library format PUT tapes) and the cumulative PTF tape format (corrective PTFs).

The format of the tape is as follows:

- File 1: History File or null file
- File 2: Service Document or null file
- File 3: Not used by MSHP
- File 4: EXCLUDE-list or null file
- File 5: PTF Coverletter or null file
- File 6: PTFs
- FILE 7: Not used by MSHP
- File 8: Not used by MSHP

The files which are not used by MSHP are present for compatibility reasons. They allow use of this service tape to service both VSE/SP 2.1 and prior releases of VSE. The files used by MSHP (files 4,5, and 6) during service application have a logical record size of 80 characters and a block size of 10320. File 4 will contain EXCLUDE statements for all PTFs contained on the tape when the tape is intended to be used for corrective service only.

New Service Application Process via SYSIN Tapes.

The user interface to install such a SYSIN tape is significantly simplified via the new command:

```
INSTALL SERVICE.
```

Via the INCLUDE and EXCLUDE detail control statements, the user may selectively include or exclude specific products, components and/or PTFs from the service application process.

Some interesting observations can be made about the new service process:

- It is a one step operation, during which MSHP generates no job streams. The SYSIN PTFs are installed directly.

- MSHP first builds a temporary history file, ensures protection of existing local and APAR fixes as well as MSHP generated members, and then prints a cross reference list of all applicable PTFs and APARs.

- The user is asked to confirm application of the service. Note that the user may include PTFs that were marked as going to be excluded in tape file 4 and that the user may choose a revokable option in order to cause punching of a series of service backout jobs to tape during the service process.

- Now MSHP applies the service directly for multiple products in one step during one pass of the tape file (provided no History file conflicts arise). MSHP no longer requires temporary work libraries or PUT libraries, thus freeing up significant DASD space.

- In the event that the user requires to backout service, then a new command

  ```
  INSTALL BACKOUT
  ```

  will process the backout tape and backout the required service, functioning in the same way as the original INSTALL SERVICE did.

- MSHP provides a new service restart capability whereby MSHP records in the History file when a logical series of PTFs have been applied. Later restart from such a checkpoint can be invoked by either of:

  ```
  INSTALL SERVICE RESTART

  INSTALL BACKOUT RESTART
  ```

  depending on whether service application or service backout was being carried out at the point of interruption.

- MSHP does not apply service directly to an IPL'd SYSLIB sublibrary. The final linkedit step output in fact goes into an internally generated sublibrary IJSYSRS.$$MSHP. When the service is properly complete without errors, then MSHP merges the $$MSHP content into SYSLIB and purges $$MSHP. If any errors occur during the final link step, then the IPL'd SYSLIB is not polluted.

Install IBM Service

Administrator	Programmer	Operator
Function Selection 1. Installation	CANNOT ACCESS	CANNOT ACCESS
Installation 4. IBM Service		

Apply PTF

The Apply PTF dialog applies PTFs from one or more service tapes. You can apply all or selected PTFs. You can also apply PTFs by:

- Library
- Product
- Component

The dialog creates a job sequence which is managed by the Job Manager.

Overview of PTF Application

The following information outlines the process of PTF application.

For all PTFs, the job sequence:

1. Backs up affected sublibraries, if requested.
2. Copies system history file to work history file.
3. Applies service.
4. Handles ICCF members.
   The job asks you to either:
   - Stop VSE/ICCF
   - Disconnect DTSFILE
5. PTF application direct or indirect
   - PTFs applied directly:
     - Service is merged directly into the system
     - The work history file is copied back to the system history file.
     - Service application is complete.
   - PTFs applied indirectly:
     - IJSYSR1 is created to store members.
     - The job sequence instructs you to shut down your system.
     - IPL from IJSYSR1.

Input for all Selections to Apply PTFs

You need the following information for all selections:

- TAPE UNIT ADDRESS
- TAPE QUANTITY
  Specify the number of service tapes you have for the PTFs.
- BACKUP
  Specify whether you want the sublibraries affected by service application backed up.

  1. YES (Backup taken)

  2. NO (Backup is not taken)

- FORCE INDIRECT
  For products in library IJSYSRS, there are two types of PTFs:

  - Direct
    PTFs are directly merged in the running system
  - Indirect
    PTFs are applied to a new SYSRES file.

PTFs for products not residing in library IJSYSRS are applied directly:

The type of application for each PTF on the service tape is predetermined. You can accept the predetermined application or apply all PTFs indirectly.

Alter Phase, Module or Source

Input for all Selections

You need the following information for all three selections which the dialog offers:

APAR/LOCAL NUMBER
APAR number in the format XXYYYYY, where XX is:

- Alphabetic for a formal APAR fix
- The character '#' for a local fix

BACKUP
Specify whether you want the libraries (sublibraries) affected by the service backed up.

1. YES (Back up libraries)
2. NO (Do not back up libraries)

REVOKABLE
Specify whether the fix can be removed at a later time.

1. YES (Fix can be removed)
2. NO (Fix cannot be removed)

The Interactive Interface provides the Undo Phase or Module dialog which removes a fix to a phase or module.

If a fix for a source member is revokable, the dialog punches a job to SYSPCH. The job is used to remove the fix.

TAPE ADDRESS
The address (cuu) of the tape unit used for this task.

AFFECTED COMPONENT
Specify the ID of the affected component.

AFFECTED MEMBER
The phase, module, or macro affected by the fix, depending on whether you are altering a phase, module, or source member.

Corrections to User Phases

The PDZAP program is not included in VSE/AF 2.1. The MSHP CORRECT function can be used to make corrections to user phases. A private History File should be created and a user Product ID and Component ID archived to allow the CORRECT function to modify theses phases.

The following steps should be performed to apply a ZAP to a user program:

- create a private History File (once only),
- archive a User Product ID and Component ID (once only),
- apply a correction to a user phase (mandatory),
- retrace the User Product ID and Component ID (optional).

```
*$$ JOB JNM=CRPHIST,DISP=D,CLASS=0
// JOB CREATE PRIVATE HISTORY FILE
// ASSGN SYS002,DISK,VOL=SYSWK2,SHR
// EXEC MSHP                     /* JOB HAS TO BE EXECUTED ONLY ONCE*/
CREATE HISTORY SYSTEM
  DEFINE HISTORY SYSTEM -
    EXTENT=128:2048 -
    UNIT=SYS002 -
    IDENTIFIER='PRIVATE.HISTORY.FILE'
PERSONALIZE 'M.MORRIS' -
  ADDRESS='60 WATER ST.,NEW YORK,N.Y.' -
  PHONE='633 4537' -
  PROGRAMMER='JOHN' -
  ENVIRONMENT='VSE/SP 2.1'
/*
/&
*$$ EOJ

* $$JOB JNM=ARCHPROD,DISP=D,CLASS=0
// JOB ARCHIVE PRODUCT- AND COMPONENT ID
// ASSGN SYS002,DISK,VOL=SYSWK2,SHR
// EXEC MSHP                     /* JOB HAS TO BE EXECUTED ONLY ONCE */
ARCHIVE ZAPUSE                   /* PRODUCT-ID */
  DEFINE HISTORY SYSTEM -
    EXTENT=128:2048 -
    UNIT=SYS002 -
    IDENTIFIER='PRIVATE.HISTORY.FILE'
COMPRISES 9999-MY-ZAP            /* COMPONENT-ID */
RESOLVES 'USER ZAP HISTORY'
ARCHIVE 9999-MY-ZAP-USE
                                 /* ANY SUBLIBRARY CAN BE SPECIFIED */
                                 /* IN THE RESIDENCE STATEMENT.     */
                                 /* IT MUST NOT BE THE TARGET       */
                                 /* SUBLIBRARY FOR THE ZAP          */
RESIDENCE PRODUCT=ZAPUSE PRODUCTION=ISCLIB.GERRIT
/*
/&
* $$ EOJ

* $$ JOB JNM=APPLZAP,DISP=D,CLASS=0
// JOB CORRECT USER PHASE
// ASSGN SYS002,DISK,VOL=SYSWK2,SHR
// EXEC MSHP                     /* THIS IS THE ACTUAL APPLICATION */
                                 /* OF THE ZAP                     */
                                 /* IN THE RESIDENCE STATEMENT     */
                                 /* THE "REAL" TARGET SUBLIBRARY   */
                                 /* IS SPECIFIED                   */
RESIDENCE PRODUCT=ZAPUSE PRODUCTION=ISCLIB.ULI
  DEFINE HISTORY SYSTEM -
    EXTENT=128:2048
    UNIT=SYS002 -
    IDENTIFIER='PRIVATE.HISTORY.FILE'
CORRECT 9999-MY-ZAP : LF00001
AFFECTS PHASES=USEPHASE
ALTER 100 5BC7:5BC2
RESOLVES 'CORRECTED TYPING ERROR'
/*
/&
* $$ EOJ
* $$ JOB JNM=RETPHIST,DISP=D,CLASS=0
// JOB RETRACE PRIVATE HISTORY FILE
// ASSGN SYS002,DISK,VOL=SYSWK2,SHR
// EXEC MSHP                    /* THIS JOB IS OPTIONAL */
RETRACE PRODUCTS
  DEFINE HISTORY SYSTEM-
    EXTENT=128:2048 -
    UNIT=SYS002 -
    IDENTIFIER='PRIVATE.HISTORY.FILE'
RETRACE COMPONENTS ID=9999-MY-ZAP
/*
/&
* $$ EOJ
```

Install Fast Service Upgrade

You need the following information:

TAPE UNIT ADDRESS
The address (cuu) of the tape unit used for the FSU.

NAME OF ASIPROC
Specify the name of the JCL procedures you will use to IPL from IJSYSR1. The name must begin with $$.

GEN-LIB INSTALL
If you have the VSE/SP Generation Feature installed, specify whether you want to reinstall the generation library during Fast Service Upgrade.

1. YES (Reinstall)
2. NO (Do not reinstall)

If you enter 2 (NO), the entry for the generation library is removed from the system history file. The sublibrary where the feature resides (PRD2.GEN1) is reinitialized. If you do not reinstall the Generation Feature, your system will then be a production system, rather than a generation system.

BACKUP LIBRARIES
Specify whether you want libraries IJSYSRS.SYSLIB and PRD1.BASE backed up.

1. YES (Back up libraries)
2. NO (Do not back up libraries)

The dialog creates a job with the default name DTRFSU. On the 'Job Disposition' panel, you can submit the job to batch, file it in your default primary library, or both.

Additional Considerations

1. The job sequence runs in the background partition (BG) you should start BG with CLASS=0.

2. Only one job sequence from this dialog can be in the reader queue at one time.

3. Start each partition with class equal to the partition number.

   Start F1 with CLASS=1
   Start F2 with CLASS=2
   ...
   Start FX with CLASS=X

   You can start a partition with more than one class. For example if you usually start F4 with CLASS=I, start it with classes I and 4.

4. Fast Service Upgrade checks where CICS/DOS/VS and ACF/VTAM are running (F1 - FB). It start CICS/ICCF and ACF/VTAM (for ACF/VTAM users) in the same partitions in which they were running.

5. You should have a current backup of your system available. You may need this if service application is not successful.

   If you do not have one, specify 1(YES) in the "BACKUP" field.

6. VSE/SP skeletons in library 59 are replaced. If you changed skeletons, you should have copied them to another ICCF library before modifying them.

7. VSE/ICCF members in libraries 1 an 2 are replaced. If you modified any members, save them in another library. For example, you may have changed the SUBMIT procedure. VSE/ICCF Installation and Operations Reference, SC33-6203 lists the ICCF members which may be replaced.

8. All members which you catalog to IJSYSRS should also be cataloged to PRD2.SAVE, so they are saved.

   For example, your $xJCL procedures should also be saved in PRD2.SAVE.

9. If you use a $ASIPROC.PROC member for IPL, save a copy in PRD2.SAVE.

10. The POWER queues must be reformatted. Your POWER startup must either have a 'first time switch' like the one shipped or must reformat the queues. It is recommended that you use a 'first time switch'.

11. Do not have your own information in the following libraries. These libraries are replaced.

    - IJSYSRS
    - PRD2.GEN1

12. If the Generation Feature is installed:

    - It must reside in library PRD2.GEN1.

    - The generation library must be online. If it is not, the job fails. You would then have to exit from Job Manager processing, restore the library and resume the Job Manager.

13. After Fast Service Upgrade, regenerate any CICS/DOS/VS tables you have modified.

    - Terminal Control Table (TCT)
      Use the 'Create Startup Books' dialog. Select the TCT. The dialog is described in VSE/SP System Use.

    - User modified tables
      Submit your own jobs for any tables you use.

14. It is recommended that you reinstall your VSE/SP optional programs and any additional VSE programs after you complete FSU. This ensures that there are matching service levels for all system components.

Overview of Fast Service Upgrade Processing

The following information outlines Fast Service Upgrade. There are two stages: Stage 1 and Stage 2.

Stage 1

In Stage 1, the job sequence:

1. Backs up libraries and history file, if specified.

2. Backs up DTSFILE
   The job asks you to either

- Stop VSE/ICCF
- Disconnect DTSFILE

Before you continue, check the list output to make sure the backup is complete.

3. Copies system history file to work history file.
4. Installs FSU for SYSRES to SYS.NEW.RES.
   MSHP lists down-level service information. After FSU, reapply any service you had installed which is missing in the FSU.
5. Copies members and procedures to IJSYSR1.SYSLIB.
6. Reinstalls Generation Feature, if specified. This is done only if the feature is already installed.
7. Stores IJSYSR1 IPL procedure and jobs needed for Stage 2.

At this point, there is a user exit. You should do the following:

- Regenerate any SYSRES phases and relink them to IJSYSR1 (for example, supervisor or POWER phase).
- Make sure that your LIBDEF chains point to IJSYSR1 instead of IJSYSRS. This is needed to:
  - Use the new code.
  - Catalog to IJSYSR1.
- Shut down all partitions, except the POWER partition.
- Empty the POWER queues.
  - Mount tape.
  - Back up queues.
  - Delete jobs in reader, list, and punch queues.
- Shut down your system.
- IPL from IJSYSR1. Use your IPL procedure with the JCL procedure specified in the dialog.

  To reformat the POWER queues, reply DELETE to message '4733D'.

Stage 2

In Stage 2, the job sequence:

1. Selectively restore FSU DTSUTIL file.
2. Installs FSU for PRD1.BASE.
   MSHP lists down-level service information. After FSU, reapply any service you had installed which is missing in the FSU.
3. Restores message files.
4. Backs up and restores DTSFILE.
   This reorganizes the DTSFILE for better ICCF performance.
5. Copies IJSYSR1 to IJSYSRS and renames system procedures.
6. Updates work history file residencies.
7. Copies work history file to system history file.

   Note: Code replacement is completed. The job sequence now processes startup information.
8. Generates 'base' startup for CICS/DOS/VS.

Note: You can regenerate your own CICS/DOS/VS tables and options later. During this step, you have to respond to certain messages. You need to define the following:

- TP access method: VTAM or BTAM
- BTAM-ES users must define up to three 3270 terminal addresses (cuu).
- VTAM users must specify whether the local control unit is a SNA control unit (YES/NO).

  If you reply YES, define:

  - Control unit address (cuu)
  - Terminal type (327x-y).
  - Up to three local ACF/VTAM terminals (port number).

  If you reply NO, define up to three local ACF/VTAM terminals (cuu).

9. Starts CICS/ICCF and ACF/VTAM (for ACF/VTAM users).

10. Completes cleanup.

    You can now generate your own CICS/DOS/VS tables and options. When you are done, do the following:

    - Load your POWER queue files.
    - Shut down the system.
    - IPL form IJSYSRS.

Problem Handling

The dialog creates a job sequence which runs under the control of the Job Manager.

If a problem occurs and you cannot resume, delete the following jobs in the VSE/POWER reader queue:

- DTRSTFSU
- DTRCLFSU
- All jobs beginning with the prefix DTRFSU

If you do not do this, the Job Manager may not work correctly at a later time.

If you cannot resume , there are other things you should also do. This depends on whether you exit during Stage 1 or Stage 2 of the FSU process.

- Stage 1

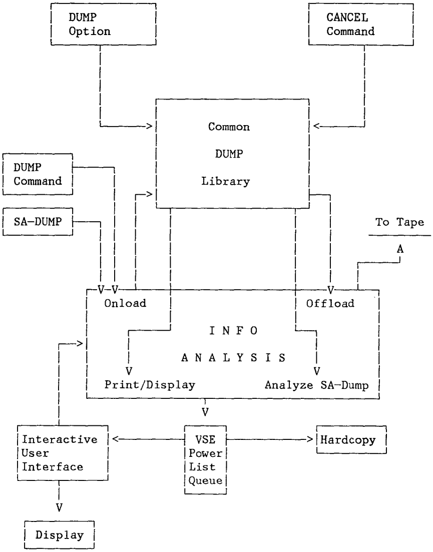
  Your system has not been changed up to the installation of the Generation Feature (for Generation Feature users only). If you exit either during or after this job step, you must restore your old generation library, if it was online.

- Stage 2

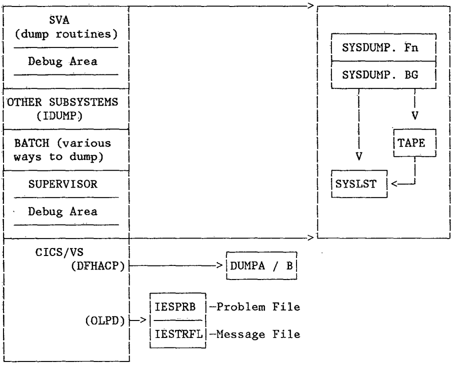
  If you exit and cannot resume during Stage 2, do the following:

  - Shut down your system
  - IPL from IJSYSRS
  - Restore the following:
    - Old DTSFILE
    - Old PRD1.BASE library
    - Old generation library

Dump Processing in VSE/SP 2.1



Summary of Dump Facilities



DUMP MEDIUM SPECIFICATION

- o STDOPT SYSDUMP=YES|NO,DUMP=PART|YES
- o OPTION DUMP|PARTDUMP,SYSDUMP|NOSYSDUMP
- o LIBDEF DUMP,CATALOG=SYSDUMP.BG|Fn

DUMP ANALYSIS TOOLS

- o INFO/ANALYSIS (batch and interactive)
- o VSE/SP Online Problem Determination

DUMP INVOCATION

- o DUMP Command
- o CANCEL Command
- o CANCEL macro
- o IPL Standalone Dump
- o SVC 50
- o PDUMP|DUMP|JDUMP macro
- o IDUMP macro
- o CICS/VS ACP Service
- o Superv. Cancel Service

Handling of Dumps via II

The II uses the VSE/SP function STORAGE DUMP MANAGEMENT to display a list of dumps either in the Dump Library or offloaded to tape. The functions selected from the Dump Management Panel calls the program Info/Analysis (INFOANA) using it's batch functions. The output of INFOANA will be put into the POWER LIST Queue, from where it may be displayed using the INSPECT DUMP MANAGEMENT OUTPUT function.

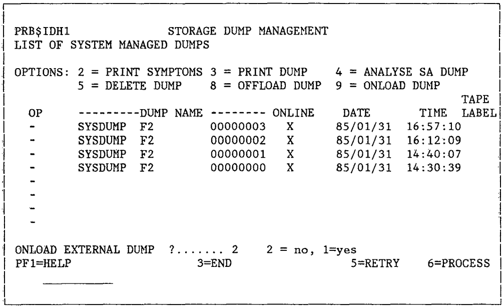
The following functions are provided to handle a dump via II:

- List dump names
- Delete a dump
- Select a dump for processing
- Off load a dump to tape
- On load a dump from tape
- Display of the Symptom String
- Display of formatted control blocks

- Display of unformatted dump data
- Search for data string
- Search for an address
- Scrolling through the dump
- Standalone dump analysis

The Storage Dump Management menu provides a list of system managed dumps.

Example of Storage Dump Management Panel



```
PRB$IDH1              STORAGE DUMP MANAGEMENT
LIST OF SYSTEM MANAGED DUMPS

OPTIONS: 2 = PRINT SYMPTOMS 3 = PRINT DUMP     4 = ANALYSE SA DUMP
         5 = DELETE DUMP    8 = OFFLOAD DUMP   9 = ONLOAD DUMP
                                                              TAPE
  OP     ---------DUMP NAME -------- ONLINE    DATE      TIME  LABEL
  -      SYSDUMP  F2        00000003   X      85/01/31  16:57:10
  -      SYSDUMP  F2        00000002   X      85/01/31  16:12:09
  -      SYSDUMP  F2        00000001   X      85/01/31  14:40:07
  -      SYSDUMP  F2        00000000   X      85/01/31  14:30:39
  -
  -
  -
  -

ONLOAD EXTERNAL DUMP  ?....... 2    2 = no, 1=yes
PF1=HELP                  3=END                  5=RETRY    6=PROCESS
```

Key setting available for DUMP viewing:

PF Key	Function
PF3	Cancel (End)
PF4	Cancel (End)
PF5	Address Search
PF6	Data String Search
PF7	Scroll Backward
PF8	Scroll Forward
PF9	Go To Top
PF10	Shift Right
PF11	Shift Left
PF12	Go to Bottom

ONLINE PROBLEM DETERMINATION (OLPD)

Online Problem Determination (OLPD) provides assistance in analyzing CICS transaction abends interactively. It collects, stores and analyzes error data from the transaction dump and provides the ability to list, display and delete this information.

Note carefully, that if one removes the invocation of IESOPDS from DFHPEP, OLPD will be inoperative. That is, it will not be invoked to collect problem information.

When a CICS transaction ends abnormally, the following actions occur:

- CICS writes the transaction dump to the active CICS dump dataset.
- OLPD gets control, builds and incident record and stores it.
- OLPD saves the screen in use at the time of the abend.
- OLPD shows the user and abend information panel.
- OLPD redisplays the original screen.
- CICS regains control and continues abend processing.
- CICS displays message DFH2005I on the user screen.

OLPD TASKS

The tasks available to the user to perform Online Problem Determination are:

- Display the last incident for a specific user.
- Purge the last incident for a specific user.
- List all incidents for a specific user.
- List all incidents for all users.

The user selects the desired task from the initial OLPD panel. Incidents are displayed either by selecting the last incident for a user or by selecting from a displayed list of incidents.

An incident report consists of several pages. The text of each page is divided into one or more paragraphs. Further information concerning each paragraph can be obtained by selecting the paragraph number. The number of pages and paragraphs varies according to the type of error.

VSAM SERVICEABILITY ENHANCEMENTS

The VSE/VSAM Rel. 3.1.0 SPE which is integrated in SP 2.1 makes the handling of the SNAP dump facility easier and faster. SNAP 0001, for example, prints error symptom messages for Catalog Management return codes at the console thus providing an error code trace during problem programs processing. The SNAP dump can now be invoked in any partition and be enabled or disabled for any partition without influencing execution or output in other partitions. A help function is also provided which shows how to enter SNAP requests.

Enabling and Disabling SNAP Dumps
The new snap dump support provides the following advantages:

There are now all 11 snap points per partition, i.e. a snap point can be enabled or disabled for only one partition at a time.

- No re-IPL is necessary after a snap point is changed; after each IPL all snap points are reset to their original states.

- The original states of the snap points after IPL are: snap 1 to 9 and 11 are in disable mode and snap 10 is in enable mode.

There are 11 snap points available in VSAM. Each snap ID, if enabled with IKQVEDA in a specified partition, will produce the result indicated (only for this partition). The output for snap dump 0001 goes to SYSLOG; the output for dumps 0002-0009 and 0011 goes to STSLST. Snap 0010 produces no output.

Snap Number Result of Enabling this Snap:

0001 This snap allows Catalog Management diagnostic information to be obtained (see Section "Using UPSI to Obtain Diagnostic Information for the VSAM Catalog" for details) (IKOVCAT).

As snap 0001 uses the UPSI byte, it cannot be run when the user program in the partition also uses the UPSI byte.

0002 This snap enables the Buffer Manager trace, which provides the current usage of VSAM buffering (IKQVRM).

0003 This snap enables the CLOSE control block dump at the beginning of CLOSE processing (IKQVCLOS).

0004 This snap enables the VSAM I/O trace facility (IKQVRM).

0005 This snap enables the I/O error trace (IKQVRM).

0006 This snap enables the OPEN control block dump facility when open processing is complete (IKQVOPEN).

0007 This snap enables the OPEN error trace. Control blocks are printed if an error occurs during open processing (IKQVOPEN).

Depending on where the error occurred in OPEN processing, Output may or may not be produced. For example, module IKQOPNDO (clean-up after open failure) invokes SNAP 0007, but IKQOPNDO is not invoked after every OPEN failure.

0008 This snap enables the Catalog Management I/O trace. All I/O operations done by catalog management are printed on SYSLST (IKQVCAT).

0009 This snap enables the VSAM Record Management error handler trace, allowing display of control blocks for any error detected by VSAM record management (IKQVRM).

0010 This snap enables automatic close. VSAM is shipped with this snap enabled. To disable automatic close, disable this snap ($$BACLOS).

0011 This support enables the managed-SAM control block trace. Refer to VSE/VSAM Management for SAM Feature Logic for further information (IKQSMMON).

CHAPTER 2. VSE/AF GENERAL INFORMATION

IPL CONTROL STATEMENTS

<table>
<tr><th>Operation</th><th>Operands</th></tr>
<tr><td>ADD</td><td>cuu[:cuu|..cuu][(S)],device-type[,ss|,ssss|
,ssssss|,SHR]</td></tr>
</table>

The ADD command is used to define the physical I/O devices attached to the system. The device addresses are entered into the PUB table. Either a single device or a series of devices of the same type can be added with one command.

cuu Indicates the channel and unit number of the device(s) to be added. Leading zeros can be omitted (for example, ADD 00C,2501 may be coded as ADD C,2501).

The format cuu:cuu or cuu..cuu indicates that a series of devices of the same type, starting with the first cuu and ending with the second cuu is to be added. For example

```
ADD 130:137,3330
```

defines eight 3330 devices with addresses 130 through 137. If this type of specification is used for 2703 devices, all addresses must designate either Start/Stop or BSC lines.

S Indicates that the device can be switched (that is, physically attached to two adjacent channels). The designated channel is the lower of the two channels. For the device on which the lock file resides, S must not be specified together with SHR.

device-type Specifies the device type code of the device to be defined;

ss
ssss
ssssss

Device specification (mode).
Tape.ss specifies the mode setting (see ASSGN Statement). If it is omitted, the following values are assigned:

C0 for 9-track tapes (2400, 3410 series)
D0 for 9-track tapes (3420, 3430 series)
60 for 8809 Magnetic Tape Unit
90 for 7-track tapes.

3284, 3286, 3287, 3288, 3289. ss must be entered as 01.

ss is required for a 3284/3286/3287 printer used as console printer for a 3277 operator console. The required entry is 02 or 04.

2702. ss specifies SADxx (Set Address) requirements:

00	for SAD0 (default)
01	for SAD1
02	for SAD2
03	for SAD3

1270, 1275, 1419, 1419P, and 1419S. ss specifies the external interrupt bit associated with magnetic ink or optical character readers. The settings 01 through 20 correspond to the external interrupt code in low storage byte 87, bits 7 through 2 respectively.

01	byte 87 bit 7
02	byte 87 bit 6
04	byte 87 bit 5
08	byte 87 bit 4
10	byte 87 bit 3
20	byte 87 bit 2

3704/3705/3725. ss is required and specifies the type of channel adapter:

01 Type 1/4 channel adapter
02 Type 2/3 channel adapter

For the 3705 SDLC Integrated Communication Adapter for the 4300 processors, the mode setting must be 10.

2703. For the 2703 BSC Integrated Communication Adapter for the 4331 Processor, ss can be:

40 EIB mode (Error Index Byte is to be set.)
00 non-EIB mode

Model 158, 3031, 3033. If you want to use, for example, a 3277 as operator communication device on a Model 158 or 3031, you can define the PA1 (request) key to be the PF-1 key by setting bit zero of the device specification byte to 1 (for example X'80'):

```
ADD 140,3277,80
```

SHR Indicates that the device to be added may be shared by two or more CPUs. SHR is valid only for the 33xx CKD device types and the 3370 FBA device. For the device on which the lock file resides, SHR may not be specified together with S. For ready disk units, the SHR option is reset during IPL if the disk unit is physically not shareable.

Operation	Operands
DEF	SYSREC={cuu|volser}[,SYSCAT={cuu|UA|volser}]

The DEF command, which is mandatory, is used to assign a physical device to

- SYSREC, the logical device for the system recorder file, the hardcopy file, and the system history file
- SYSCAT, the logical device for the VSE/VSAM master catalog

cuu Indicates the channel and unit number of the physical device to be assigned.
SYSCAT=cuu May be specified only if VSE/VSAM is installed in the system.
SYSCAT=UA Is the default value.
volser Indicates the unique volume serial number of the disk to be assigned.

The assignments cannot be changed until the next IPL.

Operation	Operands
DEL	{cuu[:cuu|..cuu]}

The DEL command is used to delete one or more of the I/O devices previously defined with the ADD command.

cuu Indicates the channel and unit number of the device(s) to be deleted.

The format cuu:cuu or cuu..cuu indicates that a series of devices of the same type, starting with the first cuu and ending with the second cuu is to be deleted. For example,

```
DEL 130:133
```

causes devices 130, 131, 132, and 133 to be deleted.

Operation	Operands
DLA	NAME=areaname{,UNIT=cuu]|,VOLID=volser}[,DSF={Y|N}] [,CYL=n[,NCYL=m]|,BLK=n[,NBLK=m]]

The DLA command, which is mandatory, defines or references a label information area. This area may be located on any disk device. Format and layout of the label information area are determined by the system.

The DLA command must be entered after the ADD (and DEL) commands and before the SVA command. Only one valid DLA command may be entered.

NAME=areaname Specifies the name of the label area, which can be one to eight alphameric characters. When the label area is first created, this name is entered in the VTOC of the device indicated in the UNIT or VOLID operand (in the form: DOS.LABEL.FILE.cpu-id.areaname). When referring to the label area during subsequent IPLs, use only the NAME and UNIT or VOLID operands.

UNIT=cuu Specifies the channel and unit number of the device containing the label area. The device type may be different from that of the SYSRES device. This operand may be specified together with VOLID.

VOLID=volser Identifies the unique volume serial number of the device containing the label area. This operand may be specified together with UNIT.

DSF=Y|N Specifies whether the label area is to be data-secured. If the operand is omitted, DSF=Y (Yes) is assumed.

CYL=n Indicates, for CKD devices, the sequential number of the cylinder, relative to zero, where the label area is to begin. **n** must be a decimal number with one to three digits, with a minimum value of 1.

BLK=n Indicates, for FBA devices, the sequential number of the block, relative to zero, where the label area is to begin. **n** must be a decimal number with one to three digits, with a minimum of 2.

NCYL=m Defines, for CKD devices, the size of the label area in number of cylinders. **m** must be a decimal number with one to three digits. If this operand is omitted, the device dependent default size is used. The maximum value for **m** is also device dependent. The table below gives the appropriate default and maximum values for the supported disk devices.

Device Type	NCYL Specification: Default	Maximum
2314	2	12
3330	2	13
3340	3	20
3350	1	8
3375	3	20
3380	3	16

NBLK=m Defines, for FBA devices, the size of the label area in number of blocks. **m** must be a decimal number with a minimum of 12 and a maximum of 992. If this operand is omitted, the default, which is 200 blocks, is used.

Note: If you want to use previously created standard labels, enter the DLA statement with only the NAME= and UNIT= or VOLID= operands. When you specify CYL and NCYL (or BLK and NBLK), any existing label area is destroyed.

Operation	Operands
DLF	{UNIT=cuu|VOLID=volser}[,CYL=n[,NCYL=n]|,BLK=n [,NBLK=n]][,DSF={Y|N}][,TYPE={N|F}][,NCPU=n]

The DLF command is used to define or reference the cross-system communication file (lock file). This file must be present when two or more VSE systems are linked in a DASD sharing environment. The DLF command is required if the supervisor was generated with DASD sharing support and if devices are present which are defined with the SHR option in the ADD command.

The lock file has to be on a disk drive which is physically shared with all systems linked in the DASD sharing environment. The device on which the lock file resides must not be defined as switchable. If used, the DLF command must be the first command after the ADD (and DEL) commands.

UNIT=cuu Specifies the channel and unit number of the device containing the lock file.

VOLID=volser Identifies the unique volume serial number of the disk containing the lock file.

No operands other than UNIT or VOLID are needed if an existing lock file is to be used. If, however, a new lock file is to be created or if a reallocation is required, the following operands are also needed:

CYL=n Specifies, for CKD devices, the sequential number of the cylinder, relative to zero, where the lock file is to begin. **n** must be a decimal number with one to three digits.

NCYL=n Specifies how many cylinders of a CKD device are to be allocated to the lock file. The default is 1.

BLK=n Specifies, for FBA devices, the sequential number of the block, relative to zero, where the lock file is to begin. **n** must be a decimal number with a minimum of 2.

NBLK=n Specifies how many blocks of an FBA device are to be allocated to the lock file. The default is 30.

DSF=Y|N Specifies whether the lock file is to be data-secured. If the operand is omitted, DSF=Y (Yes) is assumed.

TYPE=N|F N, which is default, indicates that the lock file is not to be formatted. If you specify TYPE=N, but the lock file does not exist on the specified device or volume, the system ignores the operand and formats a new lock file.

F indicates that the system should format the lock file during IPL. Use this option to reformat an already existing lock file.

NCPU=n Specifies the number of machines, real or virtual, in the DASD sharing environment. Valid specifications for **n** are 2..31. The default is 4.

If the CYL or BLK operand is specified, and a lock file has already been defined, an error message ('DUPLICATE NAME ON VOLUME') is returned if the extent limits do not match. If the extent limits do match, the existing lock file is used. A new lock file is **not** formatted, and the VTOC is **not** updated, regardless of the specification in the DSF operand.

The maximum number of resources that can be locked by a lock file of a given size can be calculated by the following formulae.

- For FBA devices:
  Number of resources = NBLK * 508 / (12 + NCPU)
- For CKD devices:
  Number of resources = NCYL * 508 / (12 + NCPU) * D,

where D is a device-type-dependent factor of:
308 for IBM 3330
144 for IBM 3340
810 for IBM 3350
480 for IBM 3375
690 for IBM 3380

<table>
<tr><th>Operation</th><th>Operands</th></tr>
<tr><td>DPD</td><td>{UNIT=cuu|VOLID=volser},{CYL=n|BLK=n}[,NCYL=m|,NBLK=m] [,TYPE={N|F}][,DSF={Y|N}]</td></tr>
</table>

The DPD command, which is mandatory, is used to define the page data set. The command is invalid if the VM/370 linkage facility is included in your system.

The operands of the DPD command may be given in any order.

UNIT=cuu Specifies the channel and unit number of the device that is to contain the page data set. If you omit this operand, you must specify the VOLID operand.

VOLID=volser Identifies the volume serial number (one to six alphabetic or numeric characters) of the disk pack that contains the page data set. If this operand is omitted the volume serial number is not checked.

CYL=n Specifies, for CKD devices, the sequential number of the cylinder, relative to zero, where the page data set is to begin (in decimal). A specification of CYL=0 indicates that the page data set extent is to begin on cylinder 0, track 1.

BLK=n Specifies, for FBA devices, the sequential number of the block, relative to zero, where the page data set is to begin. **n** must be a decimal number with a minimum of 2.

NCYL=m Specifies, for a multi-extent CKD page data set, the size of one page data set extent (in number of cylinders). **m** must be a decimal number with up to three digits.

NBLK=m Specifies, for a multi-extent FBA page data set, the size of one page data set extent (in number of blocks). **m** must be a decimal number with a minimum of 4.

TYPE=N TYPE=N is the default and indicates that the page data set need not be formatted. The TYPE operand is ignored for FBA devices.
If TYPE=N is specified, but the page data set does not exist, or the extent limits have been changed, TYPE=N is ignored and the page data set is formatted during IPL.

TYPE=F Indicates that the page data set is to be formatted during IPL. Formatting during IPL is required if the page data set has been damaged. A page data set on a shared CKD device is always formatted. The TYPE operand is ignored for FBA devices.

DSF=Y|N Indicates whether the page data set is to be data-secured. Yes is the default. For multi-extent page data sets, the DSF specification is valid for the **first** extent definition only; it is ignored for any further extent definitions.

For each extent of a multi-extent page data set, a separate DPD command has to be entered. After each command, the operator will be prompted to enter the next extent definition until

- the entire virtual storage is mapped on the specified extents, or
- the maximum number of extents allowed (which is 15) is exceeded, or
- the operator enters a DPD command without the NCYL/NBLK operand, in which case the complete remaining storage will be mapped on this extent.

Up to 15 extents can be specified, which may reside on different volumes; up to three extents may be allocated on one volume. The various extents can be placed on different CKD device types, or can be mixed with FBA device extents.

If the size specified in the NCYL/NBLK operand is larger than the size actually needed for the page data set, the free cylinders/blocks are available to the user.

Operation	Operands
SET	[DATE=mm/dd/yy,CLOCK=hh/mm/ss][,ZONE={EAST|WEST}/hh/mm

The SET command, which is optional, is used to set the system date, the time-of-day (TOD) clock, and the system time zone. It is required only if the TOD clock has not been set since the last POWER ON; IPL will then prompt the operator to enter the SET command. The command may be entered at any time before the SVA command.

DATE=mm/dd/yy Specifies the date in months (1-12), day of the month (1-31), and year (last two digits of the year).

After IPL this format can be changed to dd/mm/yy with the STDOPT command.

CLOCK=hh/mm/ss Specifies the local time-of-day in hours, minutes and seconds.

ZONE=EAST/hh/mm Specifies that the installation is located at a geographical position east of Greenwich.

ZONE=WEST/hh/mm Specifies that the installation is located at a geographical position west of Greenwich.

hh/mm Indicates the difference in hours and minutes between local time and Greenwich Mean Time. hh may be in the range 0-23, mm in the range 0-59.

The operands that have to be specified with the SET command depend upon the state of the TOD clock. The following groups can be distinguished:

1. If the TOD clock is in the set state, the command may be given in one of the three forms:

   ```
   SET ZONE=
   SET DATE= ,CLOCK=
   SET DATE= ,CLOCK= ,ZONE=
   ```

2. If the TOD clock is in the not-set state, the command **must** be given in one of the two forms:

   ```
   SET DATE= ,CLOCK=
   SET DATE= ,CLOCK= ,ZONE=
   ```

3. If the TOD clock is inoperative, the command must be given in the form:

   ```
   SET DATE= ,CLOCK=
   ```

Note: If the TOD clock is in the set state, message 0I30I is printed.
If the TOD clock is in the not-set state, message 0I31I is printed.
If the TOD clock is inoperative, messages 0I32I and 0I31I are printed.

Operation	Operands
SVA	[SDL=n][,PSIZE=nK][,GETVIS=nK]

This command is mandatory and must be the last command entered during the IPL procedure. It is used to allocate space within the SVA into which the user can later load his phases.

All operands are optional. If the operands are not entered during IPL, there will be no space reserved in the SDL and SVA for user phases.

SDL=n Specifies the decimal number of entries in the system directory list to be reserved for user phases and IBM-supplied phases, in addition to the phases loaded automatically during IPL. The maximum number that can be specified is 963, minus the number of SDL entries for the automatically loaded phases.
Note that only phases from IJSYSRS.SYSLIB and those generated with the SVA operand in the linkage editor PHASE statement can be loaded into the SVA at IPL.

PSIZE=nK Specifies the size of the area within the SVA which is to be reserved for user phases. **n** must be a decimal number and a multiple of 2.

GETVIS=nK Indicates the size of the **additional** system GETVIS area which you can specify beyond the minimum size (48K) allocated by the system. **n** must be a decimal number and a multiple of 2.

Operation	Operands
SYS	[CHANQ=n][,DASDFP=YES|NO][,SEC=YES|NO][,SUBLIB=m] [,JA=YES|NO][,SDSIZE=nK][BUFSIZE=n]

The SYS command, which is optional, specifies:

- The number of channel queue entries to be allocated,
- Whether DASD file protection should be active,
- Whether security checking should be active,
- The number of sublibraries which can be assigned.

- Whether job accounting should be active,
- The size of the shared area for system monitoring functions, and
- The number of supervisor buffers used for I/O processing,

CHANQ=n Specifies the number of channel queue entries to be allocated. If you omit the operand, the system allocates the appropriate number of channel queue entries for the number of partitions active and the type and number of devices added.

MINIMUM	n=p+a
DEFAULT	n=(2+d)*p+a+15 (If this expression yields a value greater than 255, the maximum value of 255 is used)
MAXIMUM	n=255
p= number of partitions specified in NPARTS operand. d= number of disk devices added by ADD commands. a= total number of ADDed devices.	

DASDFP=YES|NO Specifies whether DASD file protection should be active. If you omit the operand, the system does not activate file protection.

SEC=YES|NO Specifies whether access control should be active. If YES is specified, the system carries out access authorization checking, and, if VSE/Access Control is available, access logging is activated.

SUBLIB=m Specifies the number of sublibraries which may be attached to the whole VSE/Advanced Functions system at any one time. m must be a decimal integer from 10 to 2000. If the operand is omitted, the system uses the default value of 100.

JA=YES|NO Specifies whether job accounting should be activated. If YES is specified, CPU-times and SIO's for all devices are accounted.

SDSIZE=nK Specifies the size of a shared V=R area for system monitor functions, for example SDAID. This operand is valid only when the system is running in 370-mode. Valid specifications for **n** are from 0 to 256. The default is 48.

BUFSIZE=n Specifies the **number** of supervisor buffers to be used for I/O processing. The operand is invalid if the VM/370 linkage facility is included in the system (that is, the system is running in VM mode).

n must be a decimal number with a maximum of seven digits. The following table shows the minimum and default values for BUFSIZE. These are dependent on whether the supervisor is generated with the option FASTTR or not.

FASTTR=	Default	Minimum
NO	n=60	n=10

YES	370 mode: n=60+(p-2)*20
	ECPS:VSE mode: n=120+(p-2)*40
		n=30
		n=60

p = number of partitions specified in NPARTS.		

FASTTR and NPARTS are parameters of the FOPT and SUPVR generation macros, respectively.

Job Control Overview

		Valid for		
Type of Command or Statement	Operation	JCS	AR	JCC
Job Identification	JOB	X		
	/&	X		
User Identification	ID	X		X
File Definition	DLBL	X		
	EXTENT	X		
	TLBL	X		
	/*	X		
	/+	X		
Library Definition	LIBDEF	X		X
	LIBDROP	X		X
	LIBLIST	X		X
Pass Information to Operator	*	X		
Job Stream Control	BATCH		X	
	CANCEL		X	X
	PAUSE	X	X	X
	PRTY		X	X
	START		X	X
	STOP			X
	TPBAL		X	
	UNBATCH			see Note
Setting Symbolic Parameters	SETPARM	X		X
	PROC	X		X
Conditional Job Control	/.	X		
	GOTO	X		X
	IF	X		X
	ON	X		X
Setting System Parameters	ALLOC		X	X
	ALLOCR		X	X
	MSECS		X	X
	NPGR			X
	SET			X
	SIZE		X	X
	STDOPT	X		
Pass Information to Program	DATE	X		
	OPTION	X		
	OVEND	X		X
	UPSI	X		
Execution of Program	EXEC	X		X
	RSTRT	X		
Operator Communications	ALTER		X	
	DSPLY		X	
	DUMP		X	
	END or ENTER key		X	X
	IGNORE		X	X
	LOG		X	X
	MAP		X	X

Note: Valid only in a foreground partition.

Type of Command or Statement	Operation	Valid for		
		JCS	AR	JCC
Operator Communications	MODE		X	
	MSG		X	
	NEWVOL		X	
	NOLOG		X	
	ONLINE		X	
	RC		X	
	REPLID		X	
	SETMOD		X	
	UNLOCK		X	
	ZONE	X		
Control of I/O System	ASSGN	X		X
	CLOSE	X		X
	DVCDN			X
	DVCUP			X
	FREE		X	
	HOLD			X
	LFCB		X	
	LISTIO	X		X
	LUCB		X	
	MTC	X		X
	PWR	X		X
	RESERV		X	
	RESET	X		X
	ROD			X
	SETDF		X	
	SETPRT	X		X
	UCS			X
	VOLUME		X	

JOB CONTROL STATEMENTS SUMMARY

ASSGN	Used at execution time to assign a specific device address to the symbolic unit name used.
CLOSE	Closes either a system or a programmer logical unit assigned to tape, disk, or diskette.
DATE	Contains a date that is put in the communications region.
DLBL	Contains file label information for DASD or diskette label checking and creation.
EXEC	Indicates the end of job control statements for a job step and that the job step is to be executed.
EXEC PROC	Calls a cataloged procedure and defines values for symbolic Parameters.
EXTENT	Defines each area, or extent, of a DASD file or diskette volume.
GOTO	Causes JC to skip all following statements (except JOB, /&, /+) up to the specified label statement.
ID	Used to specify user identification and password.
IF	Causes skipping or execution of the following statement dependent on the specified condition.
JOB	Indicates the beginning of control information for a job.
LIBDEF	Defines library chains.
LIBDROP	Drops library chain definitions.
LIBLIST	Lists library chain definitions.
LISTIO	Used to get a listing of I/O assignments on SYSLOG or SYSLST.
MSECS	Changes or displays the time-slice for partition balancing.
MTC	Controls operations on magnetic tapes.
NPGR	Defines the number of programmer logical units which may be assigned to a partition.
ON	Causes specified action to be done if the specified condition is true after any step in the following job stream.
OPTION	Specifies one or more of the job control options.
OVEND	Indicates that no more overwrite statements will follow for the respective procedure.
PAUSE	Causes a pause immediately after processing this statement, or at the end of the current job step.
PROC	Defines and initializes symbolic parameters in a procedure.
PWR	Passes a PRELEASE or PHOLD command to POWER.
RESET	Resets I/O assignments to the standard assignments.
RSTRT	Restarts a checkpointed program.
SETPARM	Assigns a character string or return code to the specifeid parameter.
SETPRT	Loads the IBM 3800 buffers.
STDOPT	Resets system defaults.
TLBL	Contains file label information for tape label checking and writing.
UPSI	(User Program Switch Indicators.) Allows the user to set program switches that can be tested.
ZONE	Initializes the zone field in the communications region.

Delimiter Statements	
/.	Label statement.
/*	Indicates the end of a data file.
/&	Indicates the end of a job.
*	Job control comments.
/+	Indicates the end of a procedure or librarian End-of-Data.

Operation	Operands	Type
ALLOC	[space-id,]part={nK|mM}[part={nK|mM}]...	JCC,AR

space-id Indicates in which address space the specified amount of storage for the named partition(s) is to be allocated. Valid specifications are:

370 Mode:

1, 2 or 3	In the specified virtual address space
S	In the shared virtual address space
R	In the real address space (processor storage).

ECPS:VSE Mode or VM Mode:

1	In virtual address space
R	In real address space.

Note:

- An ALLOC R command for a given partition can be given only after an ALLOC n for the same partition.

- The default value of space-id for all modes is 1.

part Indicates the partition to which storage is to be allocated. Valid specifications are:

BG, F1..F9, FA, FB.

Restriction If the system is generated with less than 12 partitions, the number of foreground (Fn) partitions is reduced accordingly. For a 5-partition system, for example, valid specifications are BG and F1..F4.

nK Specifies, in kilobytes, the amount of storage to be allocated. Valid specifications are 0 or an integer. Specifying 0 means that the entire storage allocated to the named partition is freed. The partition can no longer be used.

The system rounds up the specified integer to a multiple of:

- 64 in 370 mode for virtual address spaces;
- 2 in 370 mode for real address space;
- 2 in ECPS:VSE mode for real and virtual address space;
- 4 in VM mode.

The resulting rounded-up value must be at least

- 2K in real address space allocations, and
- 128K in virtual address space allocations.

mM Specifies, in megabytes, the amount of storage to be allocated. Valid specifications are 0 or an integer. Specifying 0 means that the entire storage allocated to the named partition is freed. The partition can no longer be used.

Operation	Operands	Type
ALTER	[space-id,]address	AR

space-id Specifies in which address space the alteration at the given address is to be made. Valid specifications are:

- R, 1..3 in 370 mode;
- 1 in ECPS:VSE mode and VM mode.

The default value for all modes is 1.

address Indicates the six-digit hexadecimal address, with leading zeros if necessary, at which storage alteration is to start. The highest address that can be specified is 16 MB minus 15 (FFFFF0).

If the specified address is within the supervisor area or the shared virtual area (SVA), a message is issued and the operator has the option to cancel the command or to change the address.

If the specified address is within an invalid address range, the command is ignored and a corresponding information message issued.

If the bytes to be altered cross the boundary from a valid to an invalid address space, the command is ignored and a corresponding information message is issued.

Operation	Operands	Type
[//] ASSGN	SYSxxx, {cuu|(address-list)|UA|IGN|SYSyyy| device-class|device-type}	JCC, JCS
	Optional Operands for Disk Devices Only	
	[,TEMP|,PERM][,VOL=volser][,SHR]	
	Optional Operands for Diskette Devices Only	
	[,TEMP|,PERM][,VOL=volser]	
	Optional Operands for Tape Devices Only	
	[,ss|,ALT][,TEMP|,PERM][,VOL=volser]	
	Optional Operands for Device Types 2560 and 5424/5	
	[,H1|,H2][,TEMP|,PERM]	
	Optional Operands for Any Other Devices	
	[,TEMP|,PERM]	

SYSxxx

Represents the logical unit name. It can be one of the following:

SYSRDR
SYSIPT
SYSIN
SYSPCH
SYSLST
SYSOUT
SYSLNK
SYSLOG
SYSnnn

SYSnnn represents all the other logical units in the system. For nnn, specify a decimal number from 000 to 254.

Restrictions: The type of device assignment is restricted under certain conditions:

1. If one of the system logical units SYSRDR, SYSIPT, SYSLST or SYSPCH is assigned to a disk device or diskette, the assignment must be permanent and follow the DLBL and EXTENT statements.
2. If SYSRDR and SYSIPT are to be assigned to the same disk device or diskette, SYSIN must instead be assigned and this assignment must be permanent.
3. SYSOUT is only valid for a tape unit and must be assigned permanently.
4. SYSLOG can only be assigned permanently.
5. If SYSIPT is assigned to a tape unit, it should be a single file and a single volume.
6. You may not assign SYSLOG to a 3278 Model 2A or 3279 Model 2C with a message area of 16 lines if IPL was done from a 3277, 3278 or 3279 with a message area of 20 lines.

7. SYSLOG cannot be assigned to a console printer (3284, 3286, 3287, 3288).
8. ASSGN SYSLOG,UA and ASSGN SYSLOG,IGN are not accepted.
9. If a system logical unit is assigned to a tape, DASD, or diskette, the unit must be closed (using the CLOSE command) before it can be reassigned.
10. When SYSOUT is assigned to a magnetic tape device it must not be the permanent assignment of either SYSLST or SYSPCH.
    Before assigning a tape drive to a system output unit (SYSOUT, SYSLST, SYSPCH), all previous assignments of this tape drive to any system input units and to any programmer units (input or output) must be permanently unassigned. The assignment of SYSOUT must always be permanent.
    Also, before assigning a tape to a system input unit or any programmer unit, all previous assignments of this tape to any system output unit must be permanently unassigned.
11. A programmer logical unit cannot be assigned to SYSLST if SYSLST has been assigned to tape or disk before.
12. ASSGN SYSRDR and ASSGN SYSIPT are allowed within a cataloged procedure. SYSRDR assignments, and SYSIPT assignments in a procedure with a DATA=YES specification, become effective on returning to JC level 0. SYSIPT assignments in a procedure with DATA=NO specification become effective immediately.
13. In a system with 16 channels, ASSGN SYSxxx,FBA cannot be used to address unit BA on channel F. Use ASSGN SYSxxx,X'FBA' to distinguish the cuu specification from the device class specification FBA (for Fixed Block Architecture).

cuu|X'cuu' Indicates the physical unit address to which the specified logical unit is to be assigned.
    c = channel number
    uu = unit number

The form X'cuu' must be used when the physical address of the specified device is FBA (that is, channel F, unit BA), to distinguish it from the device class FBA (for Fixed Block Architecture disk unit). Otherwise, the X'' can be omitted.

(address-list) You can specify a list of up to seven device addresses in the form cuu, separated by commas and enclosed in parentheses. In this case the system searches only the addresses specified in the address list for a free unit, starting with the first specified device address. Once a free unit is found, it is assigned to SYSxxx for the job in which the assignment is made.

For disks, if SHR is specified, the first unit in the list is assigned, even if previously assigned.

UA Indicates that the logical unit is to be unassigned. Any operation attempted on an unassigned device cancels the job.

IGN The IGN option is not valid for SYSRDR, SYSIPT, or SYSIN, nor for PL/I programs. The IGN option can be made temporary by specifying the TEMP option.

SYSyyy This may be any system or programmer logical unit, except SYSCAT and SYSREC.

device-class In this case the specification of READER, PRINTER, PUNCH, TAPE, DISK, CKD, FBA, or DISKETTE is allowed for the devices listed below.

READER
1442N1, 2501, 2520B1, 2540R, 2560, 2596, 3505, 3525RP, 5425

PRINTER
PRT1, 1403, 1403U, 1443, 3203, 3800, 3800B, 3800C, 3800BC, 3200, 4245, 4248

PUNCH
1442N1, 1442N2, 2520B1, 2520B2, 2520B3, 2540P, 2560, 2596, 3525P, 3525RP, 5425

TAPE
2400T7, 2400T9, 8809, 3410T7, 3410T9, 3420T7, 3420T9, 3430

DISK
2311, 2314, 3330, 3330B, 3340, 3340R, 3350, FBA, 3375, 3380

CKD
2311, 2314, 3330, 3330B, 3340, 3340R, 3350, 3375, 3380

FBA

DISKETTE
3540

device-type This can be any supported device as shown under the **device-class** specification, including the 8809.

For a 3800 printing subsystem, you can use assignment by device codes as follows:

Specified code	is valid for			
	3800	3800B	3800C	3800BC
3800	X	X	X	X*,**
3800B		X		X*
3800C			X	X**
3800BC				X

* The job cannot use the additional character generation storage feature.
** The job cannot use the Burster-Trimmer-Stacker feature.

ss Specifies mode settings for magnetic tapes. If **ss** is not specified at IPL time, the system assumes:

90 for 7-track tapes
C0 for 9-track tapes (2400, 3410 series)
D0 for 9-track tapes (3420 series)
D0 for 9-track tapes (3430 series)
60 for the 9-track 8809 Magnetic Tape Unit

Density (bpi)	Parity	Convert Feature	Translate	ss
200	odd	on	off	10
200	odd	off	off	30
200	odd	off	on	38
200	even	off	off	20
200	even	off	on	28
556	odd	on	off	50
556	odd	off	off	70
556	odd	off	on	78
556	even	off	off	60
556	even	off	on	68
800	odd	on	off	90
800	odd	off	off	B0
800	odd	off	on	B8
800	even	off	off	A0
800	even	off	on	A8
800	single-density 9-track tapes			C8
800	dual-density 9-track tapes			C8
1600	single or dual-density 9-track tapes			C0
6250	single/dual density, 9-track			D0
1600	Streaming: high speed and long gap			90
D/T	Streaming: high speed and short gap			30
8809	Start-Stop: low speed and long gap			50
	Start-Stop: low speed and short gap			60

ALT Indicates an alternate magnetic tape unit that is used when the capacity of the original assignment is reached. This operand can only be specified for programs using logical IOCS.

The specifications for the alternate unit are the same as those of the original unit. The characteristics of the alternate unit must be the same as those of the original unit. The original assignment and an alternate assignment must both be permanent or both be temporary assignments. Multiple alternates can be assigned to one symbolic unit.

The ALT operand is invalid for SYSRDR, SYSIPT, SYSIN, SYSLNK, and SYSLOG.

H1 Indicates that input hopper 1 will be used for input on the 2560, 5424, or 5425. If neither H1 nor H2 is specified, H1 is assumed.

H2 Indicates that input hopper 2 will be used for input on the 2560, 5424, or 5425. Note that hopper specifications are only valid with the logical units SYSIPT, SYSRDR, SYSIN, and SYSPCH.

If both hoppers are used, they must be assigned to the same partition.

PERM
TEMP Indicates whether the assignment should be permanent (PERM) or temporary (TEMP). It is thus possible to override the // specification or omission.

A permanent assignment overrides the current assignment and deletes the stored permanent and all alternate assignments.

VOL=no. Specifies the volume serial number of the device required. This option may be specified only for tapes, disks, and diskettes.

SHR This option can be specified only for disk devices and is meaningful only in combination with **address-list, device-class**, and **device-type**.

Operation	Operands	Type
BATCH	[BG|Fn]	AR

The BATCH command (Start or Continue Processing) serves to activate or continue processing in one of the foreground partitions or in the background partition.

BG Indicates that the background partition is to be reactivated.

Fn Indicates that the specified foreground partition is to be activated or restarted after having been stopped by a STOP command.

If the operand is omitted, BG is assumed.

Operation	Operands	Type
CANCEL	blank	JCC
CANCEL	cuu	AR
CANCEL	{BG|Fn}[,DUMP|,PARTDUMP|,NODUMP] [,SYSDUMP|,NOSYSDUMP][,FORCE]	AR

The CANCEL command (Cancel Job), when issued as a JCC, cancels the execution of the current job in the partition in which the command is given.
or
when issued as an attention routine command, may

- cancel an I/O request on a device for which operator intervention was requested.
- cancel the execution of the current job in the specified partition and, optionally, to override the dump options existing for that partition.

cuu Indicates that the I/O request for the specified device is to be canceled.

BG Indicates that the background job is to be canceled.

Fn Indicates that the specified foreground job is to be canceled.

DUMP Causes a dump of the registers, of the supervisor, of the partition, the used part of the system GETVIS area, and of the SVA phase in error (if the error occurred in the SVA).

PARTDUMP Causes a dump of the registers, of supervisor control blocks, of the partition, of areas acquired through GETVIS in the partition, and of the SVA phase in error (if the error occurred in the SVA).

NODUMP Suppresses the DUMP option.

NOSYSDUMP Indicates that dumps are to be written on SYSLST. The form }NOSYSDMP} is accepted for compatibility reasons.

SYSDUMP Indicates that dumps are to be written to the dump sublibrary which is defined for the appropriate partition. If no LIBDEF DUMP statement is in effect for the partition in question, or if the defined sublibrary is full, the system assumes the option NOSYSDUMP. The form }SYSDMP} is accepted for compatibility reasons.

FORCE Causes the Cancel command to be carried out immediately, even if a critical system function has requested a delay.

Operation	Operands	Type
[//] CLOSE	SYSxxx [,cuu[,ss]|,UA|,IGN|,ALT|,SYSyyy |,device-class|,device-type]	JCC,JCS

The CLOSE **command** (Close Output Unit) is used to close either a system or programmer logical unit assigned to a tape, or a system logical unit assigned to a disk or diskette.

The CLOSE **statement** is used to close either a system or programmer logical unit assigned to tape. It applies only to temporarily assigned logical units.

SYSxxx For the CLOSE command only: For disk or diskette: SYSIN, SYSRDR, SYSIPT, SYSPCH, or SYSLST.

For both the statement and the command: For magnetic tape: SYSPCH, SYSLST, SYSOUT, or SYS000-SYS254.

cuu Specifies that, after the logical unit is closed, it will be assigned to the specified channel and unit.

ss Device specification for mode settings on 7-track and 9-track tape. The specifications are shown under ASSGN.

UA Specifies that the logical unit is to be (permanently) unassigned after the file has been closed.

IGN Specifies that the logical unit is to be (permanently) unassigned after the associated file has been closed. This operand is invalid for SYSRDR, SYSIPT, or SYSIN.

ALT Specifies that the logical unit is to be closed and an alternate unit is to be opened and used. This operand is valid only for system output logical units (SYSPCH, SYSLST, or SYSOUT) currently assigned to a magnetic tape unit.

SYSyyy Specifies that, after SYSxxx is closed, it will be assigned to the physical device to which SYSyyy is currently assigned (and to which it remains assigned). If SYSxxx is a system logical unit, it will be opened if the target device is a disk, diskette, or magnetic tape at load point, and if SYSxxx is not already assigned.

device-class Indicates that after the logical unit is closed, it will be assigned to the first available unit within the specified device class, shown under ASSGN.

device-type After the logical unit is closed, it will be assigned to the first free unit of the specified device type. Note that 8809 (not listed under TAPE) can also be specified as device type.

Operation	Operands	Type
// DATE	{mm/dd/yy|dd/mm/yy}	JCS

The DATE statement places the specified date temporarily in the communication region. This date overrides the date given in the SET command, either during or after IPL. The date from the DATE statement is reset at end of job to the date from the last SET command.

mm = month (01-12)
dd = day (01-31)
yy = year (00-99)

Operation	Operands	Type
// DLBL	filename,['file-ID'],[date],[codes][,DSF] [,BUFSP=n][,CAT=filename][,BLKSIZE=n|MAX] [,CISIZE=n][,DISP=disposition][,RECORDS=n] [,RECSIZE=n]	JCS

The DLBL statement (Disk label information) contains file label information for Disk or diskette label checking and creation.

filename This can be from one to seven alphameric characters, the first of which must be alphabetic. This unique filename is identical to the symbolic name of the program DTF that identifies the file.

'file-ID' This can be from one to 44 characters, contained within quotes. If this operand is omitted, **filename** is used. The diskette uses a maximum of eight characters in file-ID.

date This can be from one to eight characters indicating either the retention period of the file in days (dddd=0-9999), or the absolute expiration date of the file as a Julian date in the format 19yy/ddd or 20yy/ddd.

The format 20yy/ddd is not accepted for diskette files.

Default is a 7-day retention period (based on the date entered via the SET command).

codes This is a two to four character field indicating the type of file label, as follows:

SD	for sequential disk or for DTFPH with MOUNTED=SINGLE
DA	for direct access or for DTFPH with MOUNTED=ALL
DU	for diskette
ISC	for indexed sequential using load create
ISE	for indexed sequential using load extension, add, or retrieve
VSAM	for all Virtual Storage Access Method files

If this operand is omitted, SD is assumed.

DSF This operand indicates that a data secured file is to be created or processed.

BUFSP=n If a VSE/VSAM file is to be processed, this operand specifies the number of bytes of virtual storage (0-999999) to be allocated as buffer space for this file.

CAT=filename This operand is only valid in a DLBL statement for a VSE/VSAM file. It specifies the filename (1 to 7 alphameric characters) of the DLBL statement for the catalog owning this VSE/VSAM file.

BLKSIZE=n|MAX This operand permits specification of a block size different from that given in the DTFSD macro for sequential DASD files. The parameter is ignored for all DTF types except DTFSD. It is not valid for VSE/VSAM files, or files on FBA devices. The value specified for n must not exceed 65,536.
If BLKSIZE=MAX, the system automatically takes a block size equivalent to one whole track of the device on which the file resides.
CISIZE=n This operand permits specification of a control interval size for SAM files on FBA devices in order to improve space utilization on such devices. The specified size must be a number from 512 to 32,768 and a multiple of the FBA block size. If it is greater than 8K, it must be a multiple of 2K.

DISP=disposition This operand is valid only in a DLBL statement for a VSE/VSAM file. **disposition** can be specified in one of the following formats:

```
NEW
(NEW,KEEP)
(NEW,DELETE)
(NEW,DATE)
OLD
(OLD,KEEP)
```

```
                (OLD,DELETE)
                (OLD,DATE)
                (,KEEP)
                (,DELETE)
                (,DATE)
```

RECORDS=n This operand is only valid for VSE/VSAM space management for SAM feature files. The operand can be specified in one of two formats:

```
    RECORDS=n
    RECORDS=(n,n1)
```

where **n** indicates the number of records for the primary data set allocation, and **n1** the number of records for the secondary data set allocation. **n** must not be zero.
The RECORDS and RECSIZE operands must either both be specified or both be omitted.

RECSIZE=n This operand is valid only for VSE/VSAM space management for SAM feature files. It permits specification of the average record length of the file. The value specified for **n** must not be zero. The RECSIZE and RECORDS operands must either both be specified or both be omitted.

Operation	Operands	Type
DSPLY	[space-id,]address	AR

The DSPLY command (Display Virtual Storage) allows the operator to display 16 bytes of virtual storage, starting at the specified hexadecimal address, on the device assigned to SYSLOG.

space-id Indicates in which address space the specified address is to be displayed. Valid specifications are:

```
    R, 1..3 in 370 mode;
    1       in ECPS:VSE and VM mode.
```

To display virtual storage in a **shared** area (370 mode only) specify the space-id of any **active** virtual address space.

address Specifies the six-digit hexadecimal address, with leading zeros if necessary, at which the storage display is to start. The highest address that can be specified is 16MB minus 15 (FFFFF0).

Operation	Operands	Type
DUMP	{SUP|BG|Fn|SVA|[space-id,]addr-addr|BUFFER}, cuu	AR

The DUMP command allows the operator to dump specified areas of virtual storage on a printer or a tape device. The first operand specifies which areas of storage are to be dumped, as follows:

SUP The control registers and the supervisor areas.

BG, Fn The PSW, general and floating-point registers from the partition save area, and the active real or virtual partition specified.

SVA The SVA, including the system GETVIS area and the VIO V-pool area.

space-id Part of the address space specified by this operand. Valid specifications are:

```
    R, 1..3 for 370 mode;

    1       for ECPS:VSE and VM mode.
```

The default value for all modes is 1.

addr-addr The virtual storage between the specified addresses in the address space indicated by the space-id operand. If any active real or virtual partition, or a part of such a partition, lies between the specified addresses, its PSW and associated registers are dumped.

BUFFER The contents of the SDAID buffer. This operand is accepted only if the dump is directed to a tape device.

cuu Specifies the device on which the output is to be written. It can be a printer or tape device. For BUFFER only a tape unit address is accepted. Tape output is written without repositioning the tape to allow for several dumps per tape.

Operation	Operands	Type
DVCDN	cuu	JCC

The DVCDN command (Device Down) informs the system that a device is no longer available for system operation. It is used when a device is to be serviced or becomes inoperative.
A DVCUP command must be issued before the device can be used again.

Note: A DVCDN command is not accepted for a device on which SYSRES, SYSREC, SYSCAT, SYSDMP or the page data set resides.

cuu **c** is the the channel number and **uu** the unit number, in hexadecimal, of the device to be made unavailable.

Operation	Operands	Type
DVCUP	cuu	JCC

The DVCUP command (Device Up) informs the system that a device which was inoperative is now available again for system operation. As all assignments for this device were removed by the preceding DVCDN command, the device must be reassigned by an ASSGN statement or command.

cuu **c** is the channel number and **uu** the unit number, in hexadecimal, of the device to be made available.

Operation	Operands	Type
[//] EXEC	[[PGM=]progname][,REAL][,SIZE=size][,GO] [,PARM='value']	JCS,JCC
[//] EXEC	PROC=procname [,parname[=[value]]][...]	JCC,JCS
[//] EXEC	PROC=procname[,OV]	JCS,JCC

The EXEC command or statement (Execute Program or Procedure) indicates either

- the end of control information for a job step and the beginning of execution of a program.
- that a cataloged procedure is to be retrieved from a sublibrary by job control.

The operands must be entered in the specified order.

The **statement** can be issued from SYSLOG or from SYSRDR. Control will return to the unit from which the statement was issued. If specified with a procedure name, control will always return to SYSRDR. The **command** can be issued only from SYSLOG and control will return there.

Continuation lines are accepted for the EXEC statement.

PGM=progname Represents the name of the program to be executed. The program name corresponds to the first or only phase of the program in the library.

REAL Indicates that the program will be executed in real mode. If REAL is not specified, the program is always executed in virtual mode.

SIZE=size The SIZE parameter can be specified in combination with REAL or without REAL. The SIZE parameter can be specified in the following formats:

```
SIZE= nK|mM
SIZE=AUTO
SIZE=(AUTO, nK|mM )
SIZE=phasename
SIZE=(phasename, nK|mM )
```

where **n** or **m** must be greater than zero and **n** must be a multiple of 4 (if not, the system rounds the value up to the nearest 4K boundary). **nK** or **mM** must not exceed the size of the partition (as defined by ALLOC) minus the minimum partition GETVIS area of 48K bytes.

AUTO indicates that the program size, as calculated by the system from information in the sublibrary directory, is to be taken as the value for SIZE.

AUTO, nK|mM indicates that job control must take the program size plus nK or mM bytes as the value for SIZE.

phasename indicates that the length of the specified phase, increased by its relative load address in the partition, is to be taken as the value for SIZE, regardless of other phases with the same first four characters in their names.

phasename,(nK|mM) indicates that the length of the specified phase, increased by its relative load address in the partition, **plus nK or mM bytes**, is to be taken as the value for SIZE. If this value is not a multiple of two, it is rounded up.

GO Specifies, for a language translator step, that the program is to be link-edited and executed automatically after it has been compiled.

PARM='value' Specifies information which is to be passed to the program at execution. **value** can be up to 100 characters in length, enclosed in quotes. (The enclosing quotes are not passed to the program.) An quote within **value** must be coded as two single quotes.

PROC=procname Represents the name of the procedure to be retrieved from a sublibrary.

If the procedure name begins with $$, the system substitutes a partition-related character for the second $. The character that is substituted is related to the partition in which the procedure is invoked, that is,

0 for the BG partition
B for the FB partition
A for the FA partition
9 for the F9 partition
. . .
1 for the F1 partition.

The procedure corresponding to this name is then retrieved for execution. There are three methods of addressing symbolic parameters in the EXEC PROC statement or command:

```
parname1=value1

parname2

parname3=&parname3
```

parname1 Specifies the name of a symbolic parameter which is to be substituted in the specified procedure. It must consist of 1 to 7 alphanumeric (including national) characters, and the first character must be alphabetic.

value1 Specifies the actual value which is to be inserted in the specified procedure in place of the specified symbolic parameter. It must be a string of up to 50 characters. If the string is alphanumeric, no enclosing quotes are necessary. If it contains national or special characters, it must be enclosed in quotes, which will not be passed to the procedure. No quotes are allowed in the string itself.

parname2 Is the name of a symbolic parameter which is to be passed to a lower-level procedure and back.

parname3 Is the name of a symbolic parameter which is to be passed to a lower-level procedure. The symbolic parameter name after the equals sign must be coded with the ampersand (&).

OV Indicates that overriding statements follow EXEC. This operand must not be used when the called procedure contains symbolic parameters or calls a nested procedure, or if it contains IF, ON, GOTO or SETPARM statements.

Operation	Operands	Type
// EXTENT	[logical-unit], [serial-number],[type], [sequence-number], [relative-track|block], [number-of-tracks|blocks], [split-cylinder-track]	JCS

The EXTENT statement (DASD Extent Information) defines each area, or extent, of a DASD file.

logical unit A six-character field indicating the logical unit (SYSxxx) of the volume for which this extent is effective.

This operand is not required if a system file with IJSYSxx as filename is specified.

serial number From one to six characters indicating the volume serial number of the volume for which this extent is effective. If fewer than six characters are used, the field is padded on the left with zeros, unless you enclose it in quotes, in which case it is padded on the right with blank characters.

type One character indicating the type of the extent, as follows:
1 - data area
2 - independent overflow area
4 - index area
8 - data area
If this operand is omitted, type 1 is assumed. Type 1 is the only valid parameter for diskette files.

sequence number One to three characters containing a decimal number 0 to 255 indicating the sequence number of this extent within a multi-extent file.

relative track|block For CKD devices, this operand is one to five characters indicating the sequential number of the track, relative to zero, where the data extent is to begin.

For FBA devices, this operand is a number from 2 to 2,147,483,645 which specifies the physical block at which the extent is to start.

For VSE/VSAM, this operand must be specified when a data space or a file with the UNIQUE option is being created.

number of tracks|blocks For CKD devices, this operand is one to five characters indicating the number of tracks to be allocated to the file.

For FBA devices, this operand is a number from 1 to 2,147,483,645 which specifies the number of physical blocks in the extent.

split cylinder track A one or two-digit decimal number, indicating the upper track number for the split cylinder in SAM files.

Operation	Operands	Type
FREE	cuu	AR

The FREE command is used to reset the RESERVED status (as caused by the RESERV command) of the specified device. The command may be issued for all DASD devices on the system.

cuu Indicates the channel and unit number of the device to be freed.

Operation	Operands	Type
[//] GOTO	label	JCC,JCS

The GOTO statement causes all statements in the following job stream to be skipped, up to the specified label statement. It is accepted only within a job.

label Specifies the operand of the /. statement at which execution of the current job is to continue. Code $EOJ to skip all statements up to end-of-job.

Operation	Operands	Type
HOLD	Fn[,Fn]...	JCC

The HOLD command is used to hold assignments or sublibrary definitions made via LIBDEF before you issue a command to unbatch a foreground partition. The partitions may be specified in any sequence; at least one partition must be given. **n** indicates the desired partition.

Operation	Operands	Type
[//] ID	USER=user-id, PWD=password	JCS,JCC

The ID statement or command is used to specify the user identification and the user's password.

user-id Specifies the user identifier, which must be four alphameric characters.

password Specifies the password of the user, which can be three to six alphameric characters.

Operation	Operands	Type
[//] IF	{$RC comparator n } [OR ...] THEN {$MRC comparator n } [| ...] {pname comparator value} [AND ...] [& ...]	JCC,JCS

The IF statement is a local conditional function. When it occurs in the job stream, the condition is checked; if it is true, the following statement is executed; if not, the following statement is skipped.

$RC Specifies the return code of the preceding job step.

$MRC Specifies the maximum return code of all preceding steps within the current job.

pname Specifies the name of a parameter to be compared.

comparator Specifies the comparison to be done.

Equal	specified as = or EQ
Not equal	specified as ¬= or NE
Greater than	specified as > or GT
Less than	specified as < or LT
Greater or equal	specified as >= or GE

Less or equal specified as <= or LE

n specifies a decimal integer from 0 to 4095.

value specifies a character string of 0 to 50 characters. If the string contains special characters, it must be enclosed in quotes.

You can specify two conditions in the IF statement, separated by one of the logical operators OR, |, AND, &. The logical operators OR, |, AND, & must be preceded and followed by a blank character.

Operation	Operands	Type
IGNORE	none	JCC,AR

On an abnormal condition, the operator will be notified by an appropriate message on SYSLOG. Depending on the situation, he may have to ignore the condition by entering an IGNORE command without any operand.

Operation	Operands	Type
// JOB	jobname [accounting information]	JCS

The JOB statement indicates the beginning of control information for a job.

jobname The name of the job. Must be one to eight alphameric characters (0-9, A-Z, #, $, @) or /, -, or '.'.

accounting information If the job accounting interface has been specified during system generation, max. 16 characters of user information are moved to the job accounting table.

Operation	Operands	Type
LFCB	cuu,phasename[,FORMS=xxxx][,LPI=n][,NULMSG]	AR

The LFCB command causes the system to load a buffer image, stored as a phase, into the forms control buffer (FCB) of the specified printer.

cuu Specifies the channel and unit number of the printer whose FCB is to be loaded.

phasename Specifies the name of the phase that contains the applicable buffer load image.

FORMS=xxxx Specifies the installation-defined forms number xxxx of the paper that is to be used with the new FCB image.

LPI=n Indicates (for a printer other than a PRT1) the required setting of the carriage clutch. For n, you can either substitute 6 (six lines per inch) or 8 (eight lines per inch).

NULMSG Specifies that the printing of a buffer load verification message is to be suppressed.

<table>
<tr><th>Operation</th><th>Operands</th><th>Type</th></tr>
<tr><td>[//] LIBDEF</td><td>{type|*} [,SEARCH=(lib.sublib,...)]
[,CATALOG=lib.sublib][{,TEMP|PERM}]</td><td>JCC,JCS</td></tr>
</table>

The LIBDEF statement defines which sublibraries are to be searched for members of a specified type or types, or the sublibrary in which new phases or dumps are to be stored.

The system sublibrary IJSYSRS.SYSLIB is always added at the end of the search chain, unless it is explicitly included at a different position in the chain.

type Defines the member types for which this LIBDEF statement applies. For type, specify:

PHASE To define a library/sublibrary chain to be used for loading or fetching program phases for execution. The CATALOG operand specifies the library and sublibrary in which phases are to be cataloged by the linkage editor.

OBJ To define a library/sublibrary chain to be used by the linkage editor when searching for object modules.

SOURCE To define a library/sublibrary chain to be used, for example by language translators, when searching for one of the predefined }SOURCE} types (A-Z, 0-9, #, $, @).

PROC To define a library/sublibrary chain to be used by Job Control when searching for a procedure to be executed.

* To indicate that the LIBDEF statement applies to all member types except DUMP and user types.

DUMP To define a library/sublibrary to be used by the system when a dump is to be produced and the option SYSDUMP is in effect, or a CANCEL command with the SYSDUMP operand is issued. You must use the keyword CATALOG if you specify DUMP as the type operand.

SEARCH=lib.sublib Is required if you specified OBJ, SOURCE or PROC in the type operand. With type PHASE, or * you must specify SEARCH or CATALOG or both.

CATALOG=lib.sublib Is applicable for LIBDEF statements with the type PHASE, DUMP or * only. It specifies the library/sublibrary into which the linkage editor or DUMP output is to be cataloged. There is no system default.

TEMP|PERM Specify the duration of the definition given in the statement.

Phase Chaining

The search chain for phases includes the system directory list (SDL), and is different for "$" and "non-$" phases, as follows:

- for "non-$" phases: SDL -- TEMP chain -- PERM chain -- IJSYSRS.SYSLIB.
- for "$" phases: SDL -- IJSYSRS.SYSLIB -- TEMP chain -- PERM chain.

Operation	Operands	Type
[//] LIBDROP	{type|*}[,SEARCH][,CATALOG][{,TEMP|PERM}]	JCC,JCS

The LIBDROP statement resets the library search and catalog definitions set up by one or more previous LIBDEF statements.

type Specifies the member type for which the search and catalog definitions are to be reset.

* Specifies all types except DUMP.

SEARCH Specifies that only the search chain is to be dropped.

CATALOG Specifies that only the sublibrary defined in the CATALOG operand of a previous LIBDEF statement is to be dropped.

TEMP|PERM Specify whether the TEMP or PERM definition is to be dropped. The system default is TEMP.

Operation	Operands	Type
[//] LIBLIST	{type|*}[,BG|Fn|*][,SYSLST|SYSLOG]	JCC,JCS

The LIBLIST statement causes the library definitions set up with the LIBDEF statement to be displayed on SYSLOG or SYSLST.

type Specifies the member type for which the library definitions are to be displayed.

Causes the library definitions of all LIBDEF statements to be displayed, except DUMP.

BG|Fn|*	Specify the partition for which the current library definitions are to be displayed. * means all partitions. If the partition operand is omitted, the partition in which the command itself is processed, is assumed.
SYSLST|SYSLOG	Specify the output device to be used for displaying the library definitions.

Operation	Operands	Type
[//] LISTIO	listtype	JCS,JCC

The LISTIO command or statement (list I/O assignment) causes the system to print a listing of I/O assignments.

listtype can be one of the following:

ALL	Lists the physical units assigned to all logical units
ASSGN	Lists the physical units assigned to all system and programmer logical units of the partition from which the command is issued.
BG	Lists the physical units assigned to all logical units of the background partition.
cuu	Lists the logical units assigned to the specified physical unit.
DOWN	Lists all physical units specified as inoperative.
Fn	Lists the physical units assigned to all logical units of the specified foreground partition.
NPGR	Lists the number of programmer logical units allocated to each partition.
PROG	Lists the physical units assigned to all programmer logical units of the partition from which the command is issued.
SYS	Lists the physical units assigned to all system logical units of the partition from which the command is issued.
SYSxxx	Lists the physical units assigned to the specified logical unit of the partition from which the command is issued.
UA	Lists all physical units not currently assigned to a logical unit.
UNITS	Lists the logical units assigned to all physical units.

Operation	Operands	Type
LOG	none	JCC,AR

The LOG command causes logging of job control commands and statements on SYSLOG for that partition, where the command has been issued. (effective until NOLOG command) The AR LOG affects all partitions.
The LOG command suppresses OPTION ACANCEL.

Operation	Operands	Type
LUCB	cuu,phasename[,FOLD][,NOCHK] [,TRAIN=xxxxxx][,NULMSG]	AR

The LUCB command causes the system to load the buffer image, contained in the named phase, into the universal character set buffer (UCB) of the specified printer.

cuu	Channel and unit number of the printer whose UCB is to be loaded.
phasename	Name of the phase which contains the applicable buffer load image.
FOLD	Causes lower case characters to be printed as upper case characters.
NOCHK	Suppresses data checks from mismatches between printline characters and the UCB.
TRAIN=xxxxxx	Indicates that the print train identified by xxxxxx is to be mounted on the printer.
NULMSG	Suppresses printing of buffer load verification message.

Operation	Operands	Type
MAP	none	JCC,AR

The MAP command produces, on SYSLOG, a map of all storage areas in the system, with their sizes and starting addresses.

Operation	Operands	Type
MODE	{IR| CR| CE,cuu[,I[,xx,y]|,D[,xx,y]|,N}| R| STATUS| HIR{[,R|,Q][,E=eeee][,T=tttt]}| ECC{,M|,C}[,R|,Q|,TH][,E=eeee][,T=tttt]}	AR

The MODE command allows you to alter the recording mode.

In a VM/370 environment the R, HIR, and ECC operands are not accepted (Message 1I94I COMMAND IGNORED IN VM/370 ENVIRONMENT) VM/370 ENVIRONMENT to be issued.

For 4300 Processors, only operands IR, CR, and CE are valid.

For Models 135/138, the only valid MODE commands are:

```
    MODE CE,...
    MODE STATUS
    MODE ECC,Q
    MODE ECC,R
```

The meanings of the operands are:

IR|CR Recording mode for nonstandard labeled and unlabeled tape. (IR) Individual Recording for recording and then resetting the tape error statistics at each tape OPEN. (CR) Combined Recording for accumulating all statistics from nonstandard labeled and unlabeled tape on a specific tape unit until a standard labeled tape is opened.

CE, cuu The recording mode for device cuu may be reset. The possible recording modes are:

Normal. The default, normal, is assumed.
I Intensive. Normal recording continues.
D Diagnostic. Normal recording continues.
N No recording.

STATUS A report is printed on SYSLOG.
HIR Hardware Instruction Retry.
ECC Error Correction Code.
R Recording Mode

MODE R - places both HIR and ECC in recording mode.

MODE HIR,R - places HIR in recording mode.

MODE ECC,R (Models 155-II/158, and 3031) places ECC in recording mode. (if HIR is in recording mode)

MODE ECC,M,R (Models 145/148) places processor storage in recording mode. (if HIR is in recording mode)

MODE ECC,C,R (Models 145/148) places control storage in recording mode. (if HIR is in recording mode)

Q Quiet Mode

MODE HIR,Q - places both HIR and ECC in quiet mode.

MODE ECC,Q (Models 135/138, 155-II/158, and 3031) - places ECC in quiet mode.

MODE ECC,M,Q (Models 145/148) - places real storage in quiet mode.

MODE ECC,C,Q (Models 145/148) - places control storage in quiet mode.

M or C (only valid for Models 145/148)
M indicates real storage and C control storage.

TH Threshold Mode: (only valid for Models 145/148) Control storage is placed in quiet mode on next ECC control storage error. If TH is specified, T = tttt must also be specified.

E=eeee E: 8 (IBM-supplied value) through 9999
T=tttt T: 8 (IBM-supplied value) through 9999

Operation	Operands	Type
MSECS	[n]	JCC,AR

The MSECS command displays or changes the time slice for partition balancing.

n time slice in milliseconds (100 to 10000)
If n is omitted, the system displays the current time slice in milliseconds.

Operation	Operands	Type
MSG	{BG|Fn}	AR

The MSG command transfers control to an operator communications routine for which linkage has been established via a STXIT macro.

BG Communication with background partition is desired.
Fn Indicates the desired foreground partition.

Operation	Operands	Type
[//] MTC	opcode,{cuu|SYSxxx}[,nn]	JCC,JCS

The MTC command or statement controls magnetic tape operations.

opcode Specifies the operation to be performed.

BSF	Backspace File
BSR	Backspace Record
DSE	Data Security Erase (3400 only)
ERG	Erase Gap
FSF	Forward Space File
FSR	Forward Space Record
REW	Rewind
RUN	Rewind and Unload
WTM	Write Tape Mark

SYSxxx Logical unit to which the tape is assigned.
cuu Channel and unit number.
nn Decimal number (1 through 99) of times.
The default is 1.

Operation	Operands	Type
NEWVOL	[BG|Fn][,IGNORE]	AR

Indicates that a new volume has to be mounted for the specified partition.

BG|Fn Indicates the partition for which the new volume was mounted. If no operand is specified, BG is assumed.
IGNORE Specifies that the mount request is to be ignored.

Operation	Operands	Type
NOLOG	none	JCC,AR

The NOLOG command (suppress logging) terminates the listing, on SYSLOG, of job control commands and statements (except ALLOC, ALLOCR, DVCUP, HOLD, IGNORE, JOB, MAP, PAUSE, PRTY, SIZE, STOP, UNBATCH /*, /& and /+) that occur in the partition in which the NOLOG is issued.

Operation	Operands	Type
NPGR	[BG=m][,Fn=m]	JCC

The NPGR command defines the number of programmer logical units which may be allocated in a given partition.

BG|Fn Partition for which the number (m) of programmer logical units to be allocated.
m Decimal number from 10 through 255.

30 programmer logical units per partition is the default.

Operation	Operands	Type

[//] ON	{$RC comparator n} [OR ...] {GOTO label}
	{$CANCEL } [| ...] {CONTinue }
	{$ABEND } [AND ...]
	[& ...]
		JCC,JCS

The ON statement is a global conditional function. During execution of a job in which an ON statement occurs, the specified condition is tested at the end of each job step following the ON statement. If the condition is true, the specified action is taken, otherwise processing continues with the next statement.

$RC Specifies the return code of the preceding step.
comparator Specifies a comparison to be done. This can be one of the following six possibilities:

Equal	specified as	= or EQ
Not equal	specified as	= or NE
Greater than	specified as	> or GT
Less than	specified as	< or LT
Greater or equal	specified as	>= or GE
Less or equal	specified as	<= or LE

n A decimal integer from 0 to 4095, to be used for comparison with the return codes.
$CANCEL Specifies that the action is to be taken if the CANCEL command is given for the job.
$ABEND Action is to be taken if the step terminates abnormally.
GOTO label Label where processing is to continue.
CONTINUE Processing should continue if the specified condition is true.

The following are default ON-conditions.
ON $RC<16 CONTINUE ON $RC>=16 GOTO $EOJ ON $ABEND GOTO $EOJ ON $CANCEL GOTO $EOJ

Operation	Operands	Type
ONLINE	cuu	AR

The ONLINE command is used to simulate a 'device ready' status for a device.

cuu Channel and unit number, in hex, of the particular device.

Operation	Operands	Type
// OPTION	option [,option]...	JCS

The OPTION statement specifies one or more job control options which temporarily override the system defaults.

The options, which can appear in any order, are as follows:

ACANCEL The job must be canceled if an attempt to assign a device is unsuccessful.

NOACANCEL The system awaits operator intervention in the case of an unsuccessful assignment.

ALIGN The assembler aligns constants and data areas on proper boundaries.

NOALIGN Suppresses the ALIGN option.

CATAL A phase or program is permanently cataloged in a library at the completion of a link-edit run. CATAL also sets the LINK option.

DECK Language translators produce object modules on SYSPCH.

NODECK Suppresses the DECK option.

DUMP Dumps the registers, supervisor area, partition, the used part of the system GETVIS area, the SVA phase in error (if the error occurred in the SVA), and the phase load list.

PARTDUMP Dumps selected areas of storage.

NODUMP Suppresses the DUMP or PARTDUMP option.

EDECK The assembler punches all valid source macro definitions in edited format on SYSPCH.

NOEDECK Suppresses the EDECK option.

ERRS The FORTRAN, DOS/VS COBOL, and PL/I compilers summarize all errors in the source program on SYSLST.

NOERRS Suppresses the ERRS option.

JCANCEL System should skip to End-of-Job if a job control error occurs.

NOJCANCEL Suppresses the JCANCEL option, and is the system default.

LINK Indicates that the object module is to be link-edited. When the LINK option is used.

NOLINK Suppresses the LINK option.

LIST Language translators write the source module listing on SYSLST.

NOLIST Suppresses the LIST option.

LISTX The COBOL compiler produces a PROCEDURE DIVISION map on SYSLST. The PL/I and FORTRAN compilers produce the object modules on SYSLST.

NOLISTX Suppresses the LISTX option.

LOG Lists columns 1-80 of all control statements and commands on SYSLST.

NOLOG Suppresses the listing of all valid control statements and commands on SYSLST until a LOG option is encountered.

LOGSRC This operand causes JC statements which contain symbolic parameters to be printed twice, once in source form (as coded), once with substituted symbolic parameters (as processed by job control.)

NOLOGSRC Suppresses LOGSRC, and is the system default. JC statements will be printed only once showing the substitution of the symbolic parameters, if any.

PARSTD All disk, diskette, and tape label statements are written into the partition standard subarea of the system's label information area.

PARSTD=ADD All label information will be stored permanently into the partition standard subarea of the label information area without overwriting existing information.

PARSTD=DELETE This option must be followed by the filename(s) of the DLBL statement(s) to be deleted from the partition standard subarea of the label information area. The last (or only) filename must be followed by /*. PARSTD (or STDLABEL)=DELETE must be the last option of the OPTION statement.

PARSTD=Fn All label information will be stored permanently into the specified partitions standard subarea of the label information area. The option can only be submitted in the background, and the partition specified by Fn must be inactive.

RLD The assembler writes the relocation list dictionary on SYSLST. This option is suppressed if NOLIST is specified.

NORLD Suppresses the RLD option.

STDLABEL All disk, diskette, and tape labels are written into the system standard subarea of the label information area, to be available to all subsequent jobs in all partitions until another STDLABEL option without operand or with =DELETE is submitted.

STDLABEL=ADD All label information will be stored permanently into the system standard subarea of the label information area without overwriting existing information.

STDLABEL=DELETE This option must be followed by the filename(s) of the DLBL statement(s) to be deleted from the system standard subarea of the label information area. STDLABEL (or PARSTD) with =DELETE must be specified as the last option of the OPTION statement.

SUBLIB=DF Directs the assembler and ESERV program to retrieve non-edited macros and copy-books from sublibrary members of type D instead of from sublibrary members of type A, and to retrieve edited macros from sublibrary or type F instead of from sublibrary members of type E.

SUBLIB=AE Redirects the assembler and the ESERV program to retrieve non-edited macros and copy books from sublibrary members of type A and to retrieve edited macros from sublibrary members of type E.

SYM The COBOL compiler produces a DATA DIVISION map on SYSLST; the PL/I compiler produces the symbol table on SYSLST.

NOSYM Suppresses the SYM option.

SYSDUMP Indicates that dumps are to be written to the dump sublibrary which is active for the partition. The old form of this operand (SYSDMP) is accepted for compatibility reasons.

NOSYSDUMP Indicates that dumps are to be written on SYSLST. The old form of this operand (NOSYSDMP) is accepted for compatibility reasons.

SYSPARM='string' Specifies a value for the assembler system variable symbol &SYSPARM.

TERM Error messages are written on SYSLOG (applies only to compilers that support this function).

NOTERM Suppresses the TERM option.

USRLABEL ALL disk, diskette, and tape labels are written temporarily (for one job or job step) into the partition temporary sub-area of the label information area.

XREF The assembler writes the symbol cross-reference list on SYSLST.

SXREF The assembler writes the symbol cross-reference list on SYSLST; printing of all unreferenced labels is suppressed.

NOXREF Suppresses the XREF or SXREF option.

NOFASTTR Suppresses fast CCW translation for the current job. (Note that FASTTR is a system generation option only.)

48C Specifies the 48-character set on SYSIPT (for PL/I).

60C Specifies the 60-character set on SYSIPT (for PL/I).

Operation	Operands	Type
[//] OVEND	[comment]	JCS,JCC

The OVEND (Override End) statement or command indicates the end overriding statements for cataloged procedures. The OVEND statement has no operand.

Operation	Operands	Type
[//] PAUSE	[any user comment]	JCS,JCC
PAUSE	[BG|Fn][,EOJ]	AR

The PAUSE statement causes a pause immediately after processing this statement.

The PAUSE command causes a pause at the end of the current job step.

The PAUSE statement or command always appears on SYSLOG. If SYSLOG is assigned to a line printer, the PAUSE statement or command is ignored.

BG or Fn Indicates the partition in which processing is to be interrupted.
EOJ Indicates that the interruption will occur at the end of the current job.

Operation	Operands	Type
[//] PROC	[parname=[value]][,...]	JCS,JCC

The PROC command or statement, when used, is the first line of a cataloged procedure. It is required only when the procedure contains symbolic parameters to which you want to assign initial values.

parname The name of the symbolic parameter (without a leading &) to which you want to assign the specified value.
value The value you want to assign to the specified symbolic parameter.

Operation	Operands	Type
PRTY	none	AR
PRTY	partition,partition[,partition]...	AR,JCC
PRTY	partition=partition[=partition]...	AR,JCC

The AR PRTY (priority) command allows the operator to display or change the priority sequence of partitions in the system. In both cases also the current status (if active) of the TP Balancing (TPBAL) function is displayed.

The JCC PRTY command can be used only in the BG during ASI (Automated System Initialization) to modify the priority sequence of the partitions in the system.

The AR PRTY command without operands displays, on SYSLOG, the current priorities of all partitions.

The operands can be specified in two forms, as shown, or in a mixed form, to provide for priority setting and partition balancing together.

partition,partition,partition ...
specifies the desired sequence of processing priority.
partition=partition=partition ...
specifies that dynamic partition balancing is to be used for the partitions which you list with a separating equals sign(=).
Mixed format: Specifies a desired sequence and in addition partition balancing.
example: PRTY BG,F1=F2=F3=F4,F5,F6

Operation	Operands	Type
[//] PWR	{PRELEASE...|PHOLD...}	JCS,JCC

The PWR job control statement makes it possible to pass the commands PRELEASE and PHOLD to POWER at any point in the job stream. The operand of the PWR statement is taken as a POWER command, and its syntax is checked by the POWER routine.

PWR specifies that the rest of the statement is a POWER command.
PRELEASE|PHOLD are the POWER commands which will be accepted.

Operation	Operands	Type
RC	none	AR

With asynchronous operator communication, the operator can use the RC (Request Communication) command to enter an AR command when the attention routine is not available (the attention routine identifier AR does not appear).

Operation	Operands	Type
REPLID	none	AR

The REPLID command displays the reply-IDs (and partition indicators) of all messages for which replies are still pending.

Operation	Operands	Type
RESERV	cuu	AR

The RESERV command reserves a device for VSAM space management usage. The reserved status can be reset only by a FREE command.

cuu	Channel and unit number of the device to be reserved.

Operation	Operands	Type
[//] RESET	{SYS|PROG|ALL|SYSxxx}	JCC,JCS

The RESET command or statement resets temporary LIBDEFs and I/O assignments to their permanent values in the partition in which RESET was submitted.

SYS	Resets all system logical unit assignments and library search chain definitions to their permanent values.
PROG	Resets all programmer logical units to their permanent assignments.
ALL	Resets all logical unit assignments and library chain definitions to their permanent values.
SYSxxx	Resets the specified logical unit to its permanent assignment. SYSIN or SYSOUT cannot be specified.

Operation	Operands	Type
ROD	none	JCC

The ROD (Record on Demand) command records all statistical data record counters for all non-telecommunication devices on the recorder file on SYSREC.

Operation	Operands	Type
// RSTRT	SYSxxx,nnnn[,filename]	JCS

The RSTRT (Restart Checkpointed Program) statement is available for checkpointed programs.

SYSxxx	Logical unit name of the device on which the checkpoint file is stored.
nnnn	Identification of the checkpoint record to be used for restarting.
filename	The name of the disk checkpoint file to be used for restarting.

Operation	Operands	Type
SET	[UPSI=config][,LINECT=count][,RCLST=number] [,RCPCH=number][,RF={yes|create}] [,DATE=date] [,HC={yes|no|create}][,SDL]	JCC

The SET command sets controls for the execution of programs. Except for SET UPSI,

UPSI=config Sets the bit configuration of the UPSI byte in the communications region. Specify one to eight characters, either 0, 1, or X.

LINECT=count Sets the standard number of lines to be printed on each page of SYSLST. Specify an integer between 30 and 99.

RCLST=number Decimal number indicating the minimum number of SYSLST disk records remaining to be written before operator warning. It may be any decimal number from 100 through 65535.
Default is: RCLST = 1000.

RCPCH=number Decimal number indicating the minimum number of SYSPCH disk records remaining to be written before operator warning. It may be any decimal number from 100 through 65535.
Default is: RCPCH = 1000.

RF=YES|CREATE Defines the status of the recorder file (IJSYSRC) on SYSREC.

YES (default) Indicates that an active recorder file exists. The system opens this file when the first JOB statement is encountered.

CREATE Instructs the system to create a recorder file when the first JOB statement is encountered.

DATE=date Sets the system date permanently to the specified value.

date can have the following formats:
mm/dd/yy or dd/mm/yy
mm = month (01-12); dd = day (01-31); yy = year (00-99).
The first format is the system default (can be changed by using the STDOPT command)

HC=YES|NO|CREATE Defines the status of the hard-copy file (IJSYSCN) on SYSREC.

YES Indicates that a hard-copy file exists in the system, and that it is to be opened. YES is the default.

NO Indicates that no recording is to be performed on the hard-copy file. (only if a console printer is attached)

CREATE Instructs the system to create a hard-copy file.

SDL Indicates that phase names are to be added to the system directory list and, optionally, that phases are to be loaded into the SVA.
If the requested phase is not present in IJSYSRS.SYSLIB, a dummy entry is created until the specific phase is cataloged.

Operation	Operands	Type
SETDF	{3800|cuu}[,BURST=[Y|N]][,CHARS=[table-name]] [,FCB=[fcb-name]] [,FLASH=[overlay-name]] [,FORMS=[forms-name]] [,LIST] [,MODIFY=[copymod-name]] [,RESET]	AR

The SETDF command allows to set and/or reset default values for the IBM 3800 Printing Subsystem or to display the default values.

3800 All 3800 printers will be set with the specified default values of SETDF.

cuu Channel and unit number of the 3800 whose default values are to be set or displayed by SETDF.

BURST=, No change in the threading of the forms is requested.
Y. Specifies that the printed output is to be burst into separate sheets with the edges trimmed.
N. Specifies that the printed output is to be in continuous fanfold mode. BURST=N is the default.

CHARS=, The default for the character arrangement table is reset to the hardware default Gothic-10 folded table.

table-name specifies the 1- to 4-character suffix of the name of the default character arrangement table.

FCB=, The default for the forms control buffer is reset to the hardware default FCB.

fcb-name specifies the 1- to 4-character suffix of the name of the default FCB.

FLASH=, No flashing is done.

overlay-name specifies the 1- to 4-character name of the forms overlay frame to be used as the default.

FORMS=, The operator is requested to load the forms named STANDARD when the default is needed.

forms name specifies the 1- to 4-character name of the forms to be used.

LIST The established default settings are to be displayed at the operator console.

MODIFY=, No copy modification is done.

copymod-name specifies the 1- to 4-character suffix of the modification phase name.

RESET Sets all keywords to the hardware defaults.

Operation	Operands	Type
SETMOD	cuu[,mode]	AR

The SETMOD (set mode) command, (valid for 8809 Magnetic Tape Unit only) can be used to adjust the tape speed to the actual I/O traffic.

cuu Specifies the channel and unit number of the 8809.

mode Can be one of the following:

90 (or HL) = High Speed, Long Gap (Streaming)
30 (or HS) = High Speed, Short Gap (Streaming)
50 (or LL) = Low Speed, Long Gap (Start-Stop)
60 (or LS) = Low Speed, Short Gap (Start-Stop)

60 (or LS) is the default.

Operation	Operands	Type
[//] SETPARM	pname=[{value|$RC\|$MRC}][,...]	JCC,JCS

The SETPARM statement enables you to define a symbolic parameter and/or assign a value to it.

pname Name of the symbolic parameter to be defined - 1 to 7 alphameric characters. (first must be alphabetic)

value Character string of up to 50 characters. A string containing national or special characters, must be enclosed in quotes.

$RC Return code of the last job step which was executed.

$MRC Specifies the maximum return code of all preceding job steps.

Operation	Operands	Type
[//] SETPRT	SYSxxx[,BURST={N|Y|*}] [,CHARS={table-name|*|(table name,...)}] [,COPIES=number][,DCHK={B|U}] [,DEBUG={NORM|TERM|DUMP|TRAC}] [,DFLT={N|Y}] [,FCB={fcb-name|*(fcb-name,V)|(*,V)}] [,FLASH={overlay-name[,count]|(,count)| (*[,count]} [,FORMS={forms-name|*}][INIT={N|Y}] [,MODIFY={copymod-name|*| (copymod-name, table name)}] [,SEP=O][,TRC={N|Y}]	JCC,JCS

The SETPRT (set printer) job control statement or command sets user-specified control values for the IBM 3800 Printing Subsystem.
At least one of the optional operands must be specified.

SYSxxx Logical unit identifier for the 3800 printer to be set up. This operand is always required.

BURST= If the operand is omitted, no change to the threading is requested.

Y specifies that the operator should thread the forms through the Burst-Trimmer-Stacker.

N specifies that the operator should thread the forms to the continuous forms stacker.

* specifies that the system default BURST setting is requested.

CHARS= If omitted, the character arrangement table is not changed unless INIT=Y is coded.

table-name specifies the 1- to 4-character suffix of the character arrangement table name.

(table-name,...) specifies up to four names, separated by commas and enclosed in parentheses.

* specifies that the system default character arrangement table is requested.

COPIES= If the operand is omitted, the number of copies is not changed unless INIT=Y is coded.

n (value from 1 to 255) specifies the number of copies of each page to be reproduced before printing the next page.

DCHK= If the operand is omitted, data checks are blocked.

B specifies that data checks are to be blocked.

U specifies that data checks are allowed.

DEBUG= NORM sets a return code in register 15 and returns to the caller on any exit from the SETPRT routines.

TERM sets a return code in register 15 and cancels the activity for return codes higher than 4.

DUMP sets a return code in register 15 and cancels the job with a dump, for a return code higher than 4.

TRAC dynamically traces, on SYSLST, the activity of the SETPRT routines and then cancels the job with a dump if the SETPRT return code is greater than 4. Tracing requires 12K of GETVIS space.

DFLT= N is the default specification for this keyword and does not establish 3800 default setup.
Y specifies that the printer is to be set with the defaults that were specified by the operator in the SETDF command.

FCB= If the FCB operand is omitted, the FCB is not changed unless INIT=Y is coded.
fcb-name specifies the 1- to 4-character suffix of the name of the FCB.

V requests FCB verification.

* specifies that the system default FCB is requested.

FLASH= overlay-name is the 1- to 4-character name of the forms overlay frame.

count is the number (from 0 to 255) of copies to be flashed with the overlay.
If no count is specified, all copies are flashed.

* requests the system default forms overlay.

FORMS= forms-name is the 1- to 4-character forms identifier.

* requests the system default forms.

INIT= Y specifies that the printer be reset to hardware defaults.

N is the default and does not reset the 3800 to hardware defaults.

MODIFY= copymod-name specifies the 1- to 4-character suffix of the modification phase name.

table-name specifies the 1- to 4-character name of the character arrangement table to be used when the 3800 prints the copy modification text.

* requests the system default copy modification.

SEP= 0 indicates that, if the burster-trimmer-stacker is being used, the 3800 should offset-stack the pages that follow from the pages that were previously transmitted. If the continuous forms stacker is being used, the 3800 changes the marking on the perforation edge from one line to two lines or vice versa.

TRC= N indicates that, for any DTFPR or DTFDI operand after this SETPRT, data lines do not contain table reference characters unless specified in the DTF macro.

Y indicates that the first character of each output data line (after the optional print control character) given to the access method is a table reference character.

<table>
<tr><th>Operation</th><th>Operands</th><th>Type</th></tr>
<tr><td>SIZE</td><td>partition=<nK|mM>[,partition=<nK|mM&gt;]...</td><td>JCC,AR</td></tr>
</table>

The SIZE command is used to specify the amount of contiguous virtual storage in a partition which is reserved for program execution. The rest of the partition is available as partition GETVIS area.

partition Partition (BG, F1, F2, ...) for which storage is to be reserved.

nK|mM Amount of storage to be reserved in kilobytes (nK) or megabytes (mM). The remainder of the partition is available as partition GETVIS area.
n should be a multiple of 4.

<table>
<tr><th>Operation</th><th>Operands</th><th>Type</th></tr>
<tr><td>START</td><td>[BG|Fn]</td><td>AR</td></tr>
<tr><td>START</td><td>Fn</td><td>JCC</td></tr>
</table>

The AR START (Start or Continue Processing) command activates or continues processing in the specified partition. The function of the START command is exactly the same as that of the BATCH command.

BG Indicates that the background partition is to be reactivated.

Fn Specified foreground partition is to be activated, or restarted after a STOP command.

Operation	Operands	Type
[//] STDOPT	option[,option]...	JCC,JCS

The STDOPT (standard options) command or statement sets or resets the permanent job control options which were established at system initialization (system defaults).
The options, which can appear in any order, are as follows (the first specification is always the system default value):

ACANCEL=NO|YES Specifies whether job control is to cancel jobs automatically (ACANCEL=YES) or to wait for operator intervention (ACANCEL=NO) after an unsuccessful attempt to assign a device.

ALIGN=YES|NO Specifies whether the assembler is to align data on halfword or fullword boundaries, according to the type of instruction used.

CHARSET=48C|60C Specifies either the 48- or 60-character set for PL/I translator input on SYSIPT.

DATE=MDY|DMY Specifies the format of the date: MDY=month/day/year DMY=day/month/year.

DECK=YES|NO Specifies whether or not language translators are to produce object modules on SYSPCH.

DUMP=YES|NO|PART Specifies whether or not a dump of the registers and virtual storage is to be taken in the case of an abnormal program end. PART specifies that a dump of the major suoervisor control blocks and the virtual storage of the partition is to be taken.

EDECK=NO|YES Specifies if the assembler is to create and punch edited macros on SYSPCH.

ERRS=YES|NO Specifies whether or not language translators are to summarize all errors in source programs on SYSLST. Assembler and PL/I always assume ERRS=YES.

FASTTR=NO Specifies that fast CCW translation is to be switched off for all partitions. This operand overrides the FASTTR=YES supervisor option.
When you have submitted the STDOPT command or statement with the FASTTR=NO operand, only a new IPL can re-activate fast CCW translation.

HCTRAN=YES|NO Specifies whether the output from PRINTLOG and LISTLOG is to be translated to all upper case (YES) or in mixed upper and lower case.

JCANCEL=NO|YES Specifies whether or not the system should terminate the job abnormally when a job control error occurs (JCANCEL=YES), or wait for operator intervention (JCANCEL=NO).

LINES=56|nn Specifies the number of lines per page on SYSLST. The minimum is 30, the maximum is 99.

LIST=YES|NO Specifies whether or not language translators are to write source module listings and diagnostics to SYSLST.

LISTX=NO|YES Specifies whether or not language translators are to write hexadecimal object module listings on SYSLST.

LOG=YES|NO Specifies whether or not all job control statements are to be listed on SYSLST.

RLD=NO|YES Specifies whether or not the relocation dictionary information is to be printed.

SXREF=NO|YES Specifies whether the assembler is to print short cross-reference lists on SYSLST. The printing of unreferenced labels is suppressed. Do not specify SXREF=YES together with XREF=YES.

SYM=NO|YES SYM=YES specifies that the PL/I compiler is to produce a symbol and offset table listing on SYSLST, or that the COBOL compiler is to produce a data division glossary.

SYSDUMP=NO|YES YES indicates that dumps are to be written to the dump sublibrary which has been defined for the partition via a LIBDEF DUMP command. SYSDUMP=NO specifies that dumps are to **be** written to SYSLST. For compatibility reasons, the keyword **may** be entered as SYSDMP.

TERM=NO|YES Specifies whether messages from a compiler are to be displayed on SYSLOG.

XREF=YES|NO XREF=YES specifies that the assembler is to write symbolic cross-reference lists on SYSLST, or that American National Standard COBOL is to produce a cross-reference listing.

Operation	Operands	Type
STOP	none	JCC

The STOP (Stop Processing) command indicates that there are no more jobs to be executed in the partition in which the command is given.
This command removes the partition from the system's task selection mechanism, but the partition remains active. Job control remains in the partition and can be restarted by the START or BATCH attention routine command.

Operation	Operands	Type
// TLBL	filename,['file-id'],[date], [file-serial-number], [volume-sequence-number], [file-sequence-number], [generation-number], [version-number] [,DISP=NEW|OLD|MOD]	JCS (for EBCDIC files)
// TLBL	filename,['file-id'],[date], [set-identifier], [file-section-number], [file-sequence-number], [generation-number], [version-number] [,DISP=NEW|OLD|MOD]	JCS (for ASCII files)

The TLBL (Tape Label Information) statement contains file label information for the checking and writing of tape labels.

filename One to seven alphameric characters, the first of which must be alphabetic.

'file-id' One to seventeen alphameric characters, contained within quotes.

date Output files: Retention period in days written as a decimal number (0-9999) or the expiration date in the format 19yy/ddd or 20yy/ddd (yy = year, ddd = day.) Format yy/ddd is also accepted. (85/032 is interpreted as 1985/032.

Input files: Creation date of the file in the format: 19yy/ddd or 20yy/ddd or yy/ddd (yy=00-99, ddd=1-365).

file-serial-number (EBCDIC) / set identifier (ASCII)

One to six alphameric characters indicating the volume serial number of the first (or only) reel of the file.

volume-sequence-number (EBCDIC) / file-section-number (ASCII)

A one to four-digit decimal number specifying the volume of a multi-volume file at which you wish to start processing.

file-sequence-number A one to four-digit decimal number specifying the file of a multi-file volume at which you wish to start processing.

generation-number A one to four-digit number specifying the generation number of the file to be processed.

version-number A one or two-digit decimal number specifying the version number of the file to be processed.

DISP=NEW|OLD|MOD This operand specifies whether a new output file is to be created or an existing file extended. The specifications have the following meanings:

NEW - Specifies that the file is to be created. This is the default.

OLD - Specifies that the file already exists and is to be extended.

MOD - Specifies conditional extension or creation of the file.

Operation	Operands	Type
TPBAL	[n]	AR

The TPBAL (teleprocessing balancing) command allows the operator to change or display on SYSLOG the status of the TP balancing function.

n specifies the number of partitions in which processing can be delayed. (number of partitions in the system minus one.)

Operation	Operands	Type
UCS	SYSxxx,phasename[,FOLD] [,BLOCK][,NULMSG]	JCC

The UCS (load universal character set buffer) command causes the 240-character universal character set contained in the phase specified by phasename to be loaded as buffer storage in the IBM 2821 Control Unit.

SYSxxx, The logical unit assigned to a 1403 printer with UCS feature.

phasename Name of the phase to be loaded, followed by an 80-character verification message.

FOLD Enables printer to print lower case bit configurations as upper case characters.

BLOCK permits unprintable characters to be printed as a blanks without causing a data check stop.

NULMSG Suppresses the 80-character verification message to be printed after the buffer is loaded.

Operation	Operands	Type
UNBATCH	none	JCC

The UNBATCH command terminates foreground processing and releases the partition (making it inactive).

Operation	Operands	Type
UNLOCK	SYSTEM=sys-id	AR

The UNLOCK command is used to release all resources locked by the specified system.

sys-id Specifies the CPU-ID of the CPU which has become inoperative. The command will release all locks belonging to the named system.

Operation	Operands	Type
// UPSI	string	JCS

The UPSI (user program switch indicators) statement allows you to set program switches that can be tested by applications during execution.

string is a string of one to eight characters, which correspond to the bit positions of the UPSI byte in the communication region. The specified character string must consist of 0, 1 and X.

Operation	Operands	Type
VOLUME	[c|cu|cuu]	AR

The VOLUME command provides the operator with a short summary of the volumes mounted on disk devices, together with an indication of whether or not a volume is in use, shared by another system, or reserved for VSAM space management usage.

c	Channel address. Information for all disks on specified channel.
cu	Channel and control unit address. Information for all devices on specified channel and control unit.
cuu	Device address. Information for the specified device only.

Operation	Operands	Type
// ZONE	{EAST|WEST}/hh/mm	JCS

The ZONE statement defines the time difference between local time and Greenwich mean time.

EAST	A geographical position east of Greenwich.
WEST	A geographical position west of Greenwich.
hh/mm	A decimal value that indicates the difference in hours (00 to 23) and minutes (00 to 59) between local time and Greenwich Mean Time.

Operation	Operands	Type
/.	label	JCC,JCS

The label statement defines a point in the job stream up to which you may want to skip JC statements using a GOTO statement or the GOTO action of an ON statement. When a GOTO is raised, processing continues at the JC statement following the /. label statement specified.

Column 1 contains a slash (/) and column 2 a period (.). Column 3 must be blank.

label is a name consisting of one to eight alphanumeric characters. The first character must be alphabetic. Symbolic parameters are not allowed in this statement.

Operation	Operands	Type
/+	[comments]	JCS

The /+ statement marks the end of a job control procedure. It must be included as the last statement when a procedure is cataloged.

Operation	Operands	Type
/*	[comments]	JCS

The end-of-data file statement must be the last statement of each input data file on SYSRDR and SYSIPT.

Operation	Operands	Type
*	[comments]	JCS

The content of the comment statement is printed on SYSLOG. If followed by a PAUSE statement, the statement can be used to request operator action. Column 1 contains an asterisk, column 2 must be blank.

Operation	Operands	Type
/&	[comments]	JCS

The end-of-job statement must be the last statement of each job. Any comments, beginning in column 35, are printed at end of job.

LINKAGE EDITOR CONTROL STATEMENTS

Operation	Operands
ACTION	[,MAP|,NOMAP][,NOAUTO] [,CANCEL][,SMAP]

This statement is used to indicate linkage editor options. The statement must be the first linkage editor statement in the input stream.
At least one blank must precede ACTION.

MAP Requests the linkage editor to write to SYSLST a map of virtual storage, which can be used for problem determination.
If the MAP operand is specified, SYSLST must be assigned.

NOMAP Indicates that the MAP option should not take effect.

NOAUTO Indicates that the AUTOLINK function is to be suppressed for the present linkage editor run.

CANCEL Cancels the job automatically if any of messages the 2100I through 2170I occur.

SMAP Indicates that, in addition to the standard virtual storage map in which the control sections are ordered by load address, a listing of the CSECT names ordered alphabetically is also generated.

Operation	Operands
ENTRY	[entrypoint]

Every program, as input for the linkage editor, is terminated by an ENTRY statement.
At least one blank must precede ENTRY.

entrypoint Specifies the name (label) of an entry point. It must be the name of a CSECT or a label definition (source ENTRY) defined in the first phase. This address is used as the transfer address to the first phase in the program.

Operation	Operands
INCLUDE	[modulename][,(namelist)]

INCLUDE indicates that an object module is to be included for editing by the linkage editor.
At least one blank must precede INCLUDE.

If both operands are omitted, the object module to be included is assumed to be on SYSIPT. Job control copies it from there to SYSLNK. INCLUDE statements with no operands are recognized only on SYSRDR.

modulename (1 to 8 alphameric characters) Specifies the name of the module, as used when cataloged in the sublibrary.

(namelist) The linkage editor generates a phase from only the control sections specified in this operand. The namelist is in the following format:

(csname1,csname2,...)

Operation	Operands
PHASE	name,origin[,NOAUTO][,SVA][,PBDY]

This statement provides the linkage editor with a phase name and an origin point for the phase.
At least one blank must precede PHASE.

name
Specifies the name of the phase. One to eight alphameric (0-9, A-Z, #, $, and @) characters are used as the phase name. The name may not be ALL, S, or ROOT.

origin
Specifies the load address of the phase. The load address can be in one of the following forms:

1. symbol[(phase)][+relocation]
2. *[+relocation]
3. S[+relocation]
4. ROOT
5. +displacement

Items 1 to 4 specify a relative address, item 5 an absolute address.

A phase can be made relocatable if its origin is specified as a relative address (formats 1-4 above). However, if the address is relative to another phase which is not relocatable, the new phase will not be relocatable. If the operand origin is not specified, the phase is made relocatable.

1. **symbol:** A previously defined phase name, control section name, or external label.

   (phase): If **symbol** is a previously defined control section name or a previously defined external label that appears in more than one phase, **phase** (in parentheses) directs the linkage editor to the phase that contains the origin.

   relocation: The origin of the phase currently being processed
   a + or a - followed by: X'hhhhhh' (1 to 6 hexadecimal digits);
   a + or a - followed by: dddddddd (1 to 8 decimal digits);
   a + or a - followed by: nK,(where n is the number of kilobytes).

2. *: For the first PHASE statement processed, the origin is to be the first doubleword storage address after the partition save area, or the area assigned to the COMMON pool (if any).

   relocation: Relocation of the phase relative to the next storage location of the virtual partition.

3. **S**: if **S** is specified, the origin is determined in the same manner as for the first PHASE statement.

   relocation: Relocation of the phase relative to the start of the virtual partition.

4. **ROOT**: Tells the linkage editor that the phase that follows is a root phase.
   Only the first PHASE statement in the linkage editor input can specify ROOT.

5. **displacement:** Allows the origin (loading address) to be set at a specified location.
   displacement must be:
   X'hhhhhh' (one to six hexadecimal digits),
   dddddddd (one to eight decimal digits), or
   nK (where n is the number of kilobytes.
   A displacement of zero (+0) denotes a self-relocating program.

NOAUTO
The Automatic Library Lookup (AUTOLINK) feature is suppressed for the current phase.

SVA
The phase is SVA-eligible. This means that the phase must be re-enterable and relocatable.

PBDY
The phase is to be link-edited on a page boundary.

SUMMARY OF LIBRARIAN COMMANDS

The following is a complete list of the VSE Librarian commands:

	Command Object		
Command Name	Library	Sublibrary	Member
ACCESS		X	
BACKUP	X	X	
CATALOG			X
CHANGE		X	
COMPARE	X	X	X *
CONNECT		X	
COPY	X	X	X *
DEFINE	X	X	
DELETE	X	X	X *
GOTO			
INPUT			
LIST			X *
LISTDIR	X	X	X *
MOVE	X	X	X *
ON			
PUNCH			X *
RELEASE	X	X	
RENAME		X	X *
RESTORE	X	X	X *
TEST	X	X	X
UPDATE			X
/. label			

'*' means: generic specification accepted.

LIBRARIAN COMMANDS

```
| Access  | {Sublib=l.s | ?}                                              |
```

The ACCESS command specifies the sublibrary, qualified by the library name, to be used in any following command which has a member_name.member_type specification as its operand.

If you code a question mark as operand, the name of the library and sublibrary currently accessed is displayed on SYSLOG, if the command was entered from there, otherwise on SYSLST.

```
| Backup  | {Lib    =l ...} Tape={SYSnnn+ [Restore={Online}   ] ..      |
|         | {Sublib=l.s ..}     {cuu    }          {Standalone}          |
|         |                                                               |
|         | ..[Include=Historyfile] [ID=name] [Header=l.s.mn.mt]          |
```

The BACKUP command causes libraries, sublibraries or SYSRES files to be copied to tape. (including reorganization)

Lib=l — Library to be backed up. The library names IJSYSR1 to IJSYSR9 specify SYSRES files. IJSYSRS is the IPL'ed system.

Sublib=l.s — Sublibrary to be backed up. You may code a list of sublibrary names.

Tape=SYSnnn|cuu — Programmer logical unit or physical unit address. (Tape unit to be used for output.)

Restore=Online|Standalone — Specify how the backed up data are to be restored. This operand is applicable only to libraries and SYSRES files. The default value is ONLINE.

Include=Historyfile — The system history file should also be backed up.

ID=name — Identification for the backup file to be created by this BACKUP command.

Header=l.s.mn.mt — Optional header to be written to the first file on the backup tape.

```
| CAtalog | mn.mt [Eod=xx] [Data={Yes}] [Replace={Yes}]                 |
|         |                      {No}             {No}                  |
```

The CATALOG command causes the data following it to be cataloged under the name and type specified.

Any member types (except PHASE and DUMP) can be cataloged into any sublibrary using this command.

mn.mt — Name and type under which the following data are to be cataloged. mn and mt may each be 1 to 8 characters long, and must be alphanumeric. The member type must not be PHASE or DUMP. These types can be cataloged only by the Linkage Editor and Dump program respectively.

Eod=xx — Combination of two characters to be used to indicate end-of-data in the following input. The default is /+.

Data=Yes|No — Applicable only for procedures. The member type must be PROC or a user type. You must code DATA=YES if the procedure contains SYSIPT data. The system default is DATA=NO.

Replace=Yes|No — Allows conditional cataloging. REPLACE=YES is default.

For compatibility with the old CATALS function, BKEND, MACRO and MEND are allowed as EOD statements when migrating to the VSE librarian. They can start in column 1.

```
| CHange  | Sublib=l.s ... [REUse={Automatic|Immediate}]                  |
```

The CHANGE command can be used to change the REUSE attribute of a sublibrary.

Sublib=1.s The qualified name of the sublibrary whose REUSE attribute is to be changed.

REUse=Automatic|Immediate Specifies which REUSE attribute the specified sublibrary (or sublibraries) should have from now on. (Automatic is the default)

When IMMEDIATE is specified, any space which is no longer in use in the sublibrary, but has not yet been freed, is freed at once.

```
┌─────────┬───────────────────────────────────────────────────────┐
│ COMpare │ {Lib    =1    [:] 1    ...} [Data={Directory}] ..    │
│         │ {Sublib=1.s [:] 1.s ...}        {Member   }          │
│         │ {mn.mt                  ...}                          │
│         │ .. [Punch={Yes|No}]                                   │
└─────────┴───────────────────────────────────────────────────────┘
```

The COMPARE command is used to compare libraries, sublibraries or members, and provide a listing of the differences, as explained under the DATA operand below.

Lib=1 Specifies the names of the libraries to be compared. Comparing is done for all sublibraries in each of the specified libraries.

Sublib=1.s Specifies the sublibraries to be compared. An equals sign can be coded in the second operand in place of an identical name in the first operand.

mn.mt Member(s) to be compared.

Data=Directory|Member Directories or member contents to be compared. The default is DATA=DIRECTORY.

Punch=Yes|No This operand is valid only with DATA=DIRECTORY. If you specify PUNCH=YES, the system generates COPY statements on SYSPCH.

```
┌─────────┬───────────────────────────────────────────────────────┐
│ CONnect │ {Sublib=1.s [:] 1.s | ?}                              │
└─────────┴───────────────────────────────────────────────────────┘
```

The CONNECT command must be used before COPY, MOVE, or COMPARE commands. The function is similar to that of the ACCESS command, except that CONNECT must be used before commands which require two sublibraries to be specified.

The first operand of the CONNECT command specifies the }from}} sublibrary required for the following commands, and the second operand the }to} or second sublibrary.

If a question mark is specified as operand, the current CONNECT information is displayed on SYSLOG. (If command was entered from SYSLOG.)

```
┌─────────┬───────────────────────────────────────────────────────┐
│ COPy    │ {Lib    =1    [:] 1    ...} [Replace={Yes}] [LISt={Yes}] │
│         │ {Sublib=1.s [:] 1.s ...}           {No }        {No }  │
│         │ {mn.mt                  ...}                          │
└─────────┴───────────────────────────────────────────────────────┘
```

The COPY command is used to copy libraries, sublibraries or members.

This command can also be used to merge sublibraries. (See example.)

```
    CONNECT LIBA.SUB1 : LIBB.SUB2
    COPY    *.*
```

Lib=1:1 Libraries to be used in the copy operation.

Sublib=1.s:1.s Sublibraries to be used in the copy operation.

If a name in the second operand is the same as one in the first operand, it can be replaced by an equals sign.

mn.mt Member(s) to be copied. The }from} and }to} sublibraries must be specified in a preceding CONNECT command.

Replace=Yes|No Specifies whether copying should be conditional or unconditional.

LISt=Yes|No If YES is specified, the names and types of the members copied and those of the corresponding }to} and }from} libraries and sublibraries will be printed on SYSLST.

```
DEFine {Lib=1 ...                                  }[Replace={Yes}]
       {Sublib=1.s ... [REUse={Automatic|Immediate}]}        {No }
```

The DEFINE command is used to create system libraries (SYSRES files), private libraries and sublibraries.

Lib=1 Name(s) of the library or libraries to be created. Library names may be 1 to 7 characters long, and must be alphanumeric. The first character must be alphabetic. Use the names IJSYSR1 to IJSYSR9 to define SYSRES files to be created. The name IJSYSRS must not be used, as it defines the IPL'ed system.

Sublib=1.s Qualified name(s) of the sublibrary or sublibraries to be created. Sublibrary names may be 1 to 8 characters long, and must be alphanumeric.

REUse=<u>Automatic</u>|Immediate This operand is needed in a disk sharing environment, or when several tasks share the same sublibrary.
(REUse=Automatic is the default)
REUSE=IMMEDIATE causes the space to be freed as soon as the members are deleted.
REUSE=AUTOMATIC causes the space to be freed only when the sublibrary is in use by only one CPU in the sharing environment, and by only one task.

Replace=No|Yes This operand controls conditional creation of libraries and sublibraries. (Replace=No is the default)

```
DELete | {Lib   =1   ... }
       | {Sublib=1.s ... }
       | {mn.mt ...      }
```

The DELETE command is used to delete members, sublibraries or libraries.

Lib=1 Library or libraries to be deleted. The system library IJSYSRS may not be deleted, as it contains the IPL'ed system.

Sublib=1.s Sublibrary or sublibraries to be deleted. The system sublibrary IJSYSRS.SYSLIB cannot be deleted.

mn.mt Member(s) to be deleted.

```
GOTO   | label
```

The Librarian GOTO command has the same function as the job control GOTO command.

label Specifies the operand of the LABEL command after which processing is to continue.

```
Input  | SYSIPT
```

The INPUT command cause the librarian to read any following commands from SYSIPT instead of SYSLOG until the end of the current job step.

```
List   | mn.mt ... [Unit={SYSLST}] [Format={Hex}]
       |                 {SYSLOG}
```

The LIST command causes the contents of one or more members to be displayed on SYSLST or SYSLOG.
Phases and dumps are listed in a combined hexadecimal and character string format.

mn.mt Member(s) to be displayed.

Unit=SYSLST|SYSLOG Output device to be used. If the LIST command is issued from SYSLOG, the default output device is also SYSLOG.
Output length on SYSLOG is 68.

Format=Hex FORMAT=HEX results in the character string representation of each record of a member, followed by a two-line hexadecimal translation.

```
| LISTDir | {Lib    =1   ...} [Output={Full   } ] [Unit={SYSLST}] |
| LD      | {Sublib=l.s ...}          {Normal}         {SYSLOG}  |
|         | {mn.mt      ...}          {SHort }                   |
|         | {SDL           }          {STatus}                   |
```

The LISTDIR (list directory) command is used to display the contents of a directory. The output is a list sorted in alphanumeric collating sequence.

Lib=l Specifies that the directory information of a library or libraries is to be displayed.
Sublib=l.s Specifies that the directory contents of a sublibrary or sublibraries is to be displayed.
mn.mt Specifies one or more members. Librarian displays only those parts of the sublibrary directory which are relevant to the named member(s).
SDL System directory list is to be displayed. The OUTPUT operand is not applicable when SDL is specified.
OUTPUT=Full|<u>Normal</u>|SHort|STatus Controls the kind and amount of information provided. The specifications FULL, NORMAL and SHORT are applicable for libraries, sublibraries and members. STATUS is applicable only for libraries and sublibraries. The operand may not be specified together with SDL. OUTPUT=NORMAL is the system default.
Unit=SYSLST|SYSLOG Output device to be used. Default is SYSLOG, If command is issued from SYSLOG.

```
| Move    | {Lib    =l    :] l   ...} [Replace={Yes}] [LISt={Yes}] |
|         | {Sublib=l.s [:] l.s ...}           {No }        {No }  |
|         | {mn.mt              ...}                               |
```

The MOVE command works in a similar way to the COPY command, except that the data which have been moved to a target library or sublibrary are deleted from the from-location after they have been copied.

Lib=l:l Libraries to be used in the MOVE function.
Sublib=l.s:l.s Sublibraries to be used in the MOVE function.
mn.mt members to be moved.
Replace=Yes|<u>No</u> Specifies whether moving is to be conditional or not. If YES is specified, the }from} sublibraries or members will be moved to the }to} library or sublibrary in any case. REPLACE=NO prevents the MOVE operation if a duplicate name already exists.
LISt=Yes|<u>No</u> LISt=YES forces a listing of moved member names and types on SYSLST.

```
| ON      | $RC  {>}  n  {GOTO label}                               |
|         |      {<}     {CONTinue  }                               |
|         |      {=}                                                |
```

The ON command allows conditional execution of librarian command streams in batch mode.

The ON command causes the Librarian program to test the return code after each following command. If comparison of return code matches the condition, the specified action is taken.

$RC Return code of any following Librarian command.
>|<|= Specified action is to be taken when the return code is:
    Greater than (>),
    Less than (<) or
    Equal to (=)
  the specified number.
n Number (0 - 9999) with which the return codes are to be compared.

GOTO label or CONTinue Action to be taken if specified comparison matches.

```
PUnch    | mn.mt ...    [Format=Old] [Eof={Yes|NO}]
```

The PUNCH command causes the contents of one or more members to be "punched" to the output device SYSPCH.

mn.mt — Members to be punched. (Members of type DUMP cannot be punched.)

Format=Old — Allows members to be transferred from a current library to a library of a pre-Version-2 format.

Eof=Yes|No — If EOF=YES is specified, an end-of-file indicator (/*) is written to SYSPCH. If EOF=NO is specified, no end-of-file indicator is written.
EOF=YES is the default.

```
RELease | [SPace]  {Lib    =1    ...}
        |          {Sublib=1.s ...}
```

The RELEASE command is needed when members have been deleted from a library or sublibrary which

- is shared by two or more VSE partitions, or
- resides on a disk device shared by two or more processors.

The RELEASE command causes space to be released immediately, which was formerly occupied by deleted members, when the library or sublibrary is no longer shared.

SPace — may be coded as a reminder.

Lib=l — Library where the space from deleted members is to be released.

Sublib=l.s — Sublibrary where the space from deleted members is to be released.

```
REName  | {Sublib=l.s [:] l.s    ...}
        | {mn.mt      [:] mn.mt ...}
```

The RENAME command changes the name and/or type of one or more members, or the names of one or more sublibraries.

Sublib=l.s — Old (first operand) and new (second operand) names of the sublibrary to be renamed. The library name from the first and second operand must be the same. (Not valid for system sublibrary IJSYSRS.SYSLIB.)

mn.mt — Old (first operand) and new (second operand) names and types of the members to be renamed.

```
REStore r-spec Tape={SYSnnn}[LISt={Yes}] [ID={name}] [Replace={Yes}]
                    {cuu   }[     {No }]     {*    }          {No }
                            [          ]
                            [Scan={Yes}]
                            [     {No }]
-----------------------------------------------------------------------
Values for          {Lib    =l    [:l]      ...}
r-spec              {Sublib=l.s [:l.s]     ...}
may be:             {*                         }
                    {l.s.mn.mt  [:l.s]     ...}
                    {Oldlib=oldlib:[l.s] ...}
```

The RESTORE command causes the libraries, sublibraries, members or SYSRES files which were backed up using the BACKUP command to be restored to disk.

Lib=l — Library to be restored.

:l — New name for the library to be restored.
Label and extent information must be given before the RESTORE command.

```
Sublib=l.s      Sublibrary to be restored.
                A sublibrary can be restored only into an existing library.
:l.s            New target sublibrary.
l.s.mn.mt       Member(s) to be restored. They will be searched for in the
                sublibrary specified by l.s on the backup tape.
:l.s            specified members will be copied into this sublibrary, dependent
                on the REPLACE operand of the RESTORE command.
*               All libraries and sublibraries on the backup file are to be
                restored.
Oldlib=oldlib   Filename of a pre-Version-2 format private library, to be
                restored as a sublibrary.
Tape=SYSnnn|cuu Logical unit or device address of tape containing the backup
                file.
LISt=Yes|No     If LIST=YES, the names of restored libraries, sublibraries and
                members will be printed on SYSLST.  If LIST=NO, listing will be
                suppressed.
SCan=Yes|No     If SCAN=YES, the restore function is not performed.
                SCAN=NO is the default
ID=name|*       Name of the backup file to be searched for.  (1 to 16 characters
                enclosed in quotes.)  With alphanumeric characters only, the
                quotes may be omitted.
REPLACE=YES|NO  Controls the merging of backed-up libraries, sublibraries and
                members into existing ones.
```

The standalone version of the RESTORE function restores a single SYSRES file.

```
Test {Lib   =l   ... [Area={Space} ][Repair={Yes}] } [Unit={SYSLST}]
     {                     {All  }          {No }  }       {SYSLOG}
     {Sublib=l.s ...                               }
     {mn.mt                                        }
     {Trace={Space|IO|Buffer|LEVEL1|LEVEL2|OFF]}   }
```

The TEST command is used to check the structure and contents of a library or sublibrary for consistency and correctness, and to provide a trace function for librarian services at different levels.

If TEST detects any inconsistency or incorrectness in a library, sublibrary or member, the librarian sets a return code of 2.

To analyze a possible library problem, follow this procedure:

1. Run TEST LIB=l for the library suspected of causing the problem.
2. If the TEST output does not show error lines, the problem was not caused by this library.

   If the TEST output shows errors, then:
3. Run BACKUP and RESTORE for the library.
4. Run TEST LIB=l again for the same library.
5. If the TEST output does not show error lines, the problem is probably solved.

   If the TEST output does show errors again, then:
6. There is probably a system error.

```
Update | mn.mt  [SAve=mn.mt] [SEquence={n }] [Column=start[:]end]
       |                               {FS}
       |                               {NO}
```

The UPDATE command allows you to modify the contents of a member by adding, deleting or replacing lines.

```
mn.mt             Member to be updated.
SAve=mn.mt        Unmodified version of the member is to be saved under the name
                  and type specified.
SEquence=n|FS|NO Controls the resequencing of the member being updated.

                  n (decimal number from 1 to 999) increment between line numbers
```

which will be used for resequencing. The first line will be given the value n.

FS fixed sequence; the current line numbers will not be changed.

NO the order of records in the member will not be checked. The updates must be supplied in ascending order. Sequencing is not checked.
SEQUENCE=10 is the default.

Column=start:end Start and end of the sequence field in the member. May be anywhere within the line (1 to 8 characters). The following are the defaults.
if SEQUENCE=n or SEQUENCE=NO   COLUMN=77:80 if SEQUENCE=FS
COLUMN=73:80

UPDATE Subcommands

Operation	Operands
)ADD	seq-no

The)ADD subcommand indicates that the lines following it are to be added to the member specified in the UPDATE command.

seq-no Sequence number of the line after which new lines are to be added. To add new lines in front of the first line of the member, code 0 for seq-no. (Not possible if SEQUENCE=NO in the UPDATE command.)

Operation	Operands
)DEL	first-seq-no[[,]last-seq-no | *]

The)DEL subcommand causes the deletion of lines from the member specified in the UPDATE command.

first-seq-no, last-seq-no Sequence numbers of the first and last lines of a section to be deleted. If **last-seq-no** is not specified, the line represented by **first-seq-no** is the only line deleted. To delete all lines from the line specified in first-seq-no to the end of the member, specify * in place of last-seq-no.

Operation	Operands
)END	none

The subcommand)END has no operand. Issue it to inform the system that input for the required UPDATE function is complete.

Operation	Operands
)REP	first-seq-no[[,]last-seq-no | *]

The)REP subcommand indicates following lines are to replace existing lines in the specified member.

first-seq-no,last-seq-no Represent the sequence numbers of the first and last lines of a section to be replaced. The **first-seq-no** must not be zero. To replace all lines up to the end of the member, specify * in place of last-seq-no.

```
/. label
```

The LABEL command in conditional command streams marks a point up to which commands can be skipped using a GOTO command or the GOTO action of an ON command.

/. Indicates a label. These characters must be in positions 1 and 2 of the command followed by at least one blank character.

label Name of the label.(1 to 8 alphanumeric characters)

```
/+ [comments]
```

The End-of-Data statement for input to the librarian CATALOG command is /+.

Column 1 contains a slash (/) and column 2 a plus sign (+). Column 3 must be blank.

Format for SYSIPT:

```
/*
```

Format for SYSLOG:

```
END
```

These statements indicate to the librarian program that no more librarian commands follow.
The SYSIPT format is used when a librarian job stream for batch execution is being prepared.
The SYSLOG format is used to end an interactive librarian session at the system console.

ESERV Control Statements

These statements must start in or after **column 2**.

If used, one of these statements must follow the EXEC ESERV statement directly. If neither is used, GENCATALS is assumed.

Operation	Operands
GENEND	none

This causes ESERV to place an END and a /* statement immediately after the de-edited macro on SYSPCH, so that it can be used as SYSIPT for the assembler.

Operation	Operands
GENCATALS	none

This causes a librarian catalog statement for a member "bookname.A" (or "bookname.D," if the OPTION SUBLIB=DF is in effect on the system) to be placed before each macro, and a /* to be placed after each macro. This allows the SYSPCH output to be used as SYSIPT for the librarian program to catalog the de-edited macro with the appropriate member type.

Operation	Operands
DSPLY PUNCH DSPCH	bookname.type[,bookname.type,...]

These statements must follow the GENEND or GENCATALS statement. They can act on one or more edited macros in one ESERV run.
The statements must start in or after **column 2**.

DSPLY De-edits macros and displays them on SYSLST.
PUNCH De-edits macros and punches them on SYSPCH.
DSPCH De-edits macros, punches them on SYSPCH and displays them on SYSLST.
bookname.type Member name and member type of the macro to be de-edited.

Operation	Operands
) ADD	seq-no[+rel]

The) ADD statement indicates that following source statement(s) are to be inserted at specified positions in the macro.

seq-no Sequence number of the macro definition statement after which the new source statements are to be inserted. (1 to 8 decimal digits)
+rel Position of the macro statement after which the new statements are to be added, **relative** to the statement specified in "seq-no"

Operation	Operands
) COL	startcol,n

The) COL statement specifies the position of the sequence number within the source statements of the de-edited macro. If used, it must immediately follow the DSPLY, PUNCH or DSPCH statement to which it applies.

startcol Column in which the sequence number is to start. (73..80 - default is 73)
n Length of the sequence number. (1..8 - default is 6)

Operation	Operands
) DEL	first-seq-no[+rel][,last-seq-no[+rel]]

The) DEL statement causes deletion of one or more source statements from the de-edited macro.

first-seq-no Sequence number of the **first or only** source statement to be deleted from the de-edited macro.
last-seq-no Sequence number of the **last** of a series of source statements to be deleted from the de-edited macro.
+rel Position of the first or last statements to be deleted, **relative** to the statement with the specified sequence number.

Operation	Operands
) END	none

The) END statement indicates the end of ESERV update or verify statements on SYSIPT. It is required in every update run.

Operation	Operands
) REP	first-seq-no[+rel][,last-seq-no[+rel]]

The) REP statement indicates that the following source statements on SYSIPT are to replace one or more existing statements in the de-edited macro.

first-seq-no Sequence number of the **first or only** source statement to be replaced in the de-edited macro.
last-seq-no Sequence number of the **last** of a series of source statements to be replaced in the de-edited macro.
+rel Position of the first or last statements to be replaced, **relative** to the statement with the specified sequence number.

Operation	Operands
) RST	seq-no[+rel]

The) RST statement causes the sequence numbers of the statements in a macro definition to restart at a lower number after the statement specified in the) RST operand.

seq-no Sequence number of the source statement after which the new series of sequence numbers starts.
+rel Position of the statement after which the new series of numbers is to start, **relative** to the statement with the specified sequence number.

Operation	Operands
) VER	seq-no[+rel],len

The) VER statement causes all or part of the specified source statement in the de-edited macro to be verified against the contents of the statement following) VER statement on SYSIPT.

seq-no   Sequence number of the source statement to be verified in the de-edited macro.

+rel   Position of the source statement to be verified, relative to the statement specified in "seq-no."

len   Length of the field to be verified. (decimal integer; 1..80)

Sample coding for de-editing without updating a macro definition

```
// JOB NOUPDATE
// LIBDEF SOURCE,SEARCH=(lib.sublib,....)
// EXEC ESERV
 PUNCH E.MAC1,E.MAC2
/*
/&
```

Sample coding for de-editing and updating a macro definition

```
// JOB UPDATE
// LIBDEF SOURCE,SEARCH=(lib.sublib,....)
// EXEC ESERV
 GENEND
 DSPCH E.MAC1
)COL 77,3
)VER 72+1,5
.PP9
)ADD 72+1
   AIF (&PCH NE 1400)D4
)DEL 102,103
)REP 245
JOYCE CLC 0(4,REG6),BLANKS
)END /*
// PAUSE CHECK LIST, MOVE DECK TO READER
// OPTION EDECK,NODECK
// EXEC ASSEMBLY
         deck produced by ESERV
// PAUSE MOVE SYSPCH DECK TO READER
// EXEC LIBR,PARM='ACCESS SUBLIB=lib.sublib'
         deck produced by Assembler
/*
/&
```

LSERV: Display Label Information Area

LSERV, a system utility program, produces a printout of the system's label information area. You invoke this program by entering:

```
// EXEC LSERV
```

via the console or by submitting control statements as follows.

```
// JOB anyname
// EXEC LSERV
/*
/&
```

The output of LSERV shows the contents of the label area on SYSRES and is to be printed on SYSLST.rd Punch

When and how to use LSERV:

1. Operator action given in the Messages indicates when LSERV must be executed.
2. LSERV can be used for error analysis. LSERV displays the TLBL, DLBL and EXTENT information.

Summary of information provided:

The printout of LSERV will show you the following details:

- Whether the correct DLBL/EXTENT information is still on the label area.
- The permanent files.
- The temporary files.
- Extent type.
- File type.

LVTOC: Display Volume Table of Contents

```
        // JOB anyname
        // ASSGN SYS004,cuu (disk)
        // ASSGN SYS005,cuu (printer)
        // EXEC LVTOC
        /&
```

A volume table of contents (VTOC) is an index of all files, and the remaining space, on a disk volume.

A VTOC display can be requested by executing the LVTOC program with SYS004 assigned to the applicable disk drive and SYS005 to a printer.

LVTOC lists the file labels contained in a VTOC in alphabetic sequence by file name. It also provides a listing of free space on the volume, with the start and end addresses and sizes of the unused space. The control statements needed to invoke that program may be submitted via SYSRDR or via the console.

LINKAGE EDITOR SUMMARY

Format of the ESD Statement

Card columns	Content
1	Multiple punch (12-2-9). Identifies this as a loader statement.
2 - 4	ESD -- External Symbol Dictionary statement.
11 - 12	Number of bytes of information contained in this statement.
15 - 16	External symbol identification number (ESID) of the first SD, PC, CM or ER on this statement. Relates the SD, PC, CM or ER to a particular control section.
17 - 72	Variable information. 8 positions Name 1 position Type code hex '00', '01', '02', '04', '05', or '0A' to indicate SD, LD, ER, PC, CM, or WX, respectively. 3 positions Assembled origin 1 position Blank 3 positions Length, if an SD-type, CM-type, or a PC-type. If an LD-type, this field contains the external symbol identification number (ESID) of the SD containing the label.
73 - 80	May be used by the programmer for identification.

Format of the TXT Statement

Card columns	Content
1	Multiple punch (12-2-9). Identifies this as a loader statement.
2 - 4	TXT -- Text statement.
6 - 8	Assembled origin (address of first byte to be loaded from this statement).
11 - 12	Number of bytes of text to be loaded.
15 - 16	External symbol identification number (ESID) of the control section (SD or PC) containing the text.
17 - 72	Up to 56 bytes of text -- data or instructions to be loaded.
73 - 80	May be used for program identification.

Format of the RLD Statement

Card columns	Content
1	Multiple punch (12-2-9). Identifies this as a loader statement.
2 - 4	RLD -- Relocation List Dictionary statement.
11 - 12	Number of bytes of information contained in this statement.
17 - 72	Variable information (multiple items).

a. Two positions - (relocation identifier) pointer to the ESID number of the ESD item on which the relocation factor of the contents of the address constant is dependent.

b. Two positions - (position identifier) pointer to the ESID number of the ESD item on which the position of the address constant is dependent.

c. One position - flag byte indicating type of constant, as follows:

Bits	Setting and Meaning
0-2	(ignored)
3	0 - a nonbranch type load constant 1 - a branch type load constant
4-5	00 - load constant length = 1 byte 01 - load constant length = 2 bytes

10 - load constant length = 3 bytes
11 - load constant length = 4 bytes

6 0 - relocation factor is to be added
1 - relocation factor is to be subtracted

7 0 - Next load constant has different R and P identifiers; therefore, both R and P must be present.
1 - Next load constant has the same R and P identifiers; therefore, they are both omitted.

The five significant bits of this byte are expanded in the RSERV printout.

d. Three positions - assembled origin of load constant.

73 - 80 May be used for program identification.

Format of the END Statement

Card columns	Content
1	Multiple punch (12-2-9). Identifies this as a loader statement.
2 - 4	END
6 - 8	Assembled origin of the label supplied to the assembler in the END statement (optional).
15 - 16	ESID number of the control section to which this END statement refers (only if 6-8 present).
17 - 22	Symbolic label supplied to the assembler if this label was not defined within the assembly.
29 - 32	Control section length (if not specified in last SD or PC).
73 - 80	Not used.

Format of the REP (User Replace) Statement

Card columns	Content
1	Multiple punch (12-2-9). Identifies this as a loader statement.
2 - 4	REP -- Replace text statement.
5 - 6	Blank.
7 - 12	Assembled address of the first byte to be replaced (hexadecimal). Must be right justified with leading zeros if needed to fill the field and must be equal to or greater than the starting address of the control section (columns 14-16). Note that there is no check to determine if the assembled address is actually within this control section.
13	Blank.
14 - 16	External symbol identification number (ESID) of the control section (SD) containing the text (hexadecimal). Must be right justified with leading zeros if needed to fill the field.
17 - 70	From 1 to 11 4-digit hexadecimal fields separated by commas, each replacing two bytes. A blank indicates the end of information in this statement.
71 - 72	Blank.
73 - 80	May be used for program identification.

External Symbol Dictionary

The external symbol dictionary (ESD) contains control section definitions and inter-module references. Six types of entries are defined in the control dictionary:

ESD Type	Definition
SD	Section definition: provides control section name, assembled origin and length. Generated by a named START or a named CSECT in a source module.
WX	Generated by weak external reference (WXTRN), which has a function similar to EXTRN, except that WXTRN suppresses AUTOLINK. The linkage editor treats WX as an ER, NOAUTO.
PC	Private code: provides assembled origin and length for an unnamed control section.
LD/LR	Label definition: specifies the assembled address and the associated SD of a label that may be referred to by another module. The LD entry is termed LR (Label Reference) when the entry is matched to an ER entry.
ER	External reference: specifies the location of a reference made to another module. ER is generated by EXTRN or a V-type address constant in a source module.
CM	Common: indicates the amount of storage to be reserved for common use by different phases. CM is generated by COM in a source module.

SUPVR macro

The SUPVR generation macro describes the system environment:

ID= Supervisor Identification Character

MICR= Support for magnetic ink or optical character reader/sorter.

- NO specifies no MICR support
- 1419 indicates support for 1419s with singel address adapter.
- 1419D indicates support for 1419s with dual address adapter.

MODE= 370|E|VM Specifies whether 370 or ECPS/VSE mode is supported. MODE=VM corresponds to the old specification MODE=E and VM=YES, and generates the VM linkage enhancement version with 'DAT' off.

NPARTS= 5|n specifies the number of partitions to be supported. The minimum is 2, the maximum is 12.

FOPT macro

The FOPT generation macro describes the functional supervisor options:

DASDSHR= NO|YES specifies whether disk sharing support is used or not.

FASTTR= NO|YES specifies whether fast CCW translation is to be supported.

RPS= NO|YES Support for the rotational positioning sensing (RPS) capabilities.

TRKHLD= NO|n Specifies whether the Track Hold feature is to be supported for disk in a supervisor that supports multiprogramming. n indicates the maximum number of tracks/blocks to be held at any one time. (Accepted values are from 1 to 255).

TTIME= NO|partition ID Specifies whether th timer is to be supported and if so, the partition owning the task timmer. Only one partition ID can be specified.

USERID= id print supervisor ID at IPL completion.

IOTAB macro

The IOTAB generation macro describes installation requirements for I/O tables:

IODEV= 25|n Number of I/O devices attached to the system. The maximum specification is 254, the minimum is 4. Each unit requiring an ADD entry must be included in n.

NPGR= 30|n Number of programmer logical units for all partitions. (Accepted values are from 30 to 255).

DEVICE TYPE CODES

IPL Device Code	Actual IBM Device	PUB Device Type X'nn'	Device Type
7770	7770 Audio Response Unit	D3	Audio Response Units
7772	7772 Audio Response Unit	D4	
2501	2501 Card Reader	10	Card Reader
2540R	2540 Card Reader	11	
3505	3505 Card Reader	12	
1442N2	1442N2 Card Punch	22	Card Punch
2520B2	2520B2 Card Punch	20	
2520B3	2520B3 Card Punch	20	
2540P	2540 Card Punch	21	
3525P	3525 Card Punch	23	
1442N1	1442N1 Card Read Punch	30	Card Read Punch
2520B1	2520B1 Card Read Punch	31	
2560	2560 Multifunction Card Machine	33	
3525RP	3525 Card Punch (with optional read/punch feature)	32	
5425	5424 Multifunction Card Unit	34	
	5425 Multifunction Card Unit		
FBA	3310 Fixed Block Storage Device	90	Disk
FBA	3370 Fixed Block Storage Device	90	
FBA	3370-2 Fixed Block Storage Device	90	
2311	2311 Disk Storage Device	60	
2314	2314 Disk Storage Device	62	
2314	2319 Disk Storage Device	62	
3330	3330 Disk Storage, Models 1 and 2	63	
	3330-1		
3330B	3330 Disk Storage Model 11	65	
3340	3340 Disk Storage (General)	68	
3340	3340 Disk Storage without RPS feature	69/6A	
	3344 Disk Storage w/o RPS feature	6A	
3340R	3340 Disk Storage with RPS feature	69/6A	
	3344 Disk Storage with RPS feature	6A	
3350	Disk Storage	67	
3375	Disk Storage	6B	
3380	Disk Storage	6C	
3540	3540 Diskette Input/Output Unit	80	Diskette
7443	7443 System Recording File	88	
3277	3277 Display Operator Console	B0	Display Operator Console and Console Printers
	3284 Console Printer	B0	
	3286 (the MODE operand must be		
	3287 entered as X'02')		
1050A	3210 Console Printer Keyboard	00	Printer Keyboard
	3215 Console Printer Keyboard	00	
	3286-2 in Printer Keyboard Mode	00	

Device Type Codes (cont...)

IPL Device Code	Actual IBM Device	PUB Device Type X'nn'	Device Type
2260	2260 Display Station	C0	Display Station
3277 (local)	3277 Display Station MODE operand 3278 must be omitted) 3279	B0	
3277B (local)	3277 Display Station, attached in 3278 Burst Mode to a Multiplexor 3279 Channel (MODE operand must be omitted)	B0	
3277	3277 Display Station, attached via 3274-1D Control Unit, mode = X'05'	B0	
2400T7	2400 7-Track Magnetic Tape Unit	50	Magnetic Tape
2400T9	2400 9-Track Magnetic Tape Unit	50	
3410T7	3410 7-Track Magnetic Tape Unit	53	
3410T9	3410 9-Track Magnetic Tape Unit	53	
3420T7	3420 7-Track Magnetic Tape Unit	52	
3420T9	3420 9-Track Magnetic Tape Unit	52	
3430	3430 9-Track Magnetic Tape Unit	53	
8809	8809 Magnetic Tape Unit	5A	
1419	1255 Magnetic Character Reader	72	MICR (Magnetic Ink Character Recognition Device)
1419	1259 Magnetic Character Reader	72	
1419	1419 Magnetic Character Reader	72	
1419P	1419 Dual Address Adapter Primary Control Unit	73	
1419S	1419 Dual Address Adapter Secondary Control Unit	74	
3890	3890 Document Reader/Inscriber	7E	Reader/ Inscriber
3895	3895 Document Reader/Inscriber	7D	
1287	1287 Optical Reader	77	Optical Reader
1288	1288 Optical Page Reader	77	
1419	1270 Optical Reader Sorter	72	
1419P	1275 Optical Reader Sorter Primary Control Unit	73	
1419S	1275 Optical Reader Sorter Secondary Control Unit	74	
3881	3881 Optical Mark Reader	11	
3886	3886 Optical Character Reader	7C	
PRT1	3211 Printer 3203-4 Printer 3203-5 Printer 3262-2 Printer 3262-5 Printer 3262-12 Printer 3289-4 Printer 4245 Printer 4248 Printer	43	Printer
1403	1403 Printer	40	Printer
1403U	1403 Printer with UCS feature	42	
1443	1443 Printer	41	
3211	3211 same as PRT1	43	
3800	3800 Printing Subsystem	45	
3800B	3800 Printing Subsystem with Burster-Trimmer-Stacker (BTS)	45	
3800C	3800 Printing Subsystem with additional Character Generation Storage (CGS)	45	
3800BC	3800 Printing Subsystem with BTS and CGS	45	

Device Type Codes (cont...)

IPL Device Code	Actual IBM Device		PUB Device Type X'nn'	Device Type
3277 (local)	3284 3286 3287 3288	Printer with 3277 or 3274-1B Control Unit (MODE operand must be entered as X'01')	B0	Terminal Printer
	3289	Printer with 3274-1B Control Unit (MODE operand must be entered as X'01')		
3277B (local)	3284 3286 3287 3288	Printer with 3277 or 3274-1B Control Unit (MODE operand must be entered as X'01')	B0	
	3289	Printer with 3274-1B Control Unit (MODE operand must be entered as X'01') attached in burst mode to a multiplexor channel		
3277	3277 4550	Printers attached via 3274-1D Control Unit, mode=X'06'	B0	
2701	2701	Data Adapter Unit	D0	Tele-processing Lines
	2715	Data Adapter Unit	D0	
2701	Model 135	Integrated Communication Adapter (ICA)	D0	
2702	2702	Transmission Control Unit	D1	
2703	2703	Transmission Control Unit	D2	
2703	Model 138	Integrated Communication Adapter (ICA)	D2	
2703	4331	Communication Adapter (ICA) for BSC or Start/Stop lines	D2	
2703	3704 3705	Communications Controller in Emulation Mode	D2	
3704	3704	Communications Controller	DC	
3705	3705	Communications Controller	DC	
3705	4331	Communications Adapter (ICA) for SDLC Mode=X'10'	DC	
3725	3725	Communications Controller	DC	
3791L	3791	Local Communications Controller	DE	
3791L	3274-1A	Local Communications Contr.	DE	
UNSP		Unsupported Device	FF	Unsupported Device
UNSPB		Unsupported Device (burst)	FF	

CHAPTER 3. IOCS (GENERAL)

Standard volume labels on disk

Standard Volume Label 1 Fields (Disk)

Displ.	Field	Length	Content
0	K1	3	Identifier: VOL. IOCS checks whether a VOLUME Label is present on the volume.
3	K2	1	Volume Label No. VSE supports only VOL1.
4	D1	6	Volume serial number provides a unique identification for the volume. It is generally assigned when the volume is first received in the installation. The source of information is the EXTENT statement. IOCS checks the Serial No. given in the EXTENT statement against this field. If no Serial No. Operand is specified in the EXTENT Statement IOCS assumes the correct volume mounted.
A	D2	1	Security byte used by OLTEP
B	D3	5	VTOC address. Contains the address of the Format-4 label. (OCCHH for CKD or OBBB for FBA) This address is written at initialization time.
10	D4	5	Blank
15	D5	4	CI-size for FBA, blanks for CKD
19	D6	4	Number of blocks per CI for FBA, blanks for CKD
1D	D7	4	Number of labels per CI for FBA, blanks for CKD
21	D8	4	Blank
25	D9	E	Owner code for LVTOC listing
33	D10	1D	Blank

Note:

K. Fields = Key Fields
D. Fields = Data Fields

Displ.	Field	Length	Content
0	K1	2C	File-ID: 1-35 bytes if generation number (Gnnn) and version number (Vnn) are specified, else 1 to 44. Source of Information: DLBL or IOCS VSAM catalog routines, AMS DEFINE command. Processing: The File-ID may be specified in the DLBL-File-ID field, if this specification is omited, IOCS uses the DTF-name specified in the DLBL-filename-field (stored in the key area of the label record) to search (on input) in the VTOC key areas for the file entry. Under VSAM a data space name (VSAM catalog routine) or the name of an index or data is the contents (AMS DEFINE Stmt. or generated by VSAM).
2C	D1	1	Format ID: 1. Written (on output) and checked by IOCS to distinguish this label from the other types (Format 2-5).
2D	D2	6	Volume serial no.: numeric identification for the first volume of the file. Written by IOCS on output.
33	D3	2	Volume sequence number within the file to identify the volume in an multivolume file. Written (on output) and checked by IOCS.
35	D4	3	Creation date: yyddd. By IOCS from SYSCOM (on output), checked against label record (DLBL) on input. The actual year may be calculated by adding yy to 1900.
38	D5	3	Expiration date indicates when the data record is considered inactive. (Same format as creation date.) Source: DLBL, IOCS, AMS or System (creation date + 7 by default).
3B	D6	1	Number of extents of the multi extent file on this volume.
3C	D7	1	Used by OS/VS
3D	D8	1	Reserved
3E	D9	D	System code: indicates the Programming System which has written the file. IBMDOSVS is the code written by IOCS if DLBL is used.
4B	D10	3	Date of last access: yyddd; not used by VSE
4E	D10A	2	Reserved
50	D10B	2	Number of blocks per CI for FBA, blanks for CKD

IBM-Standard Disk File Label (Format-1) (cont...)

Displ.	Field	Length	Content
52	D11	2	File type: hex 0008 for VSAM hex 2000 for DAM hex 4000 for SAM (default, field in DLBL omitted) hex 8000 for ISAM Checked against type of DTF on input. Written from DLBL by IOCS on output.
54	D12	1	Record Format: Used by OS/VS. IOCS writes 0
55	D13	1	Flags for optional areas used for ISAM file: Bit 2: Master index Bit 3: Independent overflow area Bit 4: Cylinder overflow area From DTF and EXTENT
56	D14	2	Byte length of ISAM blocks, from DTF
58	D15	2	Record length of ISAM files. From DTF
5A	D16	1	Key length of ISAM blocks. From DTF
5B	D17	2	Key field location in ISAM block. From DTF
5D	D18	1	Flags: Bit 0: Last volume (SAM only) Bit 3: File security. From DLBL
5E	D19	1	Original space request was: Bit 1: in blocks 4: for continuous extent 5: for maximum continuous extent 6: not under specified minimum
5F	D19A	3	Used by OS/VS. IOCS writes blanks.
62	D20	5	Used by OS/VS. IOCS writes zeros.
67	D21	2	Start of next record to end-of-data distance (negative displacement).
69	D22	1	Type of extent: Categorie of records (from EXTENT) 01: (prime) data area or data space extent (default) 02: independent overflow area extent 04: master/cylinder index area extent 40: extent for user-standard labels 80: split cylinder extent (SAM)
6A	D23	1	Sequence number of extent in the file. From EXTENT or IOCS
6B	D24	4	Extent lower limit (cchh for CKD, bbbb for FBA)
6F	D25	4	Extent upper limit (cchh for CKD, bbbb for FBA)
73	D26	1	Type of first additional extent (if available): Categorie of records (from EXTENT) – see D22
74	D27	1	Sequence number of extent in the file. From EXTENT or IOCS
75	D28	4	Extent lower limit (cchh for CKD, bbbb for FBA)
79	D29	4	Extent upper limit (cchh for CKD, bbbb for FBA)

IBM-Standard Disk File Label (Format-1) (cont...)

Displ.	Field	Length	Content
7D	D30	1	Type of second additional extent (if available): Categorie of records (from EXTENT) - see D22
7E	D31	1	Sequence number of extent in the file. From EXTENT or IOCS
7F	D32	4	Extent lower limit (cchh for CKD, bbbb for FBA)
83	D33	4	Extent upper limit (cchh for CKD, bbbb for FBA)
87	D34	5	Address of next label for the file on this volume. Written and used by IOCS.

IBM-Standard Disk File Label (Format-2)

Displ.	Field	Length	Content
0	K1	1	Key code for continuation label (X'02') written by IOCS
1	K2	7	Used by OS/VS
8	K3	5	Used by OS/VS
D	K4	7	Used by OS/VS
14	K5	5	Used by OS/VS
19	K6	11	Reserved
24	K7	8	Last prime track address on the last prime cylinder. (CCHHR)
2C	D1	1	Format ID: Checked by IOCS to distinguish this label from the other types (Format 1 and 3–5)
2D	D2	1	Number of Index levels 1 = Cylinder Index 2 = Cylinder Index and Master Index binary
2E	D3	1	Used by OS/VS
2F	D4	3	First Data Record in Cylinders (HHR)
32	D5	2	Last Data Track in Cylinders (HH)
34	D6	1	Used by OS/VS
35	D7	1	Highest 'R' on High level Index tracks
36	D8	1	Highest 'R' on Prime Data tracks
37	D9	1	Highest 'R' on Overflow tracks
38	D10	1	'R' of last Data Record on shared tracks
39	D11A	1	'R' of last Data Record on unshared track
3A	D11B	1	Highest 'R' on Independent Overflow track
3B	D12	2	TAG deletion count Passed by VSE between Format–2 label and DTF. Used along with fields D13, D16, D27, D28, and D29.

IBM-Standard Disk File Label (Format-2) (cont...)

Displ.	Field	Length	Content
3D	D13	3	Non-first Overflow reference count Passed by IOCS to accumulate this count in FILENAMER during retrieve operation. Used along with fields D12, D16, D27, D28, and D29.
40	D14	2	Number of bytes for highest level Master Index
42	D15	1	Number of tracks for highest level Master Index
43	D16	4	Prime record count Used along with fields D12, D13, D27, D28, and D29.
47	D17	1	Status (Codes for filled area) If on, bits have following meaning: Bit 2 – File has been successfully closed Bit 6 – Last track full Bit 7 – Last block full Otherwise each bit is off
48	D18	7	Address of Cylinder Index (MBBCCHH) M = Extent Sequence Number
4F	D19	7	Used by OS/VS
56	D20	7	Used by OS/VS
5D	D21	8	Last Prime Data Record Address (MBBCCHHR) M = Extent sequence number
65	D22	5	Last Track Index Entry Address (CCHHR)
6A	D23	5	Last Cylinder Entry Address (CCHHR)
6F	D24	5	Last Master Index Entry (CCHHR)
74	D25	8	Last Independent Overflow Record Address (MBBCCHHR) – M = Extent sequence number
7C	D26	2	Used by OS/VS
7E	D27	2	Number of Independent Overflow Tracks Used by IOCS on ADD operations to update the count in FILENAMEI. Used along with fields D12, D13, D16, D28, and D29.
80	D28	2	Overflow Record Count – IOCS accumulates this count on ADD operations into FILENAMEO. Used along with fields D12, D13, D16, D27, and D29.
82	D29	2	Cylinder Overflow Area Count – IOCS accumulates this count on ADD operations into FILENAMEA. Used along with fields D12, D13, D16, D27, and D28.
84	D30	3	Dummy Track Index Entry (HHR)
87	D31	5	Used by OS/VS

IBM-Standard Disk File Label (Format-3)

Displ.	Field	Length	Content
0	K1	4	Key code for continuation label(03030303) Written by IOCS
4	K2	1	Type of extent, from EXTENT: 01 = data extent (default) 80 = split cylinder extent
5	K3	1	Extent sequence number (3 or more)
6	K4	4	Extent lower limit (cchh for CKD, bbbb for FBA)
A	K5	4	Extent upper limit (cchh for CKD, bbbb for FBA)
The fields K2 to K5 are repeated three times as K6 – K17, to describe the extents 2, 3, and 4 of the key area.			
2C	D1	1	Continuation label code: EBCDIC 3, from IOCS
The fields K2 to K5 are repeated nine more times as D2 – D37, to describe the nine extents of the data area.			
87	D38	5	Address of next contin.label (cchhr or 0bbbb) or zeros. From SAM IOCS only

VTOC Label (Format-4)

Every field in this label, except the VSAM indicators (D9A), is written by DSF at initialization time.

Disp.	Field	Length	Content
0	K1	2C	Key code for VTOC label: 44 times 04
2C	D1	1	VTOC label identifier: EBCDIC 4.
2D	D2	5	Used by OS/VS
32	D3	2	Number of available file label spaces in VTOC at initialization (tracks x cylinder minus 2)
34	D4	4	Address of next alternate track (cchh), for FBA: zeros. From DSF
38	D5	2	Number of alternate tracks left. For FBA zeros From DSF
3A	D6	1	Flags: Bit 0: always on Bit 3: Volume reserved for emulators Bit 5: VTOC being updated by VSAM
3B	D7	1	Extent count. Always 1. VTOC is 1 extent
3C	D8	2	Reserved
3E	D9	E	CKD device constants: (FBA: zeros)
3E		2	Number of cylinders
40		2	Tracks per cylinder
42		2	Track length
44		1	Overhead bytes for I*
45		1	Overhead bytes for L*
46		1	Overhead bytes for K*
47		1	Flag byte Bit 4: I or L value* has two bytes for 3350 Bit 7: A tolerance is added to each record except the last on a track
48		2	Tolerance** per device type
4A		1	Number of labels on VTOC track
4B		1	Reserved
4C	D9A	B	VSAM indicators, from VSAM catalog routines
4C		8	Time when last data space was added
54		1	Ownership byte: Bit 0: Owned by VSAM catalog
55		2	Number of first track of CKD catalog recovery area, for FBA zeros
57	D10A	9	Used by OS/VS
	D10B		Used by OS/VS
60	D10C	4	Number of first block of FBA catalog recovery area, for CKD zeros
64	D10D	5	Reserved
69	D11	1	Extent type: 01 for VTOC extent
6A	D12	1	Extent sequence number: 00 (VTOC has 1 extent)
6B	D13	4	Start address of VTOC (label).
6F	D14	4	End address of VTOC. Used by IOCS
73	D15	19	Zeros

*) I = for a record with key area
L = for a last record with key area on a track
K = for a key area

**) The tolerance is added to the length of a record if bit 7 in the flag byte is on.

User-Standard Disk File Labels (Header and Trailer)

Displ.	Field	Length	Content
0	K1	3	UHL or UTL (User-header or User-trailer label)
3	K2	1	Label sequence number: 1 to 8 for header labels 0 to 7 for trailer labels
4	D1	3	UHL or UTL (User-header or User-trailer label)
7	D2	1	Label sequence number: 1 to 8 for all
8	D3	4C	User's label information

Volume Labels on Diskette

Displ.	Field	Length	Content
0	D1	3	Label ID: VOL
3	D2	1	Must contain a 1.
4	D3	6	Volume serial number from EXTENT
A	D4	1	Accessibility indicator: S or Blank. From DTF
B	D5 D6	1A	Reserved
25	D7	E	Name or code of volume owner
33	D8	1C	Reserved
4F	D9	1	Label standard level: W

A diskette volume has one volume label of 80 bytes. It is located on track 0, sector 7 and begins by VOL.

Displ.	Field	Length	Content
0	D1	3	Label ID: HDR
3	D2	1	Label sequence number: 1
4	D3	1	Blank
5	D4	8	File-ID from DLBL or system
D	D5	9	Blanks
16	D6	5	Record length. From IOCS
1B	D7	1	Blank
1C	D8	5	Start address of extent: Track and sector. From IOCS (CCHRR)
21	D9	1	Blank
22	D10	5	End address of extent: Track and sector. From IOCS (CCHRR)
27	D11	1	Blank
28	D12	1	Bypass byte: B or blank: B = job ends on input
29	D13	1	Security byte: S or blank
2A	D14	1	Write protection byte: P or blank
2B	D15	1	Interchange level: blank= sector length 128, unblocked, unspanned, sequential non-blank= job ends on input
2C	D16	1	Volume byte: blank = file complete on this volume C = file continued on next volume L = file ends on this volume
2D	D17	2	Volume sequence number
2F	D18	6	Creation date: YYMMD
35	D19	D	Blanks
42	D20	6	Expiration date: Default= 7 days after output (YYMMDD)
48	D21	1	Verify byte: V or blank
49	D22	1	Blank
4A	D23	5	End of data address (CCHRR)
4F	D24	1	Blank

Volume Labels on Tape

The volume label for tapes is 80 bytes long and begins by VOL1 for the first volume label. Additional volume labels are ignored by VSE.

Tape Volume Label for EBCDIC Code

Displ.	Field	Length	Content
0	D1	3	Label ID: VOL
3	D2	1	Ignored by VSE
4	D3	6	Volume serial number
A	D4	1	Ignored by VSE
B	D5 D7	1E	Reserved
29	D8	A	Volume owner name or code
33	D9	1D	Reserved

Tape Volume Label for ASCII Code

Displ.	Field	Length	Content
0	D1	3	Label ID: VOL
3	D2	1	Ignored by VSE
4	D3	6	Volume serial number
A	D4	1	Accessibility
B	D5 D6	1A	Reserved
25	D7	E	Name or code of volume owner
33	D8	1C	Reserved
4F	D9	1	Standard byte: 1= file has ANSI standards blank= file does not have ANSI standard

IBM-Standard File Labels on Tape

IBM-standard file labels are 80 bytes long. Each file has a header and a trailer label which have the same format, for reading the tape forward or backward. The first four characters of each label identify the particular label:

```
header label  -- HDR1, HDR2
trailer label -- EOF1,EOF2 at the end of a file
                 EOV1,EOV2 at the end of a volume
                           but not of the file
HDR2, EOF2 and EOV2 for ASCII only.
```

Additional labels (HDR3 to 8) are ignored by VSE.

IBM-Standard Tape File Label for EBCDIC Code

Displ.	Field	Length	Content
0	D1	3	Label ID: HDR, EOF, or EOV
3	D2	1	Label sequence number: 1
4	D3	11	File-ID from TLBL
15	D4	6	Volume serial number of the volume where the file begins
1B	D5	4	Volume sequence number within the file
1F	D6	4	File sequence number on the volume
23	D7	4	Version number of the file
27	D8	2	Sub-version number
29	D9	6	Creation date: cyyddd c indicates the century, blank=19, 0=20, 1=21
2F	D10	6	Expiration date: cyyddd
35	D11	1	Ignored by VSE
36	D12	6	Number of blocks; used in trailer labels only
3C	D13	D	System code: IBMDOSVS
49	D14	7	Reserved

IBM-Standard Tape File Label for ASCII Code

Displ.	Field	Length	Content
0	D1	3	Label ID: HDR, EOF, or EOV
3	D2	1	Label sequence number: 1
4	D3	11	File-ID from TLBL
15	D4	6	Volume serial number of first volume of the file
1B	D5	4	Volume sequence number within the file
1F	D6	4	File sequence number within volume(s)
23	D7	4	Version number of the file
27	D8	2	Sub-version number
29	D9	6	Creation date: cyyddd - c indicates the century, X'40'=19, X'F0'=20, X'F1'=21
2F	D10	6	Expiration date: cyyddd
35	D11	1	Accessibility byte
36	D12	6	Number of blocks written; only in trailer label
3C	D13	D	System code: IBMZLB followed by two blanks
49	D14	7	Reserved

User-Standard File Labels on Tape

Displ.	Field	Length	Content
0	D1	3	Label ID: UHL or UTL
3	D2	1	Label sequence number: 1 to 8
4	D3	4C	User's label information

User-standard labels are header labels located and processed before the data of the file, and trailer labels located and processed after the file. Header and trailer labels are identified by:

User header labels	UHLn
User trailer labels	UTLn

n may be 1 to 8.

User-standard file labels are 80 bytes long. The first four bytes contain UHLn or UTLn and the remaining 76 bytes contain user data.

Non-Standard File Labels on Tape

Non-standard labels are only supported on EBCDIC code tape labels. They may have any length, do not have a specified identification in the first four characters, and do not have a fixed format. They may contain whatever information the user desires, and in any arrangement. They are completely the responsibility of the user.

Declarative Macros

Name	Operation	Operands	Description
[name]	CDMOD	[CONTROL=YES] [,CRDERR=RETRY] [,CTLCHR={ASA|YES}] [,DEVICE=nnnn] [,FUNC={R|P|I|RP|RW|RPW|PW}] [,IOAREA2=YES] [,RDONLY=YES] [,RECFORM={FIXUNB|VARUNB|UNDEF}] [,SEPASMB=YES] [,TYPEFLE={INPUT|OUTPUT|CMBND}] [,WORKA=YES]	Defines a logic module for a card reader file.
[name]	DFR	FONT=code [,BCH=n] [,BCHSER=n] [,CHRSET=n] [,EDCHAR=(x,...)] [,ERASE={NO|YES}] [,NATNHP={NO|YES}] [,REJECT=x]	Defines attributes common to a group of line types on a D/T 3886.
[name]	DIMOD	[IOAREA2=YES] [,RDONLY=YES] [,SEPASMB=YES] [,TRC=YES] [,TYPEFLE={OUTPUT|INPUT}]	Defines a logic module for a device-independent file
[name]	DLINT	LFR=n,LINBEG=n [,IMAGE={NO|YES}] [,NOSCAN=(n,...)] [,FLDn=(n,NCRIT,xxx)] [,EDITn=(xxxxxx,EDCHAR)] [,FREND={NO|YES}]	Describes line types, fields in the line.
[name]	DRMOD	[DEVICE=3886] [,RDONLY=YES] [,SEPASMB=YES] [,SETDEV=YES]	Defines logic modules for a 3886 file.
[name]	DTFCD	DEVADDR=SYSxxx ,IOAREA1=name [,ASOCFLE=filename] [,BLKSIZE=n] [,CONTROL=YES] [,CRDERR=RETRY] [,CTLCHR={ASA|YES}] [,DEVICE=nnnn] [,EOFADDR=name] [,ERROPT={IGNORE|SKIP|name}] [,FUNC=xxx] [,IOAREA2=name] [,IOREG=(r)] [,MODE=xx] [,MODNAME=name] [,OUBLKSZ=n] [,RDONLY=YES] [,RECFORM={FIXUNB|VARUNB|UNDEF}] [,RECSIZE=(r)|name] [,SEPASMB=YES] [,SSELECT=n] [,TYPEFLE={INPUT|OUTPUT|CMBND}] [,WORKA=YES]	Defines a card or 3881 file.

Declarative Macros (cont...)

Name	Operation	Operands	Description
[name]	DTFCN	DEVADDR=SYSxxx ,IOAREA1=name [,BLKSIZE=n] [,INPSIZE=n] [,MODNAME=name] [,RECFORM={FIXUNB|UNDEF}] [,RECSIZE=(r)] [,TYPEFLE={INPUT|OUTPUT|CMBND}] [,WORKA=YES]	Defines a console file.
[name]	DTFDA	DEVADDR=SYSxxx ,BLKSIZE=n ,ERRBYTE=name ,IOAREA1=name ,SEEKADDR=name ,TYPEFLE={INPUT|OUTPUT} [,AFTER=YES] [,CONTROL=YES] [,DEVADDR=SYSxxx] [,DSKXTNT=n] [,ERREXT=YES] [,FEOVD=YES] [,HOLD=YES] [,IDLOC=name] [,KEYARG=name] [,KEYLEN=n] [,LABADDR=name] [,READID=YES] [,READKEY=YES] [,RECFORM=xxxxxx] [,RECSIZE=(r)] [,RELTYPE={DEC|HEX}] [,SEPASMB=YES] [,SRCHM=YES] [,TRLBL=YES] [,VERIFY=YES] [,WRITEID=YES] [,WRITEKY=YES] [,XTNTXIT=name]	Defines a direct access file.
[name]	DTFDI	DEVADDR=SYSxxx ,IOAREA1=name [,CISIZE=n] [,EOFADDR=name] [,ERROPT={IGNORE|SKIP|name}] [,IOAREA2=name] [,IOREG=(r)] [,MODNAME=name] [,RDONLY=YES] [,RECSIZE=n] [,SEPASMB=YES] [,TRC=YES] [,WLRERR=name]	Defines a device-independent file.
[name]	DTFDR	COREXIT=name ,DEVADDR=SYSxxx ,EOFADDR=name ,EXITIND=name ,FRNAME=name ,FRSIZE=n ,HEADER=name ,IOAREA1=name [,BLKSIZE=n] [,DEVICE=3886] [,MODNAME=name]	Defines a 3886 OCR file.

Declarative Macros (cont...)

<table>
<tr><th>Name</th><th>Operation</th><th>Operands</th><th>Description</th></tr>
<tr><td>[name]</td><td>DTFDR
(cont..)</td><td>[,RDONLY=YES]
[,SEPASMB=YES]
[,SETDEV=YES]</td><td>Defines a 3886 OCR file.</td></tr>
<tr><td>[name]</td><td>DTFDU</td><td>EOFADDR=name
,IOAREA1=name
,RECSIZE=n
[,CMDCHN=n]
[,DEVADDR=SYSxxx]
[,DEVICE=3540]
[,ERREXT=YES]
[,ERROPT={IGNORE|SKIP|name}]
[,FEED={YES|NO}]
[,FILESEC=YES]
[,IOAREA2=name]
[,IOREG=(r)]
[,MODNAME=name]
[,RDONLY=YES]
[,SEPASMB=YES]
[,TYPEFLE={INPUT|OUTPUT}]
[,VERIFY=YES]
[,VOLSEQ=YES]
[,WORKA=YES]
[,WRTPROT=YES]</td><td>Defines a diskette file.</td></tr>
<tr><td>[name]</td><td>DTFIS</td><td>DSKXTNT=n
,IOROUT=xxxxxx
,KEYLEN=n
,NRECDS=n
,RECFORM={FIXUNB|FIXBLK}
,RECSIZE=n
[,CYLOFL=n]
[,DEVICE=nnnn]
[,ERREXT=YES]
[,HINDEX=nnnn]
[,HOLD=YES]
[,INDAREA=name]
[,INDSKIP=YES]
[,INDSIZE=n]
[,IOAREAL=name]
[,IOAREAR=name]
[,IOAREAS=name]
[,IOAREA2=name]
[,IOREG=(r)]
[,IOSIZE=n]
[,KEYARG=name]
[,KEYLOC=n]
[,MODNAME=name]
[,MSTIND=YES]
[,RDONLY=YES]
[,SEPASMB=YES]
[,TYPEFLE={RANDOM|SEQNTL
|RANSEQ}]
[,VERIFY=YES]
[,WORKL=name]
[,WORKR=name]
[,WORKS=YES]</td><td>Defines an indexed-sequential file.</td></tr>
<tr><td>[name]</td><td>DTFMR</td><td>DEVADDR=SYSxxx
,IOAREA1=name
[,ADDAREA=n]
[,ADDRESS=DUAL]
[,BUFFERS={25|n}]
[,ERROPT=name]
[,EXTADDR=name]</td><td>Defines a MICR/OCR file.</td></tr>
</table>

Declarative Macros (cont...)

Name	Operation	Operands	Description
[name]	DTFMR (cont..)	[,IOREG=(r)] [,MODNAME=name] [,RECSIZE={80|n}] [,SECADDR=SYSnnn] [,SEPASMB=YES] [,SORTMDE={ON|OFF}]	Defines a MICR/OCR file.
[name]	DTFMT	BLKSIZE=n ,DEVADDR=SYSxxx ,EOFADDR=name ,FILABL={NO|STD|NSTD} ,IOAREA1={name|(r)} [,ASCII=YES] [,BUFFOFF=n] [,CKPTREC=YES] [,ERREXT=YES] [,ERROPT={IGNORE|SKIP|name}] [,HDRINFO=YES] [,IOAREA2={name|(r)}] [,IOREG=(r)] [,LABADDR=name] [,LENCHK=YES] [,NOTEPT={YES|POINTS}] [,RDONLY=YES] [,READ={FORWARD|BACK}] [,RECFORM=xxxxxx] [,RECSIZE={n|(r)}] [,REWIND={UNLOAD|NORWD}] [,SEPASMB=YES] [,TPMARK={YES|NO}] [,TYPEFLE={INPUT|OUTPUT|WORK}] [,VARBLD=(r)] [,WLRERR=name] [,WORKA=YES]	Defines a magnetic tape file.
[name]	DTFOR	COREXIT=name ,DEVADDR=SYSxxx ,EOFADDR=name ,IOAREA1=name [,BLKFAC=n] [,BLKSIZE=n] [,CONTROL=YES] [,DEVICE=xxxxx] [,HEADER=YES] [,HPRMTY=YES] [,IOAREA2=name] [,IOREG=(r)] [,MODNAME=name] [,RECFORM={FIXUNB|FIXBLK|UNDEF}] [,RECSIZE=(n)] [,SEPASMB=YES] [,WORKA=YES]	Defines a 1287 or 1288 optical reader file.
[name]	DTFPH	TYPEFLE={INPUT|OUTPUT} [,ASCII=YES] [,CISIZE=n] [,CCWADDR=name] [,DEVADDR=SYSxxx] [,DEVICE=xxxx] [,HDRINFO=YES] [,LABADDR=name] [,MOUNTED={ALL|SINGLE}] [,XTNTXIT=name]	Defines a Physical IOCS file.

Declarative Macros (cont...)

<table>
<tr><th>Name</th><th>Operation</th><th>Operands</th><th>Description</th></tr>
<tr><td>[name]</td><td>DTFPR</td><td>DEVADDR=SYSxxx
,IOAREA1=name
[,ASOCFLE=filename]
[,BLKSIZE=n]
[,CONTROL=YES]
[,CTLCHR={YES|ASA}]
[,DEVICE=nnn]
[,ERROPT={RETRY|IGNORE|name}]
[,FUNC=xxxx]
[,IOAREA2=name]
[,IOREG=(r)]
[,MODNAME=name]
[,PRINTOV=YES]
[,RDONLY=YES]
[,RECFORM={FIXUNB|VARUNB|UNDEF}]
[,RECSIZE=(r)]
[,SEPASMB=YES]
[,STLIST=YES]
[,TRC=YES]
[,UCS={ON|OFF}]
[,WORKA=YES]</td><td>Defines a printer file.</td></tr>
<tr><td>[name]</td><td>DTFSD</td><td>BLKSIZE={n|MAX}]
,EOFADDR=name
[,CISIZE=n]
[,DELETEFL=NO]
[,DEVADDR=SYSxxx]
[,EOXPTR=name]
[,ERROPT={IGNORE|SKIP|name}]
[,FEOVD=YES]
[,HOLD=YES]
[,IOAREA1=name]
[,IOAREA2=name]
[,IOREG=(r)]
[,LABADDR=name]
[,PWRITE=YES]
[,RECFORM=xxxxxx]
[,RECSIZE={n|(r)}]
[,SEPASMB=YES]
[,TRUNCS=YES]
[,TYPEFLE={INPUT|OUTPUT|WORK}]
[,UPDATE=YES]
[,VARBLD=YES]
[,VERIFY=YES]
[,WLRERR=name]
[,WORKA=YES]</td><td>Defines a sequential disk file.</td></tr>
<tr><td>[name]</td><td>DUMODFx</td><td>x = I for Input
x = O for Output
ERREXT=YES
,ERROPT=YES
[,RDONLY=YES]
[,SEPASMB=YES]</td><td>Defines a logic module for a diskette file.</td></tr>
<tr><td>[name]</td><td>ISMOD</td><td>IOROUT={LOAD|ADD|RETRVE|ADDRTR}
[,CORDATA=YES]
[,CORINDX=YES]
[,ERREXT=YES]
[,HOLD=YES]
[,IOAREA2=YES]
[,RDONLY=YES]
[,RECFORM={FIXUNB|FIXBLK|BOTH}]
[,RPS=SVA]
[,SEPASMB=YES]
[,TYPEFLE={RANDOM|SEQNTL
|RANSEQ}]</td><td>Defines a logic module for an indexed sequential file.</td></tr>
</table>

Declarative Macros (cont...)

Name	Operation	Operands	Description
[name]	MRMOD	[ADDRESS={SINGLE|DUAL}] [,BUFFERS=n] [,SEPASMB=YES]	Defines a logic module for a MICR / OCR file
[name]	ORMOD	[BLKFAC=YES] [,CONTROL=YES] ,DEVICE={1287|1288} [,IOAREA2=YES] [,RECFORM={FIXUNB|FIXBLK|UNDEF}] [,SEPASMB=YES] [,WORKA=YES]	Defines a logic module for a 1287 or 1288 optical reader file.
[name]	PRMOD	[CONTROL=YES] [,CTLCHR={YES|ASA}] [,DEVICE=xxxxx] [,ERROPT=YES] [,FUNC=xxx] [,IOAREA2=YES] [,PRINTOV=YES] [,RDONLY=YES] [,RECFORM={FIXUNB|FIXBLK|UNDEF}] [,SEPASMB=YES] [,STLIST=YES] [,TRC=YES] [,WORKA=YES]	Defines a logic module for a printer file.

Name	Operation	Operands	
[name]	CCB	SYSnnn ,command-list-name [,X'nnnn'] [,senseaddress]	Defines an IOCS command control block.
[name]	CHECK	{filename|(1)} [,control-address|,(0)]	Prevents processing until I/O data transfer is complete
[name]	CLOSE| CLOSER	{filename1|(r1)} [,filename2|,(r2)]...	Deactivates a file.
[name]	CNTRL	{filename|(1)} ,code[,n1][,n2]	Provides non-data device commands.
[name]	DISEN	{filename|(1)}	Stops feeding documents thrugh MICR or OCR devices.
[name]	DSPLY	{filename|(1)} ,(r2),(r3)	Displays document field on 1287 display scope.
[name]	ENDFL	{filename|(0)}	Ends the mode initiated by SETFL.
[name]	ERET	{SKIP|IGNORE|RETRY}	Returns control from your error-processing routine to IOCS.
[name]	ESETL	{filename|(1)}	Ends sequential mode initiated by SETL.
[name]	EXCP	{blockname|(1)} [,REAL]	Request PIOCS to start an I/O operation.
[name]	FEOV	{filename|(1)}	Forces end of volume for magn. tape file.
[name]	FEOVD	{filename|(1)}	Forces end of volume for disk file.
[name]	GENIORB	CCW={name1|(S,name1)|(r1)} {,DEVICE=SYSxxx| LOGUNIT={name2 |(S,name2)|(r2)}} [ADDRESS={name3 |(S,name3)|(r3)}} [,LENGTH=fieldlength] [,ECB={name4 |(S,name4)|(r4)}] [,ERREXIT={name5 |(S,name5)|(r5)}] [,FIXLIST={name6 |(S,name6)|(r6)}] [,FIXFLAG=(option,...)] [,IOFLAG=(option,...)]	Generates an I/O Request Block at execution time.
[name]	GET	{filename|(1)} [,workname|,(0)]	Obtains th next sequential logical record from input file

Imperative Macros (cont...)

<table>
<tr><th>Name</th><th>Operation</th><th>Operands</th><th></th></tr>
<tr><td>[name]</td><td>IORB</td><td>DSECT=YES
or
CCW=name1,DEVICE=SYSxxx
[,ECB=name2]
[,FIXLIST=name3]
[,FIXFLAG=(option,...)]
[,IOFLAG=(option,...)]</td><td>Displays the I/O Request Block or Generates an I/O Request Block at assembly time.</td></tr>
<tr><td>[name]</td><td>LBRET</td><td>{1|2|3}</td><td>Returns control to IOCS after label-processing.</td></tr>
<tr><td>[name]</td><td>LITE</td><td>{filename|(1)}
[,light-switches|,(0)]</td><td>Lights pocket lamps on 1419 or 1275.</td></tr>
<tr><td>[name]</td><td>NOTE</td><td>{filename|(1)}</td><td>Obtains identification for a physical record or logical block.</td></tr>
<tr><td>[name]</td><td>OPEN|
OPENR</td><td>{filename|(1)}
[,filename2|,(r2)],...</td><td>Activates a file.</td></tr>
<tr><td>[name]</td><td>POINTR</td><td>{filename|(1)}
,{address|(0)]</td><td>Repositions a file to a specified record.</td></tr>
<tr><td>[name]</td><td>POINTS</td><td>{filename|(1)}</td><td>Repositions a file to its beginning.</td></tr>
<tr><td>[name]</td><td>POINTW</td><td>{filename|(1)}
,{address|(0)]</td><td>Repositions a file to a specified record.</td></tr>
<tr><td>[name]</td><td>PRTOV</td><td>{filename|(1)},{9|12}
[,routine-name|,(0)]</td><td>Specifies printer action when carriage overflow occurs.</td></tr>
<tr><td>[name]</td><td>PUT</td><td>{filename|(1)}
[,workname|,(0)]
[,STLSP={controlfield|
(r1)}]
[,STLSK={controlfield|
(r2)}]</td><td>Moves (outputs) a logical record to I/O device.</td></tr>
<tr><td>[name]</td><td>PUTR</td><td>{filename|(1)}
[,{workname1|(0)},
{workname2|(2)}]</td><td>Sends message to operator's console, requiring a reply.</td></tr>
<tr><td>[name]</td><td>RDLNE</td><td>{filename|(1)}</td><td>Reads a 1287 journal tape line in correction mode.</td></tr>
<tr><td>[name]</td><td>READ</td><td>{filename|(1)}
{,SQ,{area|(0)}
[,legth|,(r1)|,S]
|,ID
|,KEY
|,OR,{name|(r2)}
|,DR,{name|(r3)|number,
number}
|,MR</td><td>Transfers data from an input file to an area in virtual storage.</td></tr>
<tr><td>[name]</td><td>RELSE</td><td>{filename|(1)}</td><td>Skip the remaining records in a block.</td></tr>
</table>

Imperative Macros (cont...)

<table>
<tr><th>Name</th><th>Operation</th><th>Operands</th><th></th></tr>
<tr><td>[name]</td><td>RESCN</td><td>{filename|(1)}
,(r1),(r2)[,n1][,n2]</td><td>Rescans a field on an OCR document.</td></tr>
<tr><td>[name]</td><td>SECTVAL</td><td>[DDKR={name1|(0)}]
,DVCTYP=name2</td><td>Calculates the sector value for a CKD disk file record.</td></tr>
<tr><td>[name]</td><td>SEOV</td><td>filename</td><td>Forces end-of-volume for a system file on tape.</td></tr>
<tr><td>[name]</td><td>SETDEV</td><td>{filename|(1)}
,{phasename|(r)}</td><td>Changes 3886 format records.</td></tr>
<tr><td>[name]</td><td>SETFL</td><td>{filename|(0)}</td><td>Sets file-load mode in ISAM.</td></tr>
<tr><td>[name]</td><td>SETL</td><td>{filename|(r1)}
,{id-name|(r2)
|KEY|BOF|GKEY}</td><td>Sets sequential retrieval mode in ISAM.</td></tr>
<tr><td>[name]</td><td>TRUNC</td><td>{filename|(1)}</td><td>Writes a short block of records.</td></tr>
<tr><td>[name]</td><td>WAIT</td><td>{blockname|(1)}</td><td>PIOCS waits for an I/O operation to be completed before continuing.</td></tr>
<tr><td>[name]</td><td>WAITF</td><td>{filename1|(r1)}
[,filename2|(r2)],...</td><td>LIOCS waits for an I/O operation to be completed before continuing.</td></tr>
<tr><td>[name]</td><td>WRITE</td><td>{filename|(r1)}
{,{SQ|IPDATE},{area|(0)}
[,length|(r)]
|,AFTER[,EOF]
|,ID
|,KEY
|,NEWKEY
|,RZERO</td><td>Transfers a record from virtual storage to an output file.</td></tr>
</table>

Name	Operation	Operands	Description
[name]	ASPL	[DSECT={NO|YES}]	Generates a mapping DSECT for the ASSIGN macro.

[name]	ASSIGN	ASPL={name1|(r1)}
		,SAVE={name2|(r2)}
			Assigns or releases tape drives dynamically.

[name]	ATTACH	{entrypoint|(S, entrypoint)|(r1)} [,SAVE={savearea |(S,savearea)|(r2)}] [,ABSAVE={absavearea |(S,absavearea)|(r3)}] [,ECB={ecbname |S,ecbname)|(r4)}] [,NAME={name|(S,name) |(r5)}] [,MFG={area|(S,area) |(r6)}]	Initiates a subtask.
[name]	CALL	{entrypoint|(15)} [,(parameterlist)]	Passes control to a specified entry point in another program.
[name]	CANCEL	[ALL]	Terminates a task or subtask.
[name]	CDLOAD	{phasename|(1)} [,PAGE={NO|YES}] [,RETPNF={NO|YES}]	Loads a specified phase into the partition GETVIS area.
[name]	CHAP		Lowers the priority of the issuing subtask.
[name]	CHKPT	SYSnnn, {restart-address|(r1)} [,end-address|,(r2)] [.tpointer|,(r3)] [.dpointer|,(r4)] [.filename|,(r5)]	Records the status of your program for later restarting.
[name]	COMREG	[REG=(r1)]	Places the partition's communication region address into the specified register.
[name]	CPCLOSE	[arglist|(r1)]	Issues a CPCLOSE command to VM/SP to release a print or punch file for output.

[name]	CPCOM	ACMD={addr|(1)},
		LCMD={length|(0)}
			Allows authorized subsystems to submit a CP command to VM/SP.

[name]	DEQ	{rcbname|(0)}	Releases an ENQed resource.
[name]	DETACH	[SAVE={savearea|(1)}]	Terminates (normally) a subtask.

Name	Operation	Operands	Description
[name]	DTL	NAME=resourcename [,CONTROL={E|S}] [,LOCKOPT={1|2|4}] [.KEEP={NO|YES}] [.OWNER={TASK |PARTITION}] [,SCOPE={INT|EXT}] [,VOLID=volume-ID]	Generates a DTL (Define The Lock) control block at assembly time.
[name]	DUMP	[RC={n|(r15)}]	Produces a hexadecimal dump.
[name]	ENQ	{rcbname|(0)}	Protects a resource.
[name]	EOJ	[RC={n|(r15)}]	Ends a job step or subtask.
[name]	EXIT	{AB|IT|MR|OC|PC|TT}	Returns control from your interrupt-checking routine.

[name]	EXTRACT	ID=BDY
		,AREA={name1|(S,name1)
		|(r1)}
		,LEN={length|(r2)}
		[,MFG={name3|(r3)}]
		[,MODE={T|P}]
			Displays partition boundaries.

[name]	EXTRACT	ID=PUB ,AREA={name1|(S,name1) |(r1)} ,LEN={length|(r2)} [,MFG={name3|(r3)}] [,PID={name4|(S,name4) |(r4)}] [,SEL={name5|(S,name5) |(r5)}] [,DISP={name6 |(S,name6)|(r6)}]	Displays unit information from the PUB table.

[name]	FCEPGOUT	{listname|(1)|}
		{beginaddr,endaddr
		[,beginaddr,endaddr]}
			Forces an area to be paged-out.

[name]	FETCH	{phasename|(S,address) |(1)} [,entrypoint |(S,entrypoint)|(0)] [,LIST={listname |(S,listname)|(r1)}] [,SYS=YES] [,DE={NO|YES|VSE [FORM]}] [,MFG={area|(S,area) |(r2)}] [,RET={NO|YES}]	Loads a phase; transfers control to it.
[name]	FREE	{filename|(1)}	Make a previously held track or CI available to other tasks.
[name]	FREEVIS	[ADDRESS={name1|(1)}] [,LENGTH={name2|(0)}] [,SPID={name3|(1)}] [,SVA={YES|NO}]	Releases blocks of virtual storage previously obtained by a GETVIS.

<table>
<tr><th>Name</th><th>Operation</th><th>Operands</th><th>Description</th></tr>
<tr><td>[name]</td><td>GENDTL</td><td>[ADDR={name1|(S,name1)
|(r1)}]
[,CONTROL={E|S}]
[,KEEP={NO|YES}]
[,LENGTH={NO|YES}]
[,LOCKOPT={1|2|4}]
[,NAME={name2
|(S,name2)|(r2)}]
[,OWNER={TASK|PART.}]
[,SCOPE={INT|EXT}]
[,VOLID={name2|
(S,name3)|(R3)}</td><td>Generates a DTL (Define The Lock) control block at execution time.</td></tr>
<tr><td>[name]</td><td>GENL</td><td>phasename1,phasen.2,..
[,ADDRESS={area
|(S,area)|(r1)}
,LENGTH=number]
[,ADDRESS={DYNAMIC
|DYN}[,ERREXIT={addr
|(S,addr)|(r2)}]]
[,DE={OLD|VSE[FORM]}]</td><td>Generates a local directory list in the partition.</td></tr>
<tr><td>[name]</td><td>GETIME</td><td>[STANDARD|BINARY
|TU|MIC]
[,LOCAL|GMT]
[,CLOCK=YES]
[,MFG={area|(S,area)
|(r)}]</td><td>Obtains the time of day.</td></tr>
<tr><td>[name]</td><td>GETVIS</td><td>[ADDRESS={name1|(1)}]
[,LENGTH={name2|(0)}]
[,PAGE={NO|YES}]
[,PFIX={NO|YES}]
[,POOL={NO|YES}]
[,SPID={name3|(1)}]
[,SVA={NO|YES}]
[,TSKSUBP={NO|YES}]</td><td>Obtains a block of virtual storage from a GETVIS area.</td></tr>
<tr><td>[name]</td><td>IJBPUB</td><td></td><td>Generates a mapping DSECT for EXTRACT ID=PUB macro.</td></tr>
<tr><td>[name]</td><td>JDUMP</td><td></td><td>Produces a hexadecimal dump; terminates the main or subtask.</td></tr>
<tr><td>[name]</td><td>JOBCOM</td><td>FUNCT={PUTCOM|GETCOM}
,AREA={address|(r1)}
,LENGTH={length|(r2)}</td><td>Permits communication between jobs or job steps in a partition.</td></tr>
<tr><td>[name]</td><td>LFCB</td><td>SYSxxx,phasename
[,FORMS=xxxx]
[,LPI=n] [,NULMSG]</td><td>Loads the forms-control buffer.</td></tr>
<tr><td>[name]</td><td>LOAD</td><td>{phasename|(S,address)
|(1)}
[,loadpoint
|(S,loadpoint)|(0)]
[,LIST={listname
|(S,listname)|(r1)}]
[,SYS=YES]
[,DE={NO|YES
|VSE[FORM]}]
[,TXT=NO]
[,MFG={area|(S,area)
|(r2)}]
[,RET={NO|YES}]</td><td>Loads specified phase; returns control to calling phase.</td></tr>
</table>

System Control Macros (cont...)

Name	Operation	Operands	Description
[name]	LOCK	{name|(S,name)|(r)} [.FAIL={RETURN|WAITC |WAIT}]	Enqueues a resource access request with protection against disallowed usage.
[name]	MAPBDY	[DSECT={NO|YES}]	Generates a mapping DSECT for the EXTRACT ID=BDY macro.
[name]	MAPPSID		Generates a mapping DSECT for the SUBSID macro.
[name]	MODDTL	ADDR={name1|(S,name1) |(r1)} [,NAME={name2 |(S,name2)|(r2)}] [,CHANGE={ON|OFF}] [,CONTROL={E|S}] [,LOCKOPT={1|2|4}] [,KEEP={NO|YES}] [,OWNER={TASK |PARTITION}] [,SCOPE={INT|EXT}] [,VOLID={name3| (S,name3)|(R3)}]	Modifies a DTL (Define The Lock) control block.
[name]	MVCOM	to,length,{from|(0)}	Modifies communication region.
[name]	PAGEIN	beginaddr,endaddr [,beginaddr,endaddr].. [,ECB={ecbname|(0)}]	Brings specified areas into real storage.
[name]	PAGEIN	{listname|(1)} [,ECB={ecbname|(0)}]	Brings specified areas into real storage.

[name]	PDUMP	{address1|(r1)}
		,{address2|(r2)}
		[,MFG={area|(S,area) |(r3)}]
			Produces a snapshot hexadecimal dump; processing continues next instruction.

[name]	PFIX	beginaddr,endaddr [,beginaddr,endaddr]..	Brings pages into real storage; fixes them
[name]	PFIX	{listname|(1)}	Brings pages into real storage; fixes them
[name]	PFREE	beginaddr,endaddr [,beginaddr,endaddr]..	Decrements a page's PFIX counter by 1.
[name]	PFREE	{listname|(1)}	Decrements a page's PFIX counter by 1.
[name]	POST	[ecbname|(1)] [,SAVE={savearea|(0)}]	Posts an ECB and removes a waiting task from the wait state.
[name]	RCB		Generates a Resource Control Block.
[name]	REALAD	{address|(1)}	Returns a real storage address corresponding to a virtual address.

System Control Macros (cont...)

Name	Operation	Operands	Description
[name]	RELEASE	(SYSnnn[,SYSnnn]...) [,savearea]	Releases programmer logical units
[name]	RELPAG	beginaddr,endaddr [,beginaddr,endaddr]..	Releases specified storage areas
[name]	RELPAG	{listname|(1)}	Releases specified storage areas
[name]	RETURN	(r1[,r2])	Restores registers, returns control to calling program
[name]	RUNMODE		Returns mode information
[name]	SAVE	(r1[,r2])	Saves regs in savearea
[name]	SETIME	{timervalue|(1)} [,tecbname|,(r)] [,PREC]	Sets interval to specified value
[name]	SETPFA	{entryaddress|(0)}	Makes or breaks a linkage to page fault appendage routine
[name]	SETT	{timervalue|(1)}	Sets the task timer to the specified value
[name]	STXIT	{AB|IT|OC|PC|TT} ,{rtnaddr|(0)} ,{savearea|(1)} [,OPTION={DUMP|NODUMP |EARLY}] [,MFG={area|(S,area) |(r)}]	Establishes linkage from supervisor to your interrupt processing routine
[name]	STXIT	{AB|IT|OC|PC|TT}	Terminates linkage from supervisor to your interrupt processing routine
[name]	SUBSID	INQUIRY ,NAME={name1|(S,name1) |(r1)} ,AREA={name2|(S,name2) |(r2)} ,LEN={length|(r3)} [,LVLTEST={NO|YES}] [,MFG={name4|(r4)}]	Retrieves information about the supervisor
[name]	TECB		Generates a timer event control block
[name]	TESTT	[CANCEL]	Tests time elapsed from task timer set by SETT
[name]	TPIN		Deactivates partition
[name]	TPOUT		Reactivates partition
[name]	TTIMER	[CANCEL]	Tests time elapsed from interval timer set by SETIME

System Control Macros (cont...)

Name	Operation	Operands	Description
[name]	UNLOCK	[{name|(S,name) |(r)}|ALL]	Releases a resource that was enqueued by the LOCK macro
[name]	VIRTAD	{address|(1)]	Returns virtual address corresponding to real address
[name]	WAIT	{blockname|(1)}	Sets a task into wait state until an ECB is posted
[name]	WAIT	{ecbname|(1)}	Sets a task into wait state until an ECB is posted
[name]	WAITM	{ecb1,ecb2,.... |listname|(1)}	Sets programs or tasks into wait state until ECB's are posted
[name]	XECBTAB	TYPE={DEFINE|DELETE| CHECK|RESET|DELETALL} ,XECB=xecbname [,XECBADR={xecbfield|) (S,xecbfield)|(r1)}] [,ACCESS={XPOST| XWAIT}] [,MFG={area|(S,area) |(r2)}]	Defines or changes a cross-partition event control block

[name]	XPOST	XECB={xecbname|(1)}
		POINTRG=(14)
			Posts a specified XECB

[name]	XWAIT	XECB={xecbname|(1)}
		POINTRG=(14)
			Waits for a specified XECB to be posted

CDMOD name = IJCabcde		
a	F	RECFORM=FIXUNB (always for INPUT, CMBND, or FUNC=I files)
	V	RECFORM=VARUNB
	U	RECFORM=UNDEF
b	A	CTLCHR=ASA (not specified if CMBND)
	Y	CTLCHR=YES
	C	CONTROL=YES
	Z	CTLCHR or CONTROL not specified
c	B	RDONLY=YES and TYPEFLE=CMBND
	C	TYPEFLE=CMBND
	H	RDONLY=YES and TYPEFLE=INPUT
	I	TYPEFLE=INPUT
	N	RDONLY=YES and TYPEFLE=OUTPUT
	O	TYPEFLE=OUTPUT
d	Z	WORKA and IOAREA2 not specified
	W	WORKA=YES
	I	IOAREA2=YES
	B	WORKA and IOAREA2
	Z	WORKA=YES not specified (CMBND file only)
e	0	DEVICE=2540
	1	DEVICE=1442, 2596
	2	DEVICE=2520
	3	DEVICE=2501
	4	DEVICE=2540 and CRDERR
	5	DEVICE=2520 and CRDERR
	6	DEVICE=3505
	7	DEVICE=3525 and FUNC=R/P or omitted
	8	DEVICE=2560 and FUNC=R/P or omitted
	9	DEVICE=5425 and FUNC=R/P or omitted
	A	DEVICE=3525 and FUNC=RP
	B	DEVICE=3525 and FUNC=RW
	C	DEVICE=3525 and FUNC=PW
	D	DEVICE=3525 and FUNC=I
	E	DEVICE=3525 and FUNC=RPW
	F	DEVICE=2560 and FUNC=RP
	G	DEVICE=2560 and FUNC=RW
	H	DEVICE=2560 and FUNC=PW
	I	DEVICE=2560 and FUNC=I
	J	DEVICE=2560 and FUNC=RPW
	K	DEVICE=5425 and FUNC=RP
	L	DEVICE=5425 and FUNC=RW
	M	DEVICE=5425 and FUNC=PW
	N	DEVICE=5425 and FUNC=I
	O	DEVICE=5425 and FUNC=RPW
	P	DEVICE=3881

DIMOD name = IJJabcde		
a	F	always
b	C	RPS=SVA is not specified
	V	RPS=SVA
c	B	TYPEFLE=OUTPUT (both input and output
	I	TYPEFLE=INPUT
d	I	IOAREA2=YES
	Z	IOAREA2=YES is not specified
e	C	RDONLY=YES
	D	RDONLY=YES is not specified

LIOCS Module Name Versus Options (cont...)

DRMOD name = IJMZabDO		
a	S	SETDEV=YES
	Z	SETDEV=YES not specified
b	R	RDONLY=YES
	Z	RDONLY=YES not specified

DUMODFx name = IJNabcde		
a	D	always
b	I	DUMODFI (Input)
	O	DUMODFO (Output)
c	C	ERROPT=YES and ERREXT=YES
	E	ERROPT=YES
	Z	neither is specified
d	Z	always
e	Y	RDONLY=YES
	Z	RDONLY not specified

ISMOD name = IJHabcde		
a	A	RECFORM=BOTH, IOROUT=ADD or ADDRTR
	B	RECFORM=FIXBLK, IOROUT=ADD or ADDRTR
	U	RECFORM=FIXUNB, IOROUT=ADD or ADDRTR
	Z	RECFORM is not specified. (IOROUT=LOAD or RETRVE)
b	A	IOROUT=ADDRTR
	I	IOROUT=ADD
	L	IOROUT=LOAD
	R	IOROUT=RETRVE
	V	IOROUT=ADDRTR, RPS=SVA
	X	IOROUT=LOAD, RPS=SVA
c	B	TYPEFLE=RANSEQ
	G	IOAREA2=YES, TYPEFLE=SEQNTL or IOROUT=LOAD
	R	TYPEFLE=RANDOM
	S	TYPEFLE=SEQNTL
	Z	neither is specified (IOROUT=LOAD or ADD)
d	B	CORINDX=YES and HOLD=YES
	C	CORINDX=YES
	O	HOLD=YES
	Z	neither is specified
e	F	CORDATA=YES, ERREXT=YES, RDONLY=YES
	G	CORDATA=YES and ERREXT=YES
	O	CORDATA=YES and RDONLY=YES
	P	CORDATA=YES
	S	ERREXT=YES and RDONLY=YES
	T	ERREXT=YES
	Y	RDONLY=YES
	Z	neither is specified

MRMOD name = IJUaZZZZ		
a	S	single address adapter
	D	dual address adapter

LIOCS Module Name Versus Options (cont...)

ORMOD name = IJMabcde		
a	F	RECFORM=FIXUNB
	X	RECFORM=FIXBLK
	U	RECFORM=UNDEF
	D	RECFORM=UNDEF and BLKFAC=YES
b	C	CONTROL=YES
	Z	CONTROL=YES is not specified
c	I	IOAREA=YES
	W	WORKA=YES
	B	both are specified
	Z	neither is specified
d	T	device is in tape mode
	D	device is in document mode
e	Z	always

PRMOD name = IJDabcde		
a	F	RECFORM=FIXUNB
	V	RECFORM=VARUNB
	U	RECFORM=UNDEF
b	A	CTLCHR=ASA
	Y	CTLCHR=YES
	C	CONTROL=YES
	S	STLIST=YES
	Z	none of these is specified
	T	DEVICE=3525 with 2-line printer
	U	DEVICE=2560
	V	DEVICE=5425
c	B	ERROPT=YES and PRINTOV=YES
	P	PRINTOV=YES, DEVICE is not 3525, and ERROPT is not specified
	I	PRINTOV=YES, DEVICE=3525, and FUNC=W[T] or omitted
	F	PRINTOV=YES, DEVICE=3525, and FUNC=RW[T]
	C	PRINTOV=YES, DEVICE=3525, and FUNC=PW[T]
	D	PRINTOV=YES, DEVICE=3525, and FUNC=RPW[T]
	Z	neither PRINTOV nor ERROPT is specified, and DEVICE is not a 3525
	O	PRINTOV=YES not specified, DEVICE=3525, and FUNC=W[T] or omitted
	R	PRINTOV=YES not specified, DEVICE=3525, and FUNC=RW[T]
	S	PRINTOV=YES not specified, DEVICE=3525, and FUNC=PW[T]
	T	PRINTOV=YES not specified, DEVICE=3525, and FUNC=RPW[T]
	E	ERROPT=YES and PRINTOV=YES is not specified
	U	FUNC=W or omitted and DEVICE=2560 or 5425
	V	FUNC=RW and DEVICE=2560 or 5425
	W	FUNC=PW and DEVICE=2560 or 5425
	X	FUNC=RPW and DEVICE=2560 or 5425
d	I	IOAREA2=YES
	Z	IOAREA2=YES is not specified
e	V	RDONLY=YES and WORKA=YES
	W	WORKA=YES
	Y	RDONLY=YES
	Z	neither is specified

Bytes*	Bits	Contents	Function
0-15 (00-0F)			CCB.
8 (8)	0		1 = 2501 double-CCW support.
16 (10)	0		1 = OMR, 0 = omitted. (Note 1)
	1		1 = ERROPT, 0 = omitted. (Note 2)
	2		COBOL open; ignore option.
	3		1 = GET issued, (Note 3) 0 = GET not issued. (Note 7)
	4		DTF table address constants relocated by OPENR.
	5-7		File Association: 000 = READ only 010 = READ/PRINT (Note 4) 101 = READ/PUNCH/PRINT (Note 5) 001 = READ/PUNCH. (Note 5)
17-19 (11-13)			Address of logic module.
20 (14)		X'02' X'05'	DTF type. DTF type for 2560 or 5424/5425.
21 (15)	0		1 = open; 0 = closed.
	1		First time switch.
	2		1 = 1442 or 2596; 0 = other
	3		1 = 2560, 3525, or 5424/5425; 0 = other.
	4		1 = 3505; 0 = other.
	5		1 = 2 I/O areas; 0 = 1 I/O area.
	6		1 = 2520; 0 = other.
	7		1 = 2540; 0 = other.
22 (16)		B'SSF0X010'	Normal command code (not for 2560 or 5424/5425). SS: 00 = pocket1, 01 = pocket2, 10 = pocket3. (Note 6) F : 1 = column binary, (Note 3) 0 = EBCDIC. X : 1 = OMR or RCE, (Note 3) 0 = neither.
		B'H0B00010'	Read command code (2560). H : 0 = hopper1, 1 = hopper2. B : 0 = EBCDIC, 1 = column binary.
		B'HMMM0011'	SS command code (5424/5425). H : 0 = hopper1, 1=hopper2. MMM: 001 = stacker1, 010 = stacker2. 011 = stacker3, 100 = stacker4.
23 (17)			Control command code (not for 2560 or 5425).
		B'H0B00010' B'HMMM0011'	Read command code (2560). SS command code (5424/5425). H : 0 = hopper1, 1=hopper2. MMM: 001 = stacker1. 010 = stacker2. 011 = stacker3. 100 = stacker4.
24-27 (18-1B)			Address of IOAREA2. (If IOAREA2 is not specified, address of IOAREA1.)

DTFCD (Input - Reader) (cont...)

Bytes*	Bits	Contents	Function
28 (1C)	0 1 2–7		1 = 2560; 0 = other. 1 = 5424/5425, 0 = other. Not used.
29–31 (1D–1F)			Address of EOF routine.
32–39 (20–27)			Read CCW (2560). Stacker select CCW (5424/5425).
Bytes 40–49 as used for all files except 2560 and 5424/5425 files.			
40–43 (28–2B)		LA &IOREG,0(14) NOP 0	Load user pointer register.
44–49		MVC 0(&BLKSIZE, 13),0(14)	Move IOAREA to WORKA.
(2C–31)		NOP 0 DC X'0000'	
The following bytes (50–105) are used for 2501 double–CCW support.			
50–55 (32–37)			Unused CCB.
56–71 (38–47)			
72 (48)	0		1 = OMR, 0 = omitted. (Note 1)
	1		1 = ERROPT, 0 = omitted. (Note 2)
	2		COBOL open; ignore option.
	3		1 = GET issued, (Note 3) 0 = GET not issued. (Note 7)
	4		DTF table address constants relocated by OPENR.
			File Association:
	5–7		000 = READ only 010 = READ/PRINT (Note 4) 101 = READ/PUNCH/PRINT (Note 5) 001 = READ/PUNCH. (Note 5)
73–75 (49–4B)			Address of logic module.
76 (4C)		X'02' X'05'	DTF type. DTF type for 2560 or 5424/5425.
77 (4D)	0		1 = open; 0 = closed.
	1		First time switch.
	2		1 = 1442 or 2596; 0 = other.
	3		1 = 2560, 3525, or 5424/5425; 0 = other.

DTFCD (Input - Reader) (cont...)

Bytes*	Bits	Contents	Function
77 (4D)	4		1 = 3505; 0 = other.
	5		1 = 2 I/O areas; 0 = 1 I/O area.
	6		1 = 2520; 0 = other.
	7		1 = 2540; 0 = other.
78 (4E)		B'SSF0X010'	Normal command code (not for 2560 or 5424/5425). SS: 00 = pocket1, 01 = pocket2, 10 = pocket3. (Note 6) F : 1 = column binary, (Note 3) 0 = EBCDIC. X : 1 = OMR or RCE, (Note 3) 0 = neither.
		B'H0B00010'	Read command code (2560, 5424/5425). H : 0 = hopper1, 1 = hopper2. B : 0 = EBCDIC, 1 = column binary.
79 (4F)			Control command code (not for 2560 or 5424/5425).
		B'H0B00010'	Read command code (2560, 5424/5425).
80-83 (50-53)			Address of IOAREA2. (If IOAREA2 is not specified, address or IOAREA1.)
84 (54)	0		1 = 2560; 0 = other.
	1		1 = 5424/5425; 0 = other.
	2-7		Not used.
85-87 (55-57)			Address of EOF routine.
88-95 (58-5F)			Read CCW.
96-99 (60-63)		LA &IOREG,0(14) NOP 0	Load user pointer register.
100-103 (64-67)		MVC 0(&BLKSIZE, 13),0(14) NOP 0	Move IOAREA to WORKA.
104-105 (68-69)		DC X'0000'	
The following bytes (50-57) are used for 3505, and 3525 associated files.			
50-53 (32-35)		DC A(name) B 16(15) B 20(15) DC F'0'	If ERROPT=name (Note 2) If ERROPT=SKIP. If ERROPT=IGNORE. If ERROPT=omitted.
54-57 (36-39)		DC A(ASOCFLE)	Address of associated DTF table. (3525 only). (Note 7)
Bytes 40 onward as used for 2560 and 5424/5425 files.			
40-47 (28-2F)			Stacker select CCW (2560). Read CCW (5424/5425).

DTFCD (Input - Reader) (cont...)

Bytes*	Bits	Contents	Function
48-51 (30-33)		LA &IOREG,0(14) NOP 0	
52-57 (34-39)		MVC 0(&BLKSIZE, 13),0(14) NOP 0 DC X'0000'	Move IOAREA to WORKA.

58-63
(3A-3F)
		CLC 0(L,14),
		64(1)
			Test for end of file.
			L=4 if MODE=E;
			L=2 in other cases.

64-67 (40-43)		DC C'/* ' DC X'0C001022'	End-of-file indicator if MODE=E. In other cases.
68-71 (44-47)		DC A(name) B 16(15) B 20(15) DC F'0'	If ERROPT=name. (Note 2) If ERROPT=SKIP. If ERROPT=IGNORE. If ERROPT=omitted.
The following bytes are added for 2560 or 5424/5425 associated files.			
72-75 (48-4B)		DC A(ASOCFLE)	Address of associated DTF table (Note 7)
76-81 (4C-51)		MVC 0(&BLKSIZE, 14),82(1)	Move card image to IOAREA1.
82 (52)		DC &BLKSIZE .C' '	Buffer for card image.

Notes:

1. OMR only for 3505.

2. ERROPT for 2560, 3505, 3525, or 5424/5425 READ file.

3. 3505, and 3525 with or without CONTROL=YES specified.

4. 2560, 3525, or 5424/5425 with or without CONTROL=YES specified.

5. 2560, 3525, or 5424/5425 without CONTROL=YES specified.

6. Defaults to pocket2 for 3505 and 3525.

7. Present only when 2560, 3525, or 5424/5425 associated files are specified for the input DTF.

* Numbers in parentheses are displacements in hexadecimal notation.

DTFCD (Output - Punch)

Bytes*	Bits	Contents	Function
0-15			CCB.
(00-0F)			
16	0		Not used.
(10)	1		1 = ERROPT; (Note 3)
			0 = omitted.
	2		COBOL open; ignore option.
	3		1 = PUT issued; (Note 2)
			0 = PUT not issued.
	4		DTF table address constants
			relocated by OPENR.
			File Association:
	5-7		000 = PUNCH only
			011 = PUNCH/PRINT (Note 3)
			001 = READ/PUNCH (Note 3)
			101 = READ/PUNCH/PRINT (Note 3)
			100 = PUNCH/INTERPRET (Note 3)
17-19			Address of logic module.
(11-13)			
20 (14)		X'04'	DTF type.
21	0		1 = open; 0 = closed.
(15)	1		First time switch.
	2		1 = CTLCHR.
	3		1 = fixed unblocked.
	4		1 = variable unblocked.
	5		1 = 2 I/O areas.
	6		1 = work area.
	7		1 = 2 CCWs in table.
	7		0 = 1 CCW in table.
22		B'SSF00001'	Normal command code.
(16)			SS: 00 = pocket1;
			01 = pocket2;
			10 = pocket3. (Note 4)
			F : 1 = column binary
			0 = EBCDIC.
		B'HSSS0011'	Normal stacker select command code
			(2560 or 5424/5425).
			H : 0 = hopper1, 1 = hopper2.
			SSS: stacker information.
23			Control command code
(17)			(not for 2560 or 5424/5425).
		B'HSSS0011'	Actual stacker select command code
			(2560 or 5424/5425).
24-27		DC A(IOAREA1+x)	Address of data in IOAREA1.
(18-1B)			
28-31			Bucket. (Note 1)
(1C-1F)			
32-33		LR 12,(RECSIZE)	Undefined records only.
(20-21)		NOPR 0	
34-37		LA &IOREG,4(14)	Load user pointer register.
(22-25)		NOP 0	

DTFCD (Output - Punch) (cont...)

Bytes*	Bits	Contents	Function
38 (26)	0-2 3 4 5 6 7		Not used. 1 = 5424/5425. 1 = 2560. 1 = 3525. 1 = 1442 or 2596. 1 = 2520B1.
39 (27)		DC C' '	Blank for eject last card.
For all files except 2560 and 5424/5425 files.			
40-47 (28-2F)			Punch CCW.
48-55 (30-37)			Eject CCW for last card if 2520.
For 2540 files if CRDERR is specified.			
48-55 (30-37)			Retry CCW.
56-135 (38-87)		DC CL80' '	Save area card image.
For 3525 Punch/Interpret files.			
48-55 (30-37)			Load CCW.
56-63 (38-3F)			Print CCW.
64-127 (40-7F)		DC 64C'	Print buffer.
For 3525 Associated files.			
48-51 (30-33)		DC A(ASOCFLE)	Pointer to associated file.
For 2560 and 5424/5425 files.			
40-47 (28-2F)		DC D'0'	Eject CCW. If FUNC=RP or RPW.
48-55 (30-37)			Stacker select CCW
56-63 (38-3F)			Punch and Feed CCW.
For 2560 Punch/Interpret files.			
64-71 (40-47)			Load print head buffer one CCW.
72-79 (48-4F)			Load print head buffer two CCW.
80-87 (50-57)			Print CCW.
88-151 (58-97)		64C' '	Save area for printing line 2.

DTFCD (Output - Punch) (cont...)

Bytes*	Bits	Contents	Function
For 5424/5425 Punch/Interpret files.			
64-71 (40-47)			Print CCW.
For 2560 and 5424/5425 Associated files.			
64-67 (40-43)		DC A(ASOCFLE)	
68 (44)		DC C' ' DC X'00'	If mode is EBCDIC. If mode is Column Binary.
69- (45-)		DC &BLKSIZE .C' '	Buffer for card image.

Notes:

1. The bucket bytes handle undefined length records.
2. Valid for 2560 or 3525 READ/PUNCH, PUNCH/PRINT, and READ/PUNCH/PRINT files.
3. Valid for 2560 or 3525 only.
4. Defaults to pocket2 for 3525.

* Numbers in parentheses are displacements in hexadecimal notation.

DTFCD (Combined - Reader/Punch)

Bytes	Bits	Contents	Function
0-15 (00-0F)			CCB.
16 (10)	0-1 2 3 4 5-7		Not used. COBOL open; ignore option. Not used. OPENR relocates DTF address constants. Not used.
17-19 (11-13)			Address of logic module.
20 (14)		X'00'	DTF type.
21 (15)			Command code (X'02' for 1442, X'C2' for 2520, 2540).
22 (16)			Command code (X'01' for 1442, X'09' for 2520, 2540).
23 (17)			Command code (X'01' for 1442, X'09' for 2520, 2540).
24-31 (18-1F)			CCW.
32-35 (20-23)			Input area address.
36-39 (24-27)			Output area address.
40-41 (28-29)			Input block size.
42-43 (2A-2B)			Output block size.
44-49 (2C-31)		MVC 0(&BLKS, 13),0(14)	
50-55 (32-37)		MVC 0(&OUBL, 14),0(13)	
56-59 (38-3B)			End-of-file address.
60-67 (3C-43)			Save area.
68-73 (44-49)		MVC 1(&OUBL-1, 13),0(13)	
74-77 (4A-4D)		MVI 0(13),X'40'	
78-79 (4E-4F)			Constant (blanks).
80-83 (50-53)			Constant address (bytes 78-79).

Bytes*	Contents	Function
0-15 (00-0F)		CCB.
16 (10)	X'20' X'08'	COBOL open; ignore option. DTF table address constants relocated by OPENR.

17-19
(11-13)
		<u>Address of logic module:</u>
		GET and PUT logic if TYPEFLE=INPUT;
		PUT logic if TYPEFLE=OUTPUT;
		GET, PUT, and PUTR logic if TYPEFLE=CMBND.

20 (14)	X'03'	DTF type.
21-23 (15-17)		For input and output: not used. For combined: byte 21 contains X'01' bytes 22-23 contain INPSIZE.
24-31 (18-1F)	X'09',IOAREA1, X'00',BLKSIZE	CCW.
End of table if RECFORM=FIXUNB and WORKA not specified.		
The following bytes are added to the table if WORKA is specified.		
32-35 (20-23)	DC A(IOAREA1)	Address of I/O area.
36-39 (24-27)	DC F'0'	Register save area.
40-43 (28-2B)	DC F'0'	Register save area.
End of table if RECFORM=FIXUNB.		
The following bytes are added to the table if RECFORM=UNDEF.		
	DC F'0'	Register save area.
	DC F'0'	Register save area.
	DC H'BLKSIZE'	I/O area size.
	DC AL2(BLKSIZE-1)	For input files only.
The following bytes are added to the table if TYPEFLE=CMBND		
32-35 (20-23)	DC A(IOAREA1 +BLKSIZE)	I/O area address for input.
36-37 (24-25)	DC H'BLKSIZE'	Block size.

* Numbers in parentheses are displacements in hexadecimal notation.

Bytes*	Bits	Function
0-5 (00-05)		CCB indicators.
6-7 (06-07)		Logical class and unit numbers (primary if DUAL addressing).
8 (08)		Zero.
9-11 (09-0B)		CCW address.
12-15 (0C-0F)		Zeros.
16 (10)	0-1	Not used.
	2	COBOL open; ignore option.
	3	Not used.
	4	DTF table address constants relocated by OPENR.
	5-7	Not used.
17-19 (11-13)		Address of logic module.
20 (14)		DTF type = X'0B'
21 (15)		Logic module option switches.
	0	User disengage - 0 = off; 1 = on.
	1	Program sort mode - 0 = no; 1 = yes.
	2	First time switch (after engage) - 0 = no; 1 = yes.
	3	Addressing = DUAL - 0 = no; 1 = yes.
	4	Waiting - 0 = no; 1 = yes.
	5	Read logic indicator - 0 = no; 1 = yes.
	6	Not used.
	7	Supervisor initial read (after open) - 0 = no; 1 = yes.
22-29 (16-1D)		Symbolic filename.
30 (1E)		Open/Close option switch.
	0	Open indicator - 0 = closed; 1 = open.
31-33 (1F-21)		Open/Close option switches.
34-35 (22-23)		Logic module option switches.
36-39 (24-27)		Error information status.
40-41 (28-29)		Length of DTF table.
42-43 (2A-2B)		Device type indicator.
44-45 (2C-2D)		Record type.
46-49 (2E-31)		Reserved for future use.

Bytes*	Bits	Function
50–51 (32–33)		I/O register.
52–55 (34–37)		End–of–file address.
56–59 (38–3B)		IOAREA2/1 address.
60–63 (3C–3F)		Document buffer size.
64–65 (40–41)		Blocking factor/Number of buffers.
66–67 (42–43)		I/O area size.
68–71 (44–47)		Record length.
72–76 (48–4C)		Sense information.
77 (4D)		Supervisor switch.
78–79 (4E–4F)		Logical class and unit numbers (secondary — for DUAL addressing only).
80–81 (50–51)		Register alignment bytes.
82–83 (52–53)		Logical class and unit numbers (primary — for DUAL addressing).
84–87 (54–57)		Document buffer size.
88 (58)		Command code (4C).
89–91 (59–5B)		Address of last byte of first document buffer.
92 (5C)		Command code (4C).
93–95 (5D–5F)		Address of last byte of last document buffer.
96–99 (60–63)		Stacker–select routine address.
100–103 (64–67)		Address of stacker select CCW chain.
104–107 (68–6B)		Current buffer address pointer (Supervisor).
108–111 (6C–6F)		Supervisor count.

Bytes*	Bits	Function
112-113 (70-71)		Number of buffers minus 7.
114-115 (72-73)		Message indicator.
116-119 (74-77)		ERROPT routine address.
120-121 (78-79)		Logical class and unit numbers (secondary — for DUAL addressing only).
122-123 (7A-7B)		Reserved for future use.
124-127 (7C-7F)		Address of last buffer given to user.
128-131 (80-83)		Address of first byte of last buffer.
132-139 (84-8B)		Channel status word (CSW).
140-143 (8C-8F)		Address of active GET record.
144-147 (90-93)		GET counter.
148-159 (94-9F)		Reserved for future use.
For SINGLE Addressing		
160-167 (A0-A7)		CCW - Engage.
168-175 (A8-AF)		CCW - Read.
176-183 (B0-B7)		CCW - Sense.
184-191 (B8-BF)		CCW - NOP.
192-199 (C0-C7)		CCW - Stacker select.
200-207 (C8-CF)		CCW - TIC.
208-215 (D0-D7)		CCW - Control.
216-223 (D8-DF)		CCW - BN.
224-231 (E0-E7)		CCW - Read.
232-239 (E8-EF)		CCW - Sense.

DTFMR (MICR) (cont...)

Bytes*	Bits	Function
240-247 (F0-F7)		CCW - Disengage.
For DUAL Address Adapter		
160-167 (A0-A7)		CCW - Engage.
168-175 (A8-AF)		CCW - Read buffer 1.
176-183 (B0-B7)		CCW - Sense.
184-191 (B8-BF)		CCW - NOP.
192-199 (C0-C7)		CCW - Read buffer 2.
200-207 (C8-CF)		CCW - MOD sense.
208-215 (D0-D7)		CCW - Read buffer 1.
216-223 (D8-DF)		CCW - MOD sense.
224-231 (E0-E7)		CCW - TIC to NOP.
232-239 (E8-EF)		CCW - NOP.
240-247 (F0-F7)		CCW - MOD CTL.
248-255 (F8-FF)		CCW - Stacker select.
256-263 (100-107)		CCW - MOD sense.

* Numbers in parentheses are displacements in hexadecimal notation.

Bytes*	Bits	Function
0–15 (00–0F)		Dummy CCB.
16 (10)	0–1 2 3 4 5–7	Not used. COBOL open; ignore option. Not used. DTF table address constants relocated by OPENR. Not used.
17–19 (11–13)		Address of logic module.
20 (14)		DTF type,(X'09'). DTF type, (X'0A' if HEADER=YES).
21 (15)	 0 1 2 3 4 5 6 7	PIOCS switches. 1 = Open; 0 = Closed. 1 = Input. 1 = Control. 1 = Device is 1287. 1 = Header. Reserved for future use. 1 = RDLNE. Not used.
22 (16)		Not used.
23 (17)	0–6 7	Not used. 1 = LIOCS posts a hopper empty condition to DTF.
24–39 (18–27)		CCB.
40–47 (28–2F)		Sense CCW.
48–51 (30–33)		Lost lines (equipment check).
52–55 (34–37)		After nine retries for journal tape, or after two retries for documents.
56–59 (38–3B)		Wrong–length records.
60–63 (3C–3F)		After four retries for journal tape, or after two retries for documents.
64–67 (40–43)		Keyboard corrections.
68–71 (44–47)		Count of data check errors.
72–75 (48–4B)		Lines marked.
76–79 (4C–4F)		Total lines read (CCW chains executed).

Bytes*	Bits	Function
80 (50)		Error indicators.
	0	1 = EOP.
	1	1 = Lost reference mark indicator.
	2	1 = Late stacker selection.
	3	1 = Non-recovery error.
	4	1 = Equipment check.
	5	1 = Wrong-length record.
	6	1 = Hopper empty.
	7	1 = Data check.
81 (51)		LIOCS switches.
	0	1 = First time.
	1	1 = Two I/O areas.
	2	1 = WORKA=YES
	3	1 = RECFORM=FIXUNB
	4	1 = RECFORM=UNDEF
	5-7	Not used.
82 (52)		Normal command code.
83 (53)		Control command code.
84-87 (54-57)		IOAREA2 address.
88-95 (58-5F)		Read CCW.
96-103 (60-67)		Go to next line CCW.
104-111 (68-6F)		Control CCW.
112-115 (70-73)		EOF address.
116-119 (74-77)		Correction exit address.
120-123 (78-7B)		IOAREA1 address.
124-127 (7C-7F)		DC A(&BLKS-1)
128-129 (80-81)		SR 13,&RECS
130-131 (82-83)		LR &RECS,13
132-133 (84-85)		LR &IOR,13
134-135 (86-87)		Sense.

* Numbers in parentheses are displacements in hexadecimal notation.

DTFDR (3886 Optical Reader)

Bytes*	Bits	Contents	Function
0-15 (00-0F)			CCB.
16 (10)	0-1		Not used.
	2		COBOL open; ignore option.
	3		Not used.
	4		OPENR relocates DTF table addresses.
	5-7		Not used.
17-19 (11-13)			Address of logic module.
20 (14)		X'0C'	DTF type.
21 (15)			PIOCS switches:
	0		1 = open; 0 = closed.
	1	B'1'	Input.
	2-5	B'0000'	Not used.
	6	B'1'	Device is 3886.
	7	B'0'	Not used.
22 (16)			Not used.
23 (17)			LIOCS switches:
	0-4	B'00000'	Not used.
	5		1 = SETDEV.
	6		1 = Control passed to COREXIT.
	7		1 = FR loaded from disk.
24-31 (18-1F)			FR phasename at open time.
32-39 (20-27)			Phasename of currently used FR.
40-43 (28-2B)		X'00000000'	Not used.
44-47 (2C-2F)			Start address of FR area in DTF.
48-51 (30-33)			Address of 4-byte pointer at the end of the FR area in the DTF.
52-55 (34-37)			EOF routine address.
56-63 (38-3F)			Scan CCW.
64-71 (40-47)			Read CCW.
72-79 (48-4F)			Control CCW.
80-87 (50-57)			Load format record CCW.
88-91 (58-5B)			COREXIT routine address.

Bytes*	Bits	Contents	Function
92-95 (5C-5F)			IOAREA1 area address.
96-99 (60-63)			Header area address.
100-103 (64-67)			Exit indicator address.
104 (68)			Start of DR area.
105-107 (69-6B)			Header area address.
108-111 (6C-6F)			Exit indicator address.
112 (70)			Start of FR area.

* Numbers in parentheses are displacements in hexadecimal notation.

Bytes*	Bits	Contents	Function
For printer devices an 8 byte prefix is generated in front of the real DTf. VERSION3 prefix:			
-8 (2)		x'0008'	length of DTFPR extension
-6 (2)		x'0000'	reserved
-4 (4)			address of DTFPR extension
DTFPR			
0–15 (00–0F)			CCB. If 3800 extended buffering is selected, the CCW address is changed by OPEN to point to a DTF extension work area in the user virtual area. CLOSE restores it.
16 (10)	0		1 = 2–line printer; (Notes 3,4) 0 = other.
	1		1 = ERROPT; (Notes 3,4) 0 = omitted.
	2		COBOL open; ignore option.
	3		1 = 3525; 0 = other.
	4		OPENR relocates DTF address constants.
	5–7		3525 Modes: 000 = PRINT only 011 = PUNCH/PRINT (Note 3) 010 = READ/PRINT (Note 3) 101 = READ/PUNCH/PRINT (Note 3) 3800 Modes: (Bits 5 and 6) 11 = TRC=YES specified on DTF. (Note 7) 01 = TRC=Y specified via SETPRT (set by OPEN). (Note 7) non 3525 Modes: (Bit 7) 1 = indicates that Version 3
17–19 (11–13)			Address of logic module. If 3800 extended buffering is selected or the actual device is a PRT1 printer, OPEN changes this address to point to extended buffering logic module IJDPR3 or to the PRT1 logic module IJDPRT in system virtual area. CLOSE restores it.
20 (14)		X'08' X'07'	DTF type. DTF type for 2560 and 5424/5425.
21 (15)	0		1 = open; 0 = closed.
	1		First time switch.
	2		1 = Control character.
	3		1 = Fixed unblocked records.
	4		1 = Variable unblocked records.
	5		1 = Two I/O areas.
	6		1 = Work area.
	7		1 = Print overflow channel 9.
22 (16)		X'09'	Normal command code. (Note 5)
23 (17)		X'09'	Control command code. (Note 5)
24–27 (18–1B)		DC A(IOAREA1+x)	Address of data in IOAREA1.

DTFPR (Printer) (cont...)

Bytes*	Bits	Contents	Function
For printer and card punch devices.			
28-31 (1C-1F)			Bucket. (Note 1)
32-33 (20-21)		LR 12,(RECSIZE) NOPR 0	For undefined records only.
34-37 (22-25)		LA &IOREG,4(14) NOP 0	Only if IOREG=(r).
38-39 (26-27)			Bucket. (Note 2)
40-47 (28-2F)		11,*,X'60',1 9,IOAREA,X'20' ,121	CCW — Set up Selective Tape List Control. (Note 6) STLIST not specified.
48-55 (30-37)		9,IOAREA,X'20' ,121 A(Name) DC A(ASOCFLE)	CCW — STLIST specified. (Note 6) Address of user error routine (for all the 3211-compatible printers identified by device type code PRT1). If ASOCFLE=filename. (Note 3)
For printer devices behind the end of DTFPR (pointed to by the VERSION3 prefix): DTFPR extension			

end+1	
	0
	1
	2-7
			Flag for CTLCHR specification:
			CTLCHR=ASA specified
			CTLCHR=YES specified
			reserved

For the 2560 and 5424/5425 Multifunction Card Machine.			
22 (16)		X'00'	Not used.
23 (17)		B'HHHHHH00'	Print head selection byte. H = 1 specifies the corresponding head.
24-27 (18-1B)			Address of IOAREA1.
28-31 (1C-1F)			Bucket.
32-33 (20-21)		LR 12,(RECSIZE) NOPR 0	For undefined records only.
34-37 (22-25)		LA &IOREG,4(14) NOP 0	Only if IOREG=(r)
38-39 (26-27)			Number of bytes to be printed by the last specified print head.
40-43 (28-2B)		DC A(ASOCFLE) DC F'0'	If FUNC=RW, PW, or RPW. In all other cases.
44 (2C)	0 1 3		1 = 2560. Not used. 1 = print control switch for 2560 associated files.

Bytes	Bits	Contents	Function
For the 2560 and 5424/5425 Multifunction Card Machine (cont'd.)			
	4-7		Not used.
45-47 (2D-2F)		DC 3X'00'	Reserved for future use.
For 2560 simple files.			
48-55 (30-37)			Eject CCW.
56-63 (38-3F)			Load print head buffer CCW.
64-71 (40-47)			Print CCW.
For 2560 associated files.			
48-55 (30-37)			Load print head buffer CCW.
56-63 (38-3F)			Print CCW.
For 5424/5425 files.			
48-55 (30-37)			Print CCW.

Notes:

1. The bucket bytes handle undefined records. Bit 0 of byte 28 at open time determines the mode set of a printer with UCS. If bit 0 is 1, the mode is set so that data checks occur if an invalid character is printed. Otherwise, mode is set to suppress data checks. The use of the UCS parameter determines the setting of this bit. If STLIST=YES, byte 31 saves the STLIST control byte provided by the PUT macro.

2. The 2 byte bucket saves print overflow conditions if CTLCHR=ASA. If STLIST=YES, byte 38 contains the current STLIST control byte. Byte 39 is set by the PUT macro to indicate spacing or skipping. (X'00' no spacing, no skipping; X'01' spacing; X'02' skipping.)

3. Valid for 3525 READ/PRINT, PUNCH/PRINT, and READ/PUNCH/PRINT files.

4. Valid for 3525 PRINT only files.

5. X'05' for 3525, X'09' for other devices.

6. Valid for 1403 only.

7. Valid for 3800 only.

* Numbers in parentheses are displacements in hexadecimal notation.

Bytes*	Bits	Function
8-byte header preceding the DTF		
0-3 4-7		Reserved Address of the Version 3 area of the DTF (Note 1)
0-15 (00-0F)		CCB If the file is on a Disk device, the CCW address in bytes 9-11 (9-B) is changed by OPEN to point to a DTF extension in the user virtual save area. CLOSE restores it. If 3800 extended buffering is selected, the CCW address is changed by OPEN to point to a DTF extension work area in the user virtual area. CLOSE restores it.
16 (10)	0	Not used.
	1	Set by MAINT; indicates that LIOCS must retrieve extents from the VTOC instead of the label cylinder.
	2	COBOL open; ignore option.
	3	X'10' indicates an unlabeled FORTRAN tape.
	4	DTF table address constants relocated by OPENR.
	5	Used by FORTRAN (Sequent. Disk Backspace and Rewind).
	6	1 = ASCII, 0 = EBCDIC.
	7	FORTRAN is calling DTFCP.
17-19 (11-13)		Logic module address. If the file is on a Disk device, OPEN changes this address to point to the logic module residing in the system virtual area (SVA). CLOSE restores it. If 3800 extended buffering is selected, OPEN changes this address to point to the extended buffering logic module IJDPR3 in the system virtual area. CLOSE restores it.
20 (14)		DTF type X'32', except in the case of disk assigned to units other than SYSLNK. In this case, DTFCP open changes it to X'20'.
21 (15)		Open indicators: X'02' input, X'00' output, except for tapes assigned to SYS000 to SYSnnn when X'00' = input and X'08' = output. X'08' DISK=YES indicator.
	0	1 = no rewind, 0 = rewind.
22-28 (16-1C)		Filename (see byte 29).
29 (1D)		Device type code: X'00' = 2311 X'08' = 3340 general X'01' = 2314, 2319 X'09' = 3340 35MB X'04' = 3330-1,-2 X'0A' = 3340 70MB X'05' = 3330-11 X'45' = 3800 with TRC X'07' = 3350 X'90' = 3310, 3370
30-35 (1E-23)		File address for disk; block count if bit 7 of byte 16 is on.
36-37 (24-25)		Volume sequence number or work area.
38 (26)		Open switch.
39 (27)		Sequence number of current extent.
40 (28)		Sequence number of last extent, or X'80' if 1442 punch.

Note: The Version 3 area is described in VSE/Advanced Functions Diagnosis Reference: LIOCS Volume 4, SAM for DASD.

DTFCP (DISK=YES - Compiler) (cont...)

Bytes*	Bits	Function
41 (29)		X'80' indicates request for standard label tape OPEN.
42 (2A)		X'80' device is a 2560. X'40' DTF has been extended into the user virtual save area. X'20' device is a Disk. X'10' device is a tape X'08' device is a printer X'04' device is a punch X'02' device is a reader X'01' RPS is supported
43 (2B)		X'F3' device is a Disk. X'F1' device is a reader. X'F0' device is other type.
44 (2C)	0	1 = input, 0 = output.
	1	1 = eject needed for a reader punch; 0 = no eject.
	2	0 = first pass, 1 = not first pass.
	3	1 = two I/O areas, 0 = one I/O area.
	4	1 = 2540 punch.
	5	1 = SYSLST or SYSPCH.
	6	1 = SYSLST or SYSPCH on output tape.
	7	1 = TLBL is present and tape is labeled.
45–47 (2D–2F)		IOAREA2 address.
48 (30)	0	1 = Always on.
	1	Reserved for future use.
	2	1 = Version 3 DTF.
	3–7	Reserved for future use.
49–51 (31–33)		Reserved for future use.
52–53 (34–35)		Lower head limit.
54–57 (36–39)		Extent upper limit.
58–64 (3A–40)		BBCCHHR seek address or physical block number for FBA Disk.
65–67 (41–43)		EOF address.
68–71 (44–47)		Control bucket CCHH; not used for FBA Disk.
72 (48)		Number of records per track for output, number of record per track +1 for input.
73 (49)		X'00' for output, X'01' for input.
74–75 (4A–4B)		X'0020' for output, X'0018' for input for Disk. X'0008' for 2560 and 5424/5425 output. X'0000' for nondisk device.
76–80 (4C–50)		CCHHR for count field; not used for FBA Disk.
81 (51)		Key length.
82–83 (52–53)		Data length.
84–87 (54–57)		Instruction to load user I/O area address to I/O register.

DTFCP (DISK=YES - Compiler) (cont...)

Bytes*	Bits	Function
End-of-table if DTF is defined for an input file.		
88-111 (58-6F)		Seek, search, TIC CCWs; not used for FBA Disk.
112-119 (70-76)		CCW for Disk input and first CCW for Disk output; not used for FBA Disk. This CCW can be used for other devices if unit is not a Disk.
120-127 (77-7F)		Second CCW for output.
128-151 (80-97)		Verify CCWs for output.
End-of-table if DTF is defined for output file and DEVADDR does not equal SYSPCH.		
152-159 (98-9F)		2540 punch error recovery CCW 1.
160-167 (A0-A7)		2540 punch error recovery CCW 2.
168-231 (A8-E7)		Reserved.
When the CP open initializes the table and determines that the device is a 2540 punch, the following bytes in the table are changed:		
30 (1F)		X'FF' indicator to DTFCP open phases and logic module
32-35 (20-23)		Instruction to load user I/O area to I/O register.
48-55 (30-37)		CCW.
56-63 (38-3F)		2540 punch error recovery CCW 1.
64-71 (40-47)		2540 punch error recovery CCW 2.
72-151 (48-97)		80-byte card image, savearea 1.
152-231 (98-E7)		80-byte card image, savearea 2.
When the CP open initializes the table and determines that the device is a 2560 or 5425, following bytes in the table are changed:		
32-35 (20-23)		Instruction to load user's I/O area to I/O register.
48-55 (30-37)		First output CCW.
56-63 (38-3F)		Second output CCW.
64 (40)		Stacker select character V for ASCII.
65 (41)		Stacker select character W for EBCDIC.

* Numbers in parentheses are displacements in hexadecimal notation.

DTFCP (DISK=NO)

Bytes*	Bits	Function
0-15 (00-0F)		CCB.
16 (10)	0-1	Not used.
	2	COBOL open; ignore option.
	3	Not used.
	4	DTF table address constants relocated by OPENR.
	5	Not used.
	6	1 = ASCII (used only if DISK=YES), 0 = EBCDIC (used only if DISK=YES).
	7	FORTRAN is calling DTFCP.
17-19 (11-13)		Logic module address.
20 (14)		DTF type X'32' except in the case of tape assigned to units SYS000 to SYSnnn. In this case, a DTFCP open phase changes it to X'10'. (Note 1)
21 (15)		Open indicators X'02' input, X'00' output (except for tapes assigned to SYS000 to SYSnnn when it is X'00' input, X'08' output).
22-28 (16-1C)		Filename (see byte 29).
29 (1D)		Device type code: X'45' = 3800 with TRC.
30 (1E)		Indicator to DTFCP open phases and logic module. X'FF' for input files. X'00' for output files.
31 (1F)		Reserved for future use.
32-35 (20-23)		Instruction to load user's I/O area address into I/O register.
36-37 (24-25)		Volume sequence number or work area.
38 (26)		Open switch.
39 (27)		Sequence number of current extent.
40 (28)		Sequence number of last extent, or X'80' if 1442 punch.
41 (29)		X'20'
42 (2A)		X'80' device is a 2560. X'40' device is a 5424/5425. X'10' device is a tape. X'08' device is a printer. X'04' device is a punch. X'02' device is a reader.
43 (2B)		X'F1' device is a reader or tape. X'F0' device is other type.

Note 1. DTF type X'30' found in DOS LIOCS Version 1 only.

DTFCP (DISK=NO) (cont...)

Bytes*	Bits	Function
44 (2C)	0	1 = input, 0 = output.
	1	1 = eject needed for a reader-punch, 0 = no eject.
	2	1 = not first pass, 0 = first pass.
	3	1 = two I/O areas, 0 = one I/O area.
	4	1 = 2540 punch.
	5	1 = SYSLST or SYSPCH.
	6	1 = SYSLST or SYSPCH on output tape.
	7	Reserved for future use.
45-47 (2D-2F)		IOAREA2 address.
48-55 (30-37)		CCW.
End-of-table is DTF is defined as output file and DEVADDR is not equal to SYSPCH.		
56-63 (38-3F)		2540 punch error recovery CCW 1.
64-71 (40-47)		2540 punch error recovery CCW 2.
65-67 (41-43)		EOF address, input only.
End-of-table if DTF is defined as input file.		
72-151 (48-97)		80-byte card image, save area 1.
152-231 (98-E7)		80-byte card image, save area 2.
If device is a 2560 or 5424/5425, bytes 56 onward contain the following information:		
56-63 (38-3F)		Second output CCW.
64 (40)		Stacker select character V for ASCII.
65 (41)		Stacker select character W for EBCDIC.
66-75 (42-4B)		Reserved for future use.
76-235 (4C-EB)		First I/O area.
236-237 (EC-ED)		Reserved.
238-317 (EF-13D)		Second I/O area.
318-319 (13E-13F)		Reserved.

* Numbers in parentheses are displacements in hexadecimal notation.

DTFCP (DISK=parameter omitted)

<table>
<tr><td colspan="3">Input or output table generated when the parameter DISK= is omitted from the DTFCP macro (Tape Resident System Type).</td></tr>
<tr><th>Bytes*</th><th>Bits</th><th>Function</th></tr>
<tr><td>0-15
(00-0F)</td><td></td><td>CCB.</td></tr>
<tr><td>16
(10)</td><td>0-1</td><td>Not used.</td></tr>
<tr><td></td><td>2</td><td>COBOL open; ignore option.</td></tr>
<tr><td></td><td>3</td><td>Not used.</td></tr>
<tr><td></td><td>4</td><td>DTF table address constants relocated by OPENR.</td></tr>
<tr><td></td><td>5</td><td>Not used.</td></tr>
<tr><td></td><td>6</td><td>1 = ASCII (used only if DISK=YES), 0 = EBCDIC (used only if DISK=YES).</td></tr>
<tr><td></td><td>7</td><td>Used by FORTRAN.</td></tr>
<tr><td>17-19
(11-13)</td><td></td><td>Logic module address.</td></tr>
<tr><td>20
(14)</td><td></td><td>DTF type X'31' except in the case of tape assigned to units SYS000 to SYSnnn. In this case DTFCP open phase changes it to X'10'.</td></tr>
<tr><td>21
(15)</td><td></td><td>Open indicators X'02' input, X'00' output (except for tapes assigned to SYS000 to SYSnnn when it is X'00' input, X'08' output).</td></tr>
<tr><td>22-28
(16-1C)</td><td></td><td>Filename.</td></tr>
<tr><td>29 (1D)</td><td></td><td>Device type code: X'45' = 3800 with TRC.</td></tr>
<tr><td>30 (1E)</td><td></td><td>X'00' indicator to DTFCP open phases and logic module.</td></tr>
<tr><td>31
(1F)</td><td>0</td><td>1 = input, 0 = output.</td></tr>
<tr><td></td><td>1</td><td>1 = eject needed for a read punch, 0 = no eject.</td></tr>
<tr><td></td><td>2</td><td>1 = not first pass, 0 = first pass.</td></tr>
<tr><td></td><td>3</td><td>1 = two I/O areas, 0 = one I/O area.</td></tr>
<tr><td></td><td>4</td><td>1 = 2540 punch.</td></tr>
<tr><td></td><td>5</td><td>1 = SYSLST or SYSPCH</td></tr>
<tr><td></td><td>6</td><td>1 = SYSLST or SYSPCH on output tape.</td></tr>
<tr><td></td><td>7</td><td>1 = TLBL specified and tape is labeled</td></tr>
<tr><td>32 (20)</td><td></td><td>Open indicators.</td></tr>
<tr><td>33-35
(21-23)</td><td></td><td>IOAREA2 address.</td></tr>
<tr><td>36-39
(24-27)</td><td></td><td>Instruction to load user's I/O area address into I/O register.</td></tr>
<tr><td>40-47
(28-2F)</td><td></td><td>CCW.</td></tr>
<tr><td colspan="3">End of table if DTF is defined as output file and DEVADDR is not equal to SYSPCH.</td></tr>
<tr><td>48-55
(30-37)</td><td></td><td>2540 punch error recovery CCW 1.</td></tr>
<tr><td>56-63
(38-3F)</td><td></td><td>2540 punch error recovery CCW 2.</td></tr>
<tr><td>57-59
(39-3B)</td><td></td><td>EOF address, input only.</td></tr>
</table>

DTFCP (DISK=parameter omitted) (cont...)

Bytes*	Bits	Function
End of table if DTF is defined as input file.		
64–143 (40–8F)		80–byte card image, save area 1.
144–223 (90–13F)		80–byte card image, save area 2.
For 2560 and 5424/5425 bytes 48 onwards contain the following information:		
48–207 (30–CF)		IOAREA1.
208–209 (D0–D1)		Reserved.
210–369 (D2–171)		IOAREA2.
370–371 (172–173)		Reserved.
372–451 (174–1C3)		Compare area.

* Numbers in parentheses are displacements in hexadecimal notation.

DTFDI (Device Independent)

Bytes*	Bits	Function
0-15 (00-0F)		CCB. If the file is on a Disk, the CCW address in bytes 9-11 (09-0B) is changed by OPEN to point to the DTF extension. CLOSE restores it. If 3800 extended buffering is selected, the CCW address is changed by OPEN to point to a DTF extension work area in the user virtual area. CLOSE restores it.
16 (10)	0-1 2 3 4 5-7	Not used. COBOL open; ignore option. Not used. DTF table address constants relocated by OPENR. Not used.
17-19 (11-13)		Address of logic module. If the file is on a Disk, OPEN changes this address to point to the device independent logic module in the system virtual area. CLOSE restores it. If 3800 extended buffering is selected, OPEN changes the address to point to the extended buffering logic module IJDPR3 in the system virtual area. CLOSE restores it.
20 (14)		DTF Type = X'33'
21 (15)	 0	Open/Close indicators - X'82' =input, X'80' =output. Always set on for no rewind.
22-28 (16-1C)		Symbolic filename.
29 (1D)		Disk or diskette device indicators X'00' = 2311 X'01' = 2314, 2319 X'04' = 3330-1,-2 X'05' = 3330-11 X'07' = 3350 X'08' = 3340 general X'09' = 3340 35MB X'0A' = 3340 70MB. X'90' = 3310, 3370
30-35 (1E-23)		Disk address of format-1 label.
36-37 (24-25)		Disk or diskette volume sequence number.
38 (26)	 0 1-3 4 5-7	Open communications switch. 1 = No more extents — diskettes Not used Always 1 Not used.
39 (27)		Sequence number of current extent.
40 (28)		Sequence number of last extent, or X'80' for 1442 reader punch.
41 (29)		Open indicator = X'20'.

DTFDI (Device Independent) (cont...)

Bytes*	Bits	Function
42 (2A)		Device type indicators:
	0	Unused.
	1	1 = DTF has been extended into the part. GETVIS area
	2	1 = Disk
	3	1 = tape
	4	1 = printer
	5	1 = punch
	6	1 = reader
	7	1 = RPS supported.
43 (2B)		Logic module device indicators: X'F3' = Disk or diskette device X'F1' = reader or tape device X'F0' = other type device
44 (2C)		Logic module option switches
	0	1 = input; 0 = output.
	1	1 = eject for RDR-PCH; 0 = no eject.
	2	1 = not first pass; 0 = first pass.
	3	1 = two I/O areas; 0 = one I/O area.
	4	1 = 2540 Punch.
	5	1 = SYSLST/SYSPCH.
	6	1 = Tape SYSLST/SYSPCH.
	7	1 = ASCII; 0 = EBCDIC code.
45-47 (2D-2F)		Alternate I/O area address.
48 (30)		Logic flags:
	0-1	Reserved.
	2	1 = Version 3 DTF.
	3-4	Reserved.
	5	1 = TRC=YES specified on DTF
	6	1 = TRC in effect
	7	1 = 3800 printer
49-51 (31-33)		Reserved.
52-53 (34-35)		Extent lower head limit.
54-57 (36-39)		Extent upper head limit.
58-64 (3A-40)		Disk seek address. Diskette seek address at byte 60 (3C).
65-67 (41-43)		Users EOF address.
68-72 (44-48)		Control bucket CCHHR. Byte 72 (48) always X'01' for diskettes.
73 (49)		Logic module switches X'01' = input X'00' = output X'00' = both input and output on diskettes.
74-75 (4A-4B)		Logic module constants X'0020' Disk output X'0018' Disk input X'0008' Diskette devices X'0000' Non-Disk devices

DTFDI (Device Independent) (cont...)

Bytes*	Bits	Function
76-80 (4C-50)		Count field CCHHR (0CHR0 for diskettes); not used for FBA Disk.
81 (51)		Key length.
82-83 (52-53)		Data length.
84-87 (54-57)		Instruction to load IOREG with correct I/O area address.
88-103 (58-67)		Seek, Search CCWs; not used for FBA Disk. Seek, Read/Write CCW for diskette files.
104-111 (68-6F)		TIC CCW. NOP CCW for diskette output files; unused for diskette input files.
112-119 (70-77)		Input/output CCW.
120-127 (78-7F)		Second output CCW.
128-151 (80-97)		Verify CCWs for output.
152-159 (98-9F)		Error CCW1.
160-167 (A0-A7)		Error CCW2.
168-231 (A8-E7)		Save area (64 bytes).
232-235 (E8-EB)		DC A(WLRERR) if WLRERR=Address. B 28(15) if ERROPT= omitted. B 25(15) if ERROPT=SKIP. B 28(15) if ERROPT=IGNORE.
236-239 (EC-EF)		DC A(ERROPT) if ERROPT=Address. B 0(15) if ERROPT= omitted. B 24(15) if ERROPT=SKIP. B 28(15) if ERROPT=IGNORE.

* Numbers in parentheses are displacements in hexadecimal notation.

DTFDU (Diskette Unit)

Bytes*	Bits	Contents	Function
0-15 (0-F)			Command Control Block (CCB)
16 (10)	0-3	B'0000'	Not used.
	4		1 = DTF relocated by OPENR.
	5-7	B'000'	Not used.
17-19 (11-13)			Address of logic module.
20 (14)		X'1A'	DTF type for OPEN/CLOSE (X'1A' = diskette file).
		X'21'	(X'21' = DTFPH)
21 (15)	0		1 = Command chained file.
	1-2	B'00'	Not used.
	3		1 = Work area specified.
	4	B'0'	Not used.
	5		1 = Open; 0 = Close.
	6		1 = Input; 0 = Output.
	7		Not used.
22-28 (16-1C)			Filename.
29 (1D)		X'06'	Device type code. (X'06' = 3540).
30-35 (1E-23)		C'00CHR00'	Address of HDR1 label in VTOC
36-37 (24-25)			Volume sequence number.
38 (26)			Open communications byte Input File:
	0		1 = No more extents
	1-2	B'00'	Not used
	3		1 = Exit for user's EOF routine
	4		1 = Next extent on new volume
	5-6		Not used
	7		1 = Extent switch.
			Output File:
	0		1 = No more extents
	1		1 = Extents needed at Close time
	2-3	B'00'	Not used
	4		1 = Next extent on new volume
	5		1 = Extent entered via console
	6-7		Not used.
39 (27)	0		1 = Extent bypassed before file opened (input).
	0-7		Sequence number of current extent opened (output).
40 (28)			Sequence number of last extent opened.
41-43 (29-2B)		X'000000'	Reserved.
44-47 (2C-2F)			Address of IOAREA1.
48-51 (30-33)			Address of last Read/Write CCW in chain.

DTFDU (Diskette Unit) (cont...)

Bytes*	Bits	Contents	Function
52-53 (34-35)		X'0001'	Lower record limit.
54-57 (36-39)		X'00CC00RR'	End-of-data seek address (last record + 1).
58-59 (3A-3B)			Number of records in I/O area (used in short chain processing).
60-63 (3C-3F)		X'00FF0001'	Seek argument (0CHR).
64-67 (40-43)			End-of-file routine address (input); 4X'00' (output).
68-71 (44-47)		X'0049001A'	Seek argument control field.
72 (48)			Command chaining factor.
73 (49)			Switch byte 1.
	0		1 = Not first entry after open.
	1	B'0'	Not used.
	2		1 = In close routine (output).
	3		1 = Error chain to be skipped.
	4		1 = End of extent.
	5-7	B'000'	Not used.
74-75 (4A-4B)			(Record size multiplied by command chain factor) - 1
76-80 (4C-50)		X'FFFFFFFFFF'	Seek argument bucket.
81-83 (51-53)		X'000000'	Reserved.
84-87 (54-57)			Instruction to load user's I/O register (or NOP).
88-91 (58-5B)			Address of current I/O area.
92-95 (5C-5F)			Logical record size.
96-99 (60-63)			Address of last byte of the I/O area.
100 (64)			Logical indicators.
	0		1: ERROPT=address
	1		1: ERROPT=IGNORE
	2		1: ERROPT=SKIP
	3		Not used
	4		1 = Two I/O areas
	5-7		Not used.
101-103 (65-67)			Address of user's error handling routine.
104 (68)			CCW count (write command only).

DTFDU (Diskette Unit) (cont...)

Bytes*	Bits	Contents	Function
105			Allowed operations
(69)	0		1 = Allow read commands
	1		1 = Allow write commands
	2		1 = Suppress unit check on C4/C6
	3–7	B'00000'	Not used.
106		X'00'	Sector factor (X'00'=128).
(6A)			
107		X'00'	Reserved.
(6B)			
108	0		1 = Write protect
(6C)	1		1 = No feed at EOF
	2		1 = Check multivolume sequence
	3		1 = Multivolume file
	4		1 = Verify requested
	5		1 = C6s written (update ERMAP)
	6		1 = Read/Write security
	7	B'0'	Not used.
109–111		X'000000'	Not used.
(6D–6F)			
112–119			Feed CCW.
(70–77)			
120–127			Define ops CCW (output);
(78–7F)			8X'00' (input).
128–135			Seek CCW.
(80–87)			
136–143			TIC CCW.
(88–8F)			
144–X		X=143+8*	Read/Write data CCWs;
		(no. of CCWs)	1, 2, 13, or 26
(90–Y)		Y=8F+8*	Read/Write CCWs.
		(no. of CCWs)	
X+1			NOP CCW (output only).
(Y+1)			

* Numbers in parentheses are displacements in hexadecimal notation.

Bytes	Bits	Function
0-15 (0-F)		CCB.
16 (10)	0	1 = Dequeue old volume extents.
	1	Not used.
	2	1 = File assigned 'IGN' (COBOL).
	3	Not used.
	4	1 = DTF relocated by OPENR.
	5-7	Not used.
17-19 (11-13)		3X'00'
20 (14)		DTF type (X'21').
21 (15)		Open/Close indicators.
	1	1 = Blocked files.
	2	1 = Work file
	3	1 = Work area
	4	1 = Not Version 1 table type.
	5	1 = open; 0 = closed.
	6	1 = input; 0 = output.
	7	1 = User labels specified.
22-28 (16-1C)		Filename (see byte 29).
29 (1D)		Device type code: X'00' = 2311 X'09' = 3340 35MB X'01' = 2314, 2319 X'0A' = 3340 70MB X'04' = 3330-1, -2 X'0B' = 3375 X'05' = 3330-11 X'0C' = 3380 X'07' = 3350 X'90' = FBA device X'08' = 3340 general
30 (1E)		C'F' = EOF indicator for DTFPH.
30-35 (1F-23)		(BCCHHR) Address of F1 label in VTOC (output). (BCCHHR) Address of next DLBL EXTENT record (input).
36-37 (24-25)		Volume sequence number.
38 (26)		Open communications byte. OUTPUT
	0	1 = No more EXTENTS.
	1	1 = EXTENTS for LIOCS at close.
	2	1 = Process trailer labels.
	3	1 = Process header labels.
	4	1 = Next EXTENT on new volume.
	5	1 = EXTENTS entered via console.
	6	1 = Process trailer labels at close.
	7	1 = Check EXTENT for minimum of 2 tracks.
		INPUT
	0	1 = No more EXTENTS.
	1	Not used.
	2	1 = No F1 label, process EXTENTS only.
	3	Not used.
	4	1 = New volume on next EXTENT.
	5	Not used.
	6	1 = Process header labels.
	7	Not used.

Bytes	Bits	Function
39 (27)		Sequence number of current EXTENT being opened.
40 (28)		Sequence number of last EXTENT opened (not a console EXTENT entry).
41-43 (29-2B)		Address of user's label routine.
44 (2C)	2	1 = Version 3 DTF.
45-47 (2D-2F)		Not used.
48-51 (30-33)		CCHH address of user's label track. Initially X'80000000'.
52-53 (34-35)		Lower head limit (HH) X'0000' if type 1; X'00nn' if type 128 (n = head limit).
54-57 (36-39)		EXTENT upper limit (CCHH).
58-59 (3A-3B)		BB seek address: BB = X'0000' if disk device.
60-63 (3C-3F)		EXTENT lower limit (CCHH).
64 (40)		Record number. 1 = input, 0 = output.
65-67 (41-43)		Not used.

68-71 (44-47)		CCHH control bucket.
		CCHH = X'00C80009' if 2311 - type 1
		CCHH = X'00C80013' if 2314 or 2319 - type 1
		CCHH = X'01940012' if 3330 type 1
		CCHH = X'03280012' if 3330-11 - type 1
		CCHH = X'015C0009' if 3340 35MB
		CCHH = X'02B8000B' if 3340 70MB
		CCHH = X'022B001D' if 3350 - type 1
		CCHH = X'03B2000B' if 3375
		CCHH = X'0375000E' if 3380
		PBN = Maximum block size if FBA device
		Note: The last two digits of CCHH are replaced by the current head number when the type 128 version of the device is specified.

72 (48)		Record number.
73 (49)		Not used.
74-75 (4A-4B)		Not used.
76-80 (4C-50)		CCHHR bucket = extent lower limit and record number.
81-83 (51-53)		Not used.

Numbers in parantheses are displacements in hexadecimal notation.

DTFPH (Diskette)

Bytes*	Bits	Function
0-15 (0-F)		CCB.
16 (10)	0 1-3 4 5-7	1 = Dequeue old volume extents Not used 1 = DTF relocated by OPENR Not used.
17-19 (11-13)		3X'00'
20 (14)		DTF type (X'21').
21		Open/close indicators
	0-2 3 4 5 6 7	Not used 1 = Work area 1 = Not Version 1 DTF table type 1 = Open; 0 = Closed 1 = Input; 0 = Output Not used.
22-28 (16-1C)		Filename (see byte 29).
29 (1D)		Device type code (3540 = X'06').
30 (1E)		C'F' = EOF indicator for DTFPH.
30-35 (1E-23)		(OCHR00) Address of HDR1 label in VTOC (output).
36-37 (24-25)		Volume sequence number.
38 (26)		Open communications byte. Input:
	0 1-3 4 5-7	1 = No more extents Not used 1 = New volume or new extent Not used.
		Output:
	0 1 2-3 4 5 6 7	1 = No more extents 1 = Extents for LIOCS at close Not used 1 = New volume on next extent 1 = Extents entered via console Not used 1 = Check extent for minimum of 2 tracks.
39 (27)		Sequence number of current extent being opened.
40 (28)		Sequence number of last extent opened (not a console extent entry).
41-43 (29-2B)		Not used.
44-47 (2C-2F)		Address of IOAREA1.
48-51 (30-33)		Not used.
52-53 (34-35)		X'0000'

Bytes*	Bits	Function
54-57 (36-39)		Extent upper limit (OCHR).
58-59 (3A-3B)		Not used.
60-63 (3C-3F)		Extent lower limit (OCHR).
64 (40)		Record number. 1 = Input, 0 = Output.
65-67 (41-43)		Not used.
68-71 (44-47)		OCHR control bucket. OCHR = X'0049001A' for 3540 (output only).
72 (48)		Record number.
73 (49)		X'80' - verify requested. X'40' - last volume on multivolume file (input). X'10' - multivolume file (input).
74 (4A)		Record size (maximum of 128).
75 (4B)		Not used.
76-80 (4C-50)		OCHR bucket = extent lower limit and record number (output).
81-83 (51-53)		Not used.

* Numbers in parentheses are displacements in hexadecimal notation.

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename	IJICCB	0-15 (0-F)		Command Control Block (CCB).
	IJIMOD	16 (10)	0	1 = Trailer labels
			1	Used by FREE macro
			2	1 = COBOL Open/Ignore option
			3	1 = Track hold option specified
			4	1 = DTF relocated by OPENR
			5	Not used
			6	1 = SPNUNB
			7	Used by CNTRL macro
		17-19 (11-13)		Address of logic module.
		20 (14)		DTF type for OPEN/CLOSE (X'22' = direct access files).
	IJISWI	21 (15)	0	1 = Output; 0 = Input.
			1	1 = Verify option specified.
			2	1 = Search multiple track (SRCHM) specified.
			3	1 = WRITE AFTER or WRITE RZERO macro used.
			4	1 = IDLOC specified.
			5	1 = Undefined; 0 = FIXUNB, VARUNB, or SPNUNB
			6	1 = RELTYPE = DEC
			7	1 = End of file.
	IJIFNM	22-28 (16-1C)		Filename (DTF Name).
	IJIDVTP	29 (1D)		Device Type. X'00' = 2311 X'01' = 2314, 2319 X'04' = 3330-1, 3330-2 X'05' = 3330-11 X'07' = 3350 X'08' = 3340 general X'09' = 3340 35MB X'0A' = 3340 70MB X'0B' = 3375 X'0C' = 3380
	IJIUNT	30-31 (1E-1F)		Starting logical unit address of the first volume containing the data file. This value is supplied by the OPEN from EXTENT cards (can be initially zero).
	IJIRPS	32 (20)	0	Not used
			1	1 = RPS device and RPS=YES in FOPT macro
			2-6	Not used
			7	1 = Extended DTF for RPS
	IJIULB	33-35 (21-23)		Address of user's label routine.
	IJIUXT	36-39 (24-27)		Address of user's routine for processing EXTENT information.

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
	IJIRELPT	40 (28)		Pointer to relative address area: &Filename.P – &Filename 2
	IJIERC	41–43 (29–2B)		Address of a 2–byte field in which IOCS can store the error condition or status codes.
	IJITST	44–45 (2C–2D)		Macro code switch for internal use: X'0000' = READ ID X'0001' = READ KEY X'0002' = WRITE ID X'0003' = WRITE KEY X'0004' = WRITE RZERO X'0005' = WRITE AFTER
	IJIBPT	46–47 (2E–2F)		Pointer to channel program build area (&Filename.B) minus 32.
	IJICB2	48–63 (30–3F)		Control seek CCB.
&Filename.Z	IJICCW	64–71 (40–47)		Control Seek CCW for overlap seek routine.
	IJIXMD	72–75 (48–4B)		Channel program builder instruction: XI 36(2),C'0'
	IJIMSZ	76–77 (4C–4D)		Maximum data length for FIXUNB or UNDEF records;BLKSIZE for VARUNB or SPNUNB records.
	IJISPT	78 (4E)		Pointer to READ ID string (Filename.0); X'00' if no READ ID issued.
		79 (4F)		Pointer to READ KEY string (Filename.1); X'00' if no READ KEY issued.
		80 (50)		Pointer to WRITE ID string (Filename.2); X'00' if no WRITE ID issued.
		81 (51)		Pointer to WRITE KEY string (Filename.3); X'00' if no WRITE KEY issued.
		82 (52)		Pointer to WRITE RZERO string (Filename.4); X'00' if no WRITE RZERO issued.
		83 (53)		Pointer to WRITE AFTER string (Filename.5); X'00' if no WRITE AFTER issued.
		84–87 (54–57)		Reserved

DTFDA (Direct Access) (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
	IJILAT	88 (58)	0	Not used.
			1	1 = Wrong-length record.
			2	1 = Non-data transfer error.
			3	Not used.
			4	1 = No room found.
			5-6	Not used.
			7	1 = Record out of extent area.
		89 (59)	0	1 = Data check in count area.
			1	1 = Track overrun.
			2	1 = End of cylinder.
			3	1 = Data check when reading key or data.
			4	1 = No record found.
			5	1 = End of file.
			6	1 = End of volume.
			7	Not used.
	IJILBTK	90-95 (5A-5F)		Label track address, XBCCHH, where X is the volume sequence number of the device on which the label track is located.
The following section is included if UNDEF, AFTER, or RZERO is specified.				
&Filename.L	IJILST	96-143 (60-8F)		Basic CCWs to build channel program.
		144-183 (90-B7)		Basic CCWs for undefined length or formatting macros.
	IJIVIT	184-185 (B8-B9)		Instruction to give record length to user if record length is undefined. (NOPR 0 if no RECSIZE specified.)
	IJIFRU	186-187 (BA-BB)		Instruction to get record length from user if record record length is undefined. (NOPR 0 if no RECSIZE specified.)
&Filename.F	IJIFLD	188-192 (BC-C0)		Work area (used for R0 address - CCHH0).
&Filename.K	IJICNT	193-200 (C1-C8)		Work area (used for R0 data field).
&Filename.C	IJICTS	201-208 (C9-D0)		Work area (included only for spanned or variable records for record count field).
The channel program builder strings are generated following the DTFDA table, and preceding the channel program building area.				
&Filename.O		Variable		Channel program builder string for READ ID macro. If READ ID is not specified, the string is not generated.

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename.1		Variable		Channel program builder string for READ KEY macro. If READ KEY is not specified, the string is not generated.
&Filename.2		Variable		Channel program builder string for WRITE ID macro. If WRITE ID is not specified, the string is not generated.
&Filename.3		Variable		Channel program builder string for WRITE KEY macro. If WRITE KEY is not specifid, the string is not generated.
&Filename.4		Variable		Channel program builder string for WRITE RZERO macro. If WRITE RZERO or WRITE AFTER is not specified, the string is not generated.
&Filename.5		Variable		Channel program builder string for WRITE AFTER macro. If WRITE RZERO or WRITE AFTER is not specified, the string is not generated.
The following section contains the channel program build areas and varies in size.				
&Filename.B		0–7		Seek CCW that is generated at program assembly time and used by all channel programs.
		Variable		Area to build: 1. Eight CCWs if AFTER is not specified. 2. Eight CCWs if spanned or variable length records and AFTER=YES is specified 3. Seven CCWs if undefined or fixed records and AFTER=YES is specified.
		Variable		Area to build: 1. Eight CCWs if AFTER is not specified and VERIFY=YES is specified. 2. Eight CCWs if spanned or variable length records and AFTER=YES and VERIFY=YES are specified. 3. Five CCWs if undefined or fixed records and AFTER=YES and VERIFY=YES are specified.

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
The following section is added for spanned records only.				
		8 bytes		Count save area.
		8 bytes		SEEKADR save area.
		1 byte	0	1 = Relative addressing.
			1	1 = IJIGET switch on.
			2	1 = Ignore hold switch on.
			3	Reserved for use by IJGXDAV/S
			4	1 = New volume SEEKADR.
			5–7	Not used.
		1 byte		Reserved.
		2 bytes		Record size.
		12 bytes		Work area.
		8 bytes		Control word save area.
The following section is added to the DTFDA table if DSKXTNT (relative addressing) is specified.				
&Filename.P		3 bytes		3X'00' for padding.
&Filename.I		5 bytes		IDLOC record area (bucket used by module).
&Filename.S		8 bytes		SEEKADR in form: M,B1,B2,C1,C2,H1,H2,R
		4 bytes		DC A(&SEEKADR)
		4 bytes		DC A(&IDLOC)
		8 bytes		Work area for RELTYPE=DEC.
&Filename.X		4 bytes		Save area for CCHH portion of actual disk address.
		4 bytes		Alteration factor for C1 in SEEKADR (see bytes 112–119): 2311: X'00000001' 2314, 2319: X'00000001' 3330: X'00001300' 3340, 3375: X'00000C00' 3350: X'00001E00' 3380: X'00000F00'
		4 bytes		Alteration factor for C2 in SEEKADR (see bytes 112–119): 2311: X'0000000A' 2314, 2319: X'00000014' 3330: X'00000013' 3340, 3375: X'0000000C' 3350: X'0000001E' 3380: X'0000000F'
		4 bytes		Alteration factor for H1 in SEEKADR (see bytes 112–119): 2311: X'00000001' 2314, 2319: X'00000001' 3330: X'00000001' 3340: X'00000001' 3350: X'00000001' 3375: X'00000001' 3380: X'00000001'

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
		Variable to end of DTF table		DSKXTNT table composed of a variable number of 8-byte entries containing extent information in the following format: Bytes 0-2 TTT2 - cumulative number of tracks in the DSKXTNT table entries up to and including the current entry. 3 M - vol. sequence number. 4 B - 0 for disk devices. Bytes 5-7 TTT1 - relative track number of lower limit of this entry. A 1-byte end-of-table indicator containing X'FF' follows the last entry in the DSKXTNT table.

Numbers in parentheses are displacements in hexadecimal notation.

Device Independent DTF Extension for DTFDA

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
	IJIXBLD	0 (0)		CCW build area.
	IJIXSPTR	176 (B0)		Address of original channel program.
	IJIXSVMP	180 (B4)		Address of original logic module.
	IJISAVA	184 (B8)		72-byte register save area.
		266 (10A)		Not used.
	IJISECV0	267 (10B)		Sector work byte.
	IJISECV1	268 (10C)		Sector work byte.
	IJISECV2	269 (10D)		Sector work byte.
	IJIXSEC	270 (10E)		RPS CCW.
	IJIXSS0	278 (116)		RPS CCW.
	IJIXSSX	286 (11E)		RPS CCW.
	IJIXSSNF	294 (126)		RPS CCW.
	IJIXSTRG	302 (12E)		PESC byte string area.
	IJIXSPT	382 (17E)		Displacement to strings.
	IJIXMCYL	390 (186)		Maximum cylinders per volume.
	IJIXTFAC	392 (188)		Tolerance factor.
	IJIFLG1	394 (18A)		Flag byte.
	IJIXUSTF	395 (18B)		Indicator needed to use tolerance factor
	IJIFLG2	396 (18C)		Flag byte

Numbers in parentheses are displacements in hexadecimal notation.

DTFPH (DAM Files)

Bytes	Bits	Function
0-15 (0-F)		CCB.
16 (10)		X'08' indicates DTF relocated by OPENR.
17-19 (11-13)		3X'00'
20 (14)		DTF type (X'23').
21 (15)		Option codes.
	0	1 = Output, 0 = Input.
	1-7	Not Used.
22-28 (16-1C)		Filename.
29 (1D)		Device type code: X'00' = 2311 X'01' = 2314, 2319 X'04' = 3330-1, 3330-2 X'05' = 3330-11 X'07' = 3350 X'08' = 3340 general X'09' = 3340 35MB X'0A' = 3340 70MB X'0B' = 3375. X'0C' = 3380.
30-31 (1E-1F)		Logical unit address of first volume containing the file.
32 (20)	0	Not Used.
	1	1 = Device supports RPS.
	2	1 = Version 3 DTF.
	3-7	Reserved for future use.
33-35 (21-23)		Address of user label routine.
36-39 (24-27)		Address of user routine to process EXTENT information.

Numbers in parentheses are displacements in hexadecimal notation.

DTFIS LOAD

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename	IJHKCCB	0–15 (0–F)		Command Control Block (CCB).
		16 (10)	0–1	Not used.
			2	1 = COBOL open ignore option.
			3	Not used.
			4	1 = DTF table address constants relocated by OPENR.
			5	Not used.
			6	1 = Data set security.
			7	1 = Wrong block size error during file extension.
		17–19 (11–13)		Address of logic module.
		20 (14)		File type for OPEN/CLOSE (X'24' = LOAD).
	IJHKOPCO	21 (15)		Option byte.
			0	Not used.
			1	Not used.
			2	1 = Cylinder overflow option.
			3	Not used.
			4	1 = Blocked records (used by previous versions).
			5	1 = Verify.
			6	Not used.
			7	1 = Two I/O areas present.
		22–28 (16–1C)		File name.
	IJHKPDDV	29 (1D)		Prime data device type indicator. X'00' = 2311 X'01' = 2314/2319 X'04' = 3330 X'08' = 3340 general X'09' = 3340 35MB X'0A' = 3340 70MB.
&Filename.C	IJHKCCOD	30 (1E)		Status byte.
			0	1 = Uncorrectable DASD error (except WLR error).
			1	1 = WLR error.
			2	1 = Prime data area full.
			3	1 = Cylinder index area not large enough to reference prime data area. Set on only if error detected at SETFL time.
			4	1 = Master index not large enough to reference prime data area. Set on only if if error detected at SETFL time.
			5	1 = Duplicate record.
			6	1 = Sequence error.
			7	1 = No EOF record written in prime data area.

DTFIS LOAD (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
	IJHKHNDV	31 (1F)		High level index device type indicator. X'00' = 2311 X'01' = 2314/2319 X'04' = 3330 X'08' = 3340 general X'09' = 3340 35MB X'0A' = 3340 70MB.
		32 (20)		Relative position of the DSKXTN logical unit, cell number) table (in words). This value is the length of the DTF table divided by 4.
		33-34 (21-22)		First prime data track in cylinder (HH).
		35 (23)		First prime data record in cylinder (R).
		36-37 (24-25)		Last prime data track in cylinder (HH).
		38 (26)		High record on master index/cylinder index track (R).
	IJHKNRPD	39 (27)		High record on prime data track (R).
		40 (28)		High record on overflow track (R).
	IJHKNRSH	41 (29)		High record on last track index track in cylinder (whether shared or unshared).
	IJHKNRTI	42 (2A)		High record on track index track other than last in cylinder. If only one track index track track in cylinder, it is equal to Byte 41.
	IJHKFLAG	43 (2B)		Condition Code.
			0	1 = WLR checks requested (for extension).
			1	1 = First record in file.
			2	1 = Prime data extent full.
			3	1 = Master index/cylinder index extent too small.
			4	1 = Prime data upper limit has been increased (for extension).
			5	1 = Extension.
			6-7	Not used.
		44-50 (2C-32)		Prime data lower limit (MBBCCHH).
		51-57 (33-39)		Cylinder index lower limit (MBBCCHH).
		58-64 (3A-40)		Master index lower limit (MBBCCHH).

DTFIS LOAD (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
		65 (41)		Switches.
			0-3	Not used.
			4	1 = RPS type device (data).
			5	1 = RPS type DTF.
			6	1 = Master index.
			7	1 = RPS type device (index).
&Filename.H	IJHKLPDR	66-73 (42-49)		Address of last prime data record (MBBCCHHR).
	IJHKLGLN	74-75 (4A-4B)		Logical record length.
		76-77 (4C-4D)		Key length.
	IJHKBKLN	78-79 (4E-4F)		Block length (logical record length times number of records).
		80-81 (50-51)		Overflow record length (logical record length +10).
	IJHKNRCD	82-83 (52-53)		Blocking factor (number of logical records).
		84-85 (54-55)		Index entry length (key length +10).
		86-87 (56-57)		Prime data record length (key length + physical record length).
		88-89 (58-59)		Overflow record length with key (key length + logical record length + 10).
		90-91 (5A-5B)		Prime data record format length (key length + physical record length + 8).
		92-93 (5C-5D)		Overflow record format length (key length + logical record length + 18).
		94-95 (5E-5F)		Key location (in blocked records).

This is the end of the common DTF area. The format of the remainder of the table is variable and is generated according to the parameters specified in the DTFIS macro instruction.

&Filename.S	IJHKSBKT	96-103 (60-67)		Seek/Search address area (MBBCCHHR).
&Filename.P	IJHKLGCT	104-105 (68-69)		Logical record counter (for blocking).
		106-107 (6A-6B)		Number of bytes for high level index.
		108-111 (6C-6F)		Prime data record counter (logical records).

DTFIS LOAD (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
		112 (70)		Status indicators.
			0-1	Not used.
			2	1= File closed.
			3-5	Not used.
			6	1 = Last prime data track full.
			7	1 = Last block full.
	IJHKLTIR	113-117 (71-75)		Last track index normal entry address (CCHHR).
	IJHKLCIR	118-122 (76-7A)		Last cylinder index entry address (CCHHR).
	IJHKLMIR	123-127 (7B-7F)		Last master index entry address (CCHHR).
&Filename.B				CCW build area. See description of SETFL macro, phase1-$$BSETFL.
		128-135 (80-87)		Seek CCW.
		136-143 (88-8F)		Search ID equal CCW.
		144-151 (90-97)		TIC CCW.
	IJHKRDWR	152-159 (98-9F)		Read/Write CCW.
		160-167 (A0-A7)		Search ID equal CCW.
		168-175 (A8-AF)		TIC CCW.
		176-183 (B0-B7)		Verify CCW..
&Filename.M	IJHKADCN	184-187 (B8-BB)		Address of IOAREAL.
		188-191 (BC-BF)		Address of data in WORKL. FIXBLK = address of WORKL; FIXUNB = address of WORKL + key)
		192-195 (C0-C3)		Address of key in WORKL. (FIXBLK = address of WORKL + KEYLOC - 1; FIXUNB = address of WORKL.)
	IJHKBPOS	196-199 (C4-C7)		Block position indicator (addr. of logical record in IOAREAL).
	IJHKMIXT	200 (C8)		Master index, extension indicator.
			0-2	Not used.
			3	1 = Extending file, 0 = Creating file.
			4-6	Not used.
			7	1 = Master index being used, 0 = No master index being used.

DTFIS LOAD (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
		201–204 (C9–CC)		Cylinder index upper limit (CCHH).
		205–208 (CD–D0)		Master index upper limit (CCHH).
	IJHKPDUL	209–215 (D1–D7)		Prime data upper limit (old upper limit, if extension) (MBBCCHH).
		216–222 (D8–DE)		Prime data new upper limit (for extension) (MBBCCHH).
	IJHKLTM1	223 (DF)		Last prime data track in cylinder – 1.
	IJHKKLM1	224–225 (E0–E1)		Key length – 1.
	IJHKLLM1	226–227 (E2–E3)		Logical record length – 1.
	IJHKTIDR	228–229 (E4–E5)		Address of track index dummy record (HR).
	IJHKBFDR	230–231 (E6–E7)		Address of record before first prime data record in cylinder (HR).
	IJHKNRCM	232 (E8)		Number of records on master index/cylinder index track – 1.
	IJHKCMCT	233–236 (E9–EC)		Master index/cylinder index disk address control field (CCHH). 2311 = X'00C70009' 2314/2319 = X'00C70013' 3330 = X'01FF0012' 3340 = X'01FF000C'
	IJHKPDCT	237–239 (ED–EF)		Prime data address control field (CCH). 2311 = X'00C700' 2314/2319 = X'00C700' 3330 = X'01FF00' 3340 = X'01FF00'
	IJHKPDBG	240–242 (F0–F2)		Prime data beginning of volume (CCH). 2311 = X'000100' 2314/2319 = X'000100' 3330 = X'000100' 3340 = X'000100'
	IJHKPDEN	243–245 (F3–F5)		Prime data end of volume (CCH). 2311 = X'00C700' 2314/2319 = X'00C700' 3330 = X'019300' 3340 = X'015B00' (35MB) X'02B700' (70MB)
		246–247 (F6–F7)		Used for alignment.

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename.E	IJHKXTBL	248-251 (F8-FB)		First entry in DSKXTN table (logical unit, cell number). (Note 1)
		256-259 (100-103)		X'FFFFFFFF' = End of DSKXTN table. (Note 2)
		260-263 (104-107)		Address of IOAREA2.
		264-267 (108-10B)		Address used to relocate IOAREA2.

Notes:

1. Each entry in the DSKXTN table is four bytes long. The minimum number of entries is two. There is one entry per extent.

2. Location of the end-of-table indicator depends on length of DSKXTN table.

Numbers in parentheses are displacements in hexadecimal notation.

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename	IJHCCCB	0-15 (0-F)		CCB.
		16 (10)	0-1	Not used.
			2	1 = COBOL open ignore option.
			3	1 = Track hold specified.
			4	1 = DTF table address constants relocated by OPENR.
			5	Not used.
			6	1 = Data set security.
			7	1 = Wrong block size error during addition to file.
		17-19 (11-13)		Logic module address.
		20 (14)		File type for OPEN/CLOSE (X'25' = ADD).
	IJHCOPT	21 (15)		Option byte.
			0	Not used.
			1	1 = Prime data in core.
			2	1 = Cylinder overflow.
			3	1 = Cylinder index in core.
			4	1 = Blocked records.
			5	1 = Verify.
			6-7	Not used.
		22-28 (16-1C)		DTF file name.
	IJHCPDDV	29 (1D)		Prime data device type indicator. X'00' = 2311 X'01' = 2314/2319 X'04' = 3330 X'08' = 3340 general X'09' = 3340 35MB X'0A' = 3340 70MB.
&Filename.C	IJHCSTBY	30 (1E)		Status byte.
			0	1 = Uncorrectable DASD error (except WLR).
			1	1 = WLR error.
			2	1 = EOF (sequential).
			3	1 = No record found.
			4	1 = Illegal ID specified.
			5	1 = Duplicate record sensed.
			6	1 = Overflow area full.
			7	1 = Record retrieved from overflow area.
	IJHCHNDV	31 (1F)		Highest level index device type. X'00' = 2311 X'01' = 2314/2319 X'04' = 3330 X'08' = 3340 general X'09' = 3340 35MB X'0A' = 3340 70MB.
	IJHCPNT	32 (20)		Relative position of the DSKXTNN (logical unit, cell number) table (in words). This value is the length of the DTF table devided by 4.

DTFIS ADD (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
		33-35 (21-23)		First prime data record in cylinder (HHR).
		36-37 (24-25)		Last prime data track in cylinder (HH).
		38 (26)		High record number on master index/cylinder index track (R).
	IJHCPDH	39 (27)		High record number on prime data track (R).
		40 (28)		High record number on overflow track (R).
	IJHCSTH	41 (29)		High record number on shared track (R).
	IJHCTIH	42 (2A)		High record number on track index track (R).
	IJHCRTR	43 (2B)		Retrieval byte.
			0	1 = WORKR area specified.
			1	1 = WORKS area specified.
			2	Overflow switch.
			3	1 = Read.
			4	Not used.
			5	1 = Output.
			6	1 = Write key.
			7	1 = PUT macro issued.
		44-50 (2C-32)		Prime data lower limit (MBBCCHH).
	IJHCCIS	51-57 (33-39)		Cylinder index lower limit (MBBCCHH).
	IJHCMIS	58-64 (3A-40)		Master index lower limit (MBBCCHH).
	IJHCILN	65 (41)		Switches.
			0	1 = From WAITF routine.
			1	1 = WAITF seek check bit.
			2-3	Not used.
			4	1 = RPS type device (data).
			5	1 = RPS type DTF.
			6	1 = Master index.
			7	1 = RPS type device (index).
&Filename.H	IJHCCLPA	66-73 (42-49)		Last prime data record address (MBBCCHHR).
	IJHCRESZ	74-75 (4A-4B)		Logical record length (RECSIZE).
	IJHCKYSZ	76-77 (4C-4D)		Key length (KEYLEN).
	IJHCBLSZ	78-79 (4E-4F)		Block size (logical record length times number of records).
	IJHCRL10	80-81 (50-51)		Overflow record length (logical record length + 10).

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
	IJHCBFAC	82-83 (52-53)		Blocking factor (number of logical records in block (NRECDS)).
		84-85 (54-55)		Index entry length (key length + 10).
	IJHCABCD	86-87 (56-57)		Prime data record length (key length plus physical record length (block size)).
		88-89 (58-59)		Overflow record length plus key (key length + logical record length + 10).
	IJHCCMAX	90-91 (5A-5B)		Prime data record format length (key length + block size + 8).
		92-93 (5C-5D)		Overflow record format length (key length + logical record length + 18).
	IJHCKYLC	94-95 (5E-5F)		Key location (KEYLOC) for blocked records.
		96-97 (60-61)		Constant = 5.
		98-99 (62-63)		Constant = 10.
	IJHCATB2	100-101 (64-65)		Displacement of Part 2 of the DTFIS table from start of Part 1.
	IJHCATB3	102-103 (66-67)		Displacement of Part 3 of the DTFIS table from start of Part 1.
&Filename.S	IJHCSADR	104-113 (68-71)		Seek/search address area (MBBCCHHRFP).
&Filename.W	IJHCBKCT	114-123 (72-7B)		Random/sequential retrieval work area.
&Filename.P	IJHACPRC	124-127 (7C-7F)		Prime data record count.
	IJHACSTI	128 (80)	 0-1 2 3-5 6 7	Status indicators. Not used. 1 = File Closed. Not used. 1 = Last prime data track full. 1 = Block complete.
	IJHACLTA	129-133 (81-85)		Last track index normal entry address (CCHHR).
	IJHACLCA	134-138 (86-8A)		Last cylinder index entry address (CCHHR).
	IJHACLMA	139-143 (8B-8F)		Last master index entry address (CCHHR).

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
	IJHACLOA	144-151 (90-97)		Last independent overflow record address (MBBCCHHR).
&Filename.I	IJHACOTC	152-153 (98-99)		Number of independent overflow tracks.
&Filename.A	IJHACOFC	154-155 (9A-9B)		Number of full cylinder overflow areas.
&Filename.O	IJHACORC	156-157 (9C-9D)		Overflow record count.
	IJHACOLL	158-164 (9E-A4)		Independent overflow area lower limit (MBBCCHH).
	IJHACOUP	165-171 (A5-AB)		Independent overflow area upper limit (MBBCCHH).
	IJHAHRAA	172-175 (AC-AF)		A(&Filename.D) - Address of work area for cylinder overflow control record (COCR).
		176-179 (B0-B3)		A(&Filename.D+8) - Address of work area for the current track index normal entry count field.
		180-183 (B4-B7)		A(&Filename.D+16) - Address of work area for current track index overflow entry count field.
		184-187 (B8-BB)		A(&Filename.D+24) - Address of work area for current prime data record count field.
		188-191 (BC-BF)		A(&Filename.D+32) - Address of work area for current overflow record count field.
		192-195 (C0-C3)		A(&Filename.D+40) - Address of work area for track index normal entry data field.
	IJHADLNK	196-199 (C4-C7)		A(&Filename.D+50) - Address of work area for current overflow record linkage field.
	IJHAARAD	200-203 (C8-CB)		A(&IOAREAL) - Address of IOAREAL, the I/O area used for adding records to a file.
	IJHACUSE	204-207 (CC-CF)		A(&WORKL) - Address of WORKL, work area containing user data records to be added to the file.
	IJHADKEY	208-211 (D0-D3)		A(&Filename.K) - Address of the ADD key area.
		212-215 (D4-D7)		A(&IOAREAL+8) - Address of key position in IOAREAL.
	IJHAKLN8	216-219 (D8-DB)		A(&IOAREAL+8+&KEYLEN) - Address of data position in IOAREAL.

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename.2	IJHCASAD	0-3 (0-3)		A(&Filename.S+3) - Address of the seek/search address area+3.
		4 (4)	0	1 = Seek check indicated.
			1-5	Not used.
			6	1 = Over/under seek has occured.
			7	1 = An error has been found, but a seek check is indicated.
		5-7 (5-7)		A(&Filename.W) -Addr. of random/ sequential retrieval work area.
The following information is generated if the cylinder index in core option is specified.				
	IJHCORST	12-15 (0C-0F)		A(&INDAREA) - Starting address of main storage area specified for cylinder index.
		16-17 (10-11)		AL2(&INDSIZE) - Number of bytes in main storage available for cylinder index.
		18-25 (12-19)		Next cylinder index entry to be read (MBBCCHHR).
		26-30 (1A-1E)		Last cylinder index entry (CCHHR).
	IJHCORBT	31(1F)		Core index byte.
			0	1 = First time through B-transient, $$BINDEX.
			1	1 = End of cyl. index reached.
			2	1 = Index skip option specified.
			3	1 = Suppress in-core option and read cylinder index.
			4-7	Not used.
	IJHCORKY	32-35 (1D-23)		Pointer to key (stored by module).
The following information is generated if the prime data in core add function is specified. This information is aligned on a double word boundary.				
	IJHPSIZE	36-37 (24-25)		Size of IOAREAL.
	IJHPMAX	38-39 (26-27)		Maximum number of prime data records in main storage.
	IJHPDSP1	40-43 (28-2B)		Address of write CCWs.
	IJHPDSP2	44-47 (2C-2F)		Address of read CCWs.
	IJHPSW	48(30)		Switch byte.
			0	1 = EOF.
			1-7	Not used.
		49(31)		Reserved.
	IJHDCWRK	50-51 (32-33)		Work field for I/O module.

DTFIS ADD (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename.B		0-7 (0-7)		CCW X'07', &Filename.S+1, X'40', 6 - Long seek CCW with command chaining.
	IJHCCCW	8-127 (8-7F)		Channel program build area.
&Filename.D	IJHACOCR	128-135 (80-87)		Cylinder overflow control record (COCR).
	IJHACTNA	136-143 (88-8F)		Current track index normal entry count field address.
	IJHACTOA	144-151 (90-97)		Current track index overflow entry count field address.
	IJHACRID	152-159 (98-9F)		Current prime data record count field address.
	IJHACFID	160-167 (A0-A7)		Current overflow record count field address.
	IJHACTIN	168-177 (A8-B1)		Track index normal entry data field.
	IJHACLNK	178-187 (B2-BB)		Current overflow record sequence link field.
	IJHACTIA	188-197 (BC-C5)		Current track index overflow entry data field.
	IJHAGATE	198 (C6)		X'01' - Add to EOF. X'02' - Add to independent overflow area.
		199-201 (C7-C9)		Overflow control bytes (CCH).
	IJHAOCOH	202-203 (CA-CB)		High HR on overflow track.
		204-211 (CC-D3)		Volume upper limit for prime data records (MBBCCHHR).
	IJHAICOM	212-217 (D4-D9)		CLC 0(&KEYLEN,13),0(6) - Unblocked CLC 0(&KEYLEN,13),&KEYLOC-1(6) - Blocked Utility CLC for key.
	IJHAISKY	218-223 (DA-DF)		MVC 0(&KEYLEN,13),0(12) - Unblocked MVC 0(&KEYLEN,13),&KEYLOC-1(12) - Blocked Utility MVC for key.
&Filename.E		224-227 (E0-E3)		First entry in DSKXTN table (logical unit, cell number). (Note 1)
		232-235 (E8-EB)		4X'FF' - End of DSKXTN table. (Note 2)

DTFIS ADD (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename.K		236+ (EC–end)		Key area for ADD only. Number of bytes depends on key length, KEYLEN.

Notes:

1. Each entry in the DSKXTN table is four bytes long. The minimum number of entries is two. There is one entry per extent.

2. Location of the end–of–table indicator depends on length of DSKXTN table.

Numbers in parentheses are displacements in hexadecimal notation.

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename	IJHCCCB	0–15 (0–F)		Command Control Block (CCB).
		16 (10)	0	Not used.
			1	1 = GET issued.
			2	1 = COBOL open ignore option.
			3	1 = HOLD option specified.
			4	1 = DTF table address constants relocated by OPENR.
			5–6	Not used.
			7	1 = Different blocksize in format–1 label than in DTFIS.
		17–19 (11–13)		Address of logic module.
		20 (14)		File type for OPEN/CLOSE (X'26' = RETRVE).
	IJHCOPT	21 (15)		Option byte.
			0	Not used.
			1	1 = Prime data in core.
			2	1 = Cylinder overflow option.
			3	1 = Cyl. index in core option.
			4	1 = Blocked records.
			5	1 = Verify.
			6–7	Not used.
		22–28 (16–1C)		File name (DTF name).
	IJHCPDDV	29 (1D)		Prime data device type. X'00' = 2311 X'01' = 2314/2319 X'04' = 3330 X'08' = 3340 general X'09' = 3340 35MB X'0A' = 3340 70MB.
&Filename.C	IJHCSTBY	30 (1E)		Status byte.
			0	1 = Uncorrectable disk error (except WLR error).
			1	1 = WLR error.
			2	1 = EOF (sequential).
			3	1 = No record found.
			4	1 = Illegal ID specified.
			5	1 = Duplicate record sensed.
			6	1 = Overflow area full.
			7	1 = Record retrieved from overflow area.
	IJHCHNDV	31 (1F)		High level index device type. X'00' = 2311 X'01' = 2314/2319 X'04' = 3330 X'08' = 3340 general X'09' = 3340 35MB X'0A' = 3340 70MB.
	IJHCPNT	32 (20)		Relative position of the DSKXTN logical unit, cell number) table (in words). This value is the length of the DTF table divided by 4.

DTFIS RETRVE, RANDOM (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
		33-35 (21-23)		First prime data record in cylinder (HHR).
		36-37 (24-25)		Last prime data track in cylinder (HH).
		38 (26)		High record number on master index/cylinder index track (R).
	IJHCPDH	39 (27)		High record number on prime data track (R).
		40 (28)		High record number on overflow track (R).
	IJHCSTH	41 (29)		High record number on shared track (R).
	IJHCTIH	42 (2A)		High record number on track index track (R).
	IJHCRTR	43 (2B)		Retrieval byte.
			0	1 = WORKR specified.
			1	1 = WORKS specified.
			2	Overflow switch.
			3	1 = Read key.
			4	Not used.
			5	1 = Output.
			6	1 = Write key.
			7	1 = PUT macro issued.
		44-50 (2C-32)		Prime data lower limit (MBBCCHH).
	IJHCCIS	51-57 (33-39)		Cylinder index lower limit (MBBCCHH).
	IJHCMIS	58-64 (3A-40)		Master index lower limit (MBBCCHH).
	IJHCILN	65 (41)		Switches.
			0	1 = From WAITF routine.
			1	1 = Seek check from WAITF.
			2	1 = Data track held.
			3	1 = Index track held.
			4	1 = RPS type device (data).
			5	1 = RPS type DTF.
			6	1 = Master index.
			7	1 = RPS type device (index).
	IJHCCLPA	66-73 (42-49)		Last prime data record address (MBBCCHHR).
	IJHCRESZ	74-75 (4A-4B)		Logical record length.
	IJHCKYSZ	76-77 (4C-4D)		Key length.
	IJHCBLSZ	78-79 (4E-4F)		Block size (logical record length times number of records).
	IJHCRL10	80-81 (50-51)		Overflow record length (logical record length + 10).

DTFIS RETRVE, RANDOM (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
	IJHCBFAC	82-83 (52-53)		Blocking factor.
		84-85 (54-55)		Index entry length (key length + 10
	IJHCABCD	86-87 (56-57)		Prime data record length (key length + physical record length)
		88-89 (58-59)		Overflow record length with key (key length + logical record length + 10).
	IJHCCMAX	90-91 (5A-5B)		Prime data record format length (key length + physical record length + 8).
		92-93 (5C-5D)		Overflow record format length (key length + logical record length + 18).
	IJHCKYLC	94-95 (5E-5F)		Key location (blocked records).
		96-97 (60-61)		Constant = 5.
		98-99 (62-63)		Constant = 10.
	IJHCATB2	100-101 (64-65)		Displacement of Part 2 of the DTFIS table from Part 1.
	IJHCATB3	102-103 (66-67)		Displacement of Part 3 of the DTFIS table from Part 1.
&Filename.S	IJHCSADR	104-113 (68-71)		Seek/search address area (MBBCCHHRFP).
&Filename.W	IJHCBCKT	114-123 (72-7B)		Random/sequential retrieval work area.

DTFIS RETRVE, RANDOM (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename.2	IJHCASAD	0-3 (0-3)		Address of seek/search address area + 3.
		4 (4)	0 1-5 6 7	1 = Seek check indicated. Not used. 1 = Over/under seek has occurred 1 = An error has been found, but a seek check is indicated.
		5-7 (5-7)		Address of random/sequential retrieval work area.
	IJHSIOAR	8-11 (8-B)		Address of IOAREAS.

DTFIS RETRVE, RANDOM (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
	IJHCRARA	12–15 (C–F)		Address of IOAREAR.
	IJHCRKEY	16–19 (10–13)		Address of KEYARG.
	IJHCRWOR	20–23 (14–17)		Address of WORKR.
	IJHSDB1	24–27 (18–1B)		Current sequential I/O area address.
	IJHSLIOR	28–31 (1C–1F)		4-byte NO-OP instruction, or L IOREG,*-4 if IOREG was specified.
	IJHSLMIT	32 (20)		X'00' = No verify, X'40' = Verify.
		33 (21)		X'08' = Unblocked, X'00' = Blocked.
		34 (22)		R = First prime data record on shared track.
		35–39 (23–27)		Upper limit for sequential retrieval (CCHHR).
	IJHSINIT	40–41 (28–29)		H'0' = Blocked records. H'2' = Overflow record. H'8' = Unblocked records.
		42 (2A)		X'C7' = 2311, 2314, or 2319; X'FF' = 3330, 3340.
		43–47 (2B–2F)		Initial values for sequential retrieval.
&Filename.H	IJHSCADR	48–55 (30–37)		Current disk address for sequential (MBBCCHHR).
	IJHSCOVF	56–63 (38–3F)		Current overflow disk address for sequential (MBBCCHHR).
	IJHSRCNT	64–65 (40–41)		Sequential record counter.
	IJHSTICU	66–67 (42–43)		Current track index entry for sequential (HR).
&Filename.T		68–69 (44–45)		Number of records tagged for deletion.
	IJHRREGS	70–71 (46–47)		Load IOREG for random retrieval.
&Filename.G	IJHRIDSV	72–79 (48–4F)		DASD address save area (MBBCCHHR).
	IJHRADSV	80–83 (50–53)		Record pointer within I/O area for write operation.
&Filename.R	IJHROVCN	84–87 (54–57)		Nonfirst overflow record count.

DTFIS RETRVE, RANDOM (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
The following information is generated when the cylinder index in core option is specified.				
	IJHCORST	92–95 (5C–5F)		A(&INDAREA) – Starting address of main storage area specified for cylinder index.
		96–97 (60–61)		AL2(&INDSIZE) – Number of bytes in main storage available for cylinder index.
		98–105 (62–69)		Next cylinder index entry to be read (MBBCCHHR). (Initialized by $$BINDEX to cylinder index starting address.)
		106–110 (6A–6E)		Last cylinder index entry.
		111 (6F)		Core index byte:
			0	1 = First time through transient
			1	1 = End of index reached.
			2	1 = Index skip option.
			3–7	Not used.
	IJHCORKY	112–115 (70–73)		Pointer to key (stored by the module).
		116–131 (74–83)		Reserved.

DTFIS RETRVE, RANDOM (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename.B		0–7 (0–7)		X'07',&Filename.S+1,X'40',6 – Long seek CCW with command chaining.
	IJHCCCW	8–63 (8–3F)		Area to build CCW string.
&Filename.E		64–67 (40–43)		First entry in DSKXTN table (logical unit, cell number). (Note 1)
		72–75 (48–4B)		4X'FF' End of DSKXTN table. (Note 2)

Notes:

1. The length of one entry is the four bytes shown here. The minimum number of entries is 2. There is one entry per extent.
2. The location of the end-of-table indicator depends on length of DSKXTN table.

Numbers in parentheses are displacements in hexadecimal notation.

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename	IJHCCCB	0-15 (0-F)		Command Control Block (CCB).
		16 (10)	0	Not used.
			1	1 = GET issued.
			2	1 = COBOL open ignore option.
			3	1 = Track hold specified.
			4	1 = DTF table address constants relocated by OPENR.
			5	1 = EOF on sequential retrieve.
			6	1 = Data set security.
			7	1 = Different block size in format-1 label than in DTFIS.
		17-19 (11-13)		Address of logic module.
		20 (14)		File type for OPEN/CLOSE (X'26' = RETRVE).
	IJHCOPT	21 (15)		Option byte.
			0	Not used.
			1	1 = Prime data in core.
			2	1 = Cylinder overflow option.
			3	1 = Cylinder index in core option.
			4	1 = Blocked records.
			5	1 = Verify.
			6	1 = IOAREAS just used, 0 = IOAREA2 just used.
			7	1 = Two I/O areas present.
		22-28 (16-1C)		File name (DTF name).
	IJHCPDDV	29 (1D)		Prime data device type. X'00' = 2311 X'01' = 2314/2319 X'04' = 3330 X'08' = 3340 general X'09' = 3340 35MB X'0A' = 3340 70MB.
&Filename.C	IJHCSTBY	30 (1E)		Status byte.
			0	1 = Uncorrectable DASD error (except WLR error).
			1	1 = WLR error.
			2	1 = EOF (sequential).
			3	1 = No record found.
			4	1 = Illegal ID specified.
			5	1 = Duplicate record sensed.
			6	1 = Overflow area full.
			7	1 = Record retrieved from overflow area.
	IJHCHNDV	31 (1F)		High level index device type. X'00' = 2311 X'01' = 2314/2319 X'04' = 3330 X'08' = 3340 general X'09' = 3340 35MB X'0A' = 3340 70MB.

DTFIS RETRVE, SEQNTL (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
	IJHCPNT	32 (20)		Relative position of the DSKXTN (logical unit, cell number) table (in words). This value is the length of the DTF table devided by 4.
		33-35 (21-23)		First prime data record in cylinder (HHR).
		36-37 (24-25)		Last prime data track in cylinder (HH).
		38 (26)		High record number on master index/cylinder index track (R).
	IJHCPDH	39 (27)		High record number on prime data track (R).
		40 (28)		High record number on overflow track (R).
	IJHCSTH	41 (29)		High record number on shared track (R).
	IJHCTIH	42 (2A)		High record number on track index track (R).
	IJHCRTR	43 (2B)		Retrieval byte.
			0	1 = WORKR specified.
			1	1 = WORKS specified.
			2	Overflow switch.
			3	1 = Read key.
			4	1 = First record being processed (after issuing SETL macro).
			5	1 = Output.
			6	1 = Write key.
			7	1 = PUT macro issued.
		44-50 (2C-32)		Prime data lower limit (MBBCCHH)
	IJHCCIS	51-57 (33-39)		Cylinder index lower limit (MBBCCHH).
	IJHCMIS	58-64 (3A-40)		Master index lower limit (MBBCCHH).
	IJHCILN	65 (41)		Switches.
			0	1 = From WAITF routine.
			1	1 = WAITF seek check bit.
			2-3	Not used.
			4	1 = RPS type device (data).
			5	1 = RPS type DTF.
			6	1 = Master index.
			7	1 = RPS type device (index).
	IJHCCLPA	66-73 (42-49)		Last prime data record address (MBBCCHHR).
	IJHCRESZ	74-75 (4A-4B)		Logical record length.
	IJHCKYSZ	76-77 (4C-4D)		Key length.

DTFIS RETRVE, SEQNTL (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
	IJHCBLSZ	78-79 (4E-4F)		Block size (logical record length times number of records).
	IJHCRL10	80-81 (50-51)		Overflow record length (logical record length + 10).
	IJHCBFAC	82-83 (52-53)		Blocking factor.
		84-85 (54-55)		Index entry length (key length +10).
	IJHCABCD	86-87 (56-57)		Prime data record length (key length + physical record length)
		88-89 (58-59)		Overflow record length with key (key length + logical record length + 10).
	IJHCCMAX	90-91 (5A-5B)		Prime data record format length (key length + physical record length + 8).
		92-93 (5C-5D)		Overflow record format length (key length + logical record length + 18).
	IJHCKYLC	94-95 (5E-5F)		Key location (blocked records).
		96-97 (60-61)		Constant = 5.
		98-99 (62-63)		Constant = 10.
	IJHCATB2	100-101 (64-65)		Displacement of Part 2 of the DTFIS table from Part 1.
	IJHCATB3	102-103 (66-67)		Displacement of Part 3 of the DTFIS table from Part 1.
&Filename.S	IJHCSADR	104-113 (68-71)		Seek/search address area (MBBCCHHRFP).
&Filename.W	IJHCBCKT	114-123 (72-7B)		Random/sequential retrieval work area.

DTFIS RETRVE, SEQNTL (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename.2	IJHCASAD	0-3 (0-3)		Address of seek/search address area + 3.
		4 (4)	0	1 = Seek check indicated.
			1-5	Not used.
			6	1 = Over/under seek has occurred
			7	1 = An error has been found, but a seek check is indicated.

DTFIS RETRVE, SEQNTL (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
		5-7 (5-7)		Address of random/sequential retrieval work area.
	IJHSIOAR	8-11 (8-B)		Address of IOAREAS.
	IJHCRARA	12-15 (C-F)		Address of IOAREA2.
	IJHCRKEY	16-19 (10-13)		Address of KEYARG.
	IJHCRWOR	20-23 (14-17)		Address of WORKR.
	IJHSDB1	24-27 (18-1B)		Current sequential I/O area address.
	IHJSLIOR	28-31 (1C-1F)		L IOREG,*-4 - Load IOREG if IOREG was specified, or a 4-byte NO-OP instruction.
	IJHSLMIT	32 (20)		X'00' = No verify, X'40' = Verify.
		33 (21)		X'08' = Unblocked records, X'00' = Blocked records.
		34 (22)		R = First prime data record on shared track.
		35-39 (23-27)		Upper limit for sequential retrieval (CCHHR).
	IJHSINIT	40-41 (28-29)		H'0' = Blocked records, H'2' = Overflow record, H'8' = Unblocked records.
		42 (2A)		X'C7' = 2311, 2314, or 2319; X'FF' = 3330, 3340.
		43-47 (2B-2F)		Initial values for sequential (CCHHR).
&Filename.H	IJHSCADR	48-55 (30-37)		Current disk address for sequential retrieval (MBBCCHHR).
	IJHSCOVF	56-63 (38-3F)		Current overflow disk address (MBBCCHHR).
	IJHSRCNT	64-65 (40-41)		Sequential record counter.
	IJHSTICU	66-67 (42-43)		Current track index entry (HR).
&Filename.T		68-69 (44-45)		Number of records tagged for deletion.
		70-75 (46-4B)		For boundary alignment.
		76-91 (4C-5B)		Reserved.

DTFIS RETRVE, SEQNTL (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename.B		0-7 (0-7)		X'07',&Filename.S+1, X'40', 6 - Long seek CCW with command chaining.
	IJHCCCW	8-63 (8-3F)		Area to build CCW string.
&Filename.E		64-67 (40-43)		First entry in DSKXTN table (logical unit, cell number). (Note 1)
		72-75 (48-4B)		4X'FF' - End of DSKXTN table. (Note 2)

Notes:

1. The length of one entry is the four bytes shown here. The minimum number of entries is 2. There is one entry per extent.
2. The location of the end-of-table indicator depends on length of DSKXTN table.

Numbers in parentheses are displacements in hexadecimal notation.

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename	IJHCCCB	0-15 (0-F)		CCB.
		16 (10)	0	Not used.
			1	1 = GET issued.
			2	1 = COBOL open ignore option.
			3	1 = Track hold option specified.
			4	1 = DTF table address constants relocated by OPENR.
			5	EOF switch.
			6	1 = Data set security.
			7	1 = Wrong block size error during addition to file.
		17-19 (11-13)		Logic module address.
		20 (15)		File type for OPEN/CLOSE (X'27' = ADDRTR).
	IJHCOPT	21 (15)		Option byte.
			0	Not used.
			1	1 = Prime data in core.
			2	1 = Cylinder overflow.
			3	1 = Cylinder index in core.
			4	1 = Blocked records.
			5	1 = Verify.
			6	1 = IOAREAS just used, 0 = IOAREA2 just used.
			7	1 = Two I/O areas present.
		22-28 (16-1C)		DTF file name.
	IJHCPDDV	29 (1D)		Prime data device type indicator. X'00' = 2311 X'01' = 2314/2319 X'04' = 3330 X'08' = 3340 general X'09' = 3340 35MB X'0A' = 3340 70MB.
&Filename.C	IJHCSTBY	30 (1E)		Status byte.
			0	1 = Uncorrectable disk error (except WLR).
			1	1 = WLR error.
			2	1 = EOF (sequential).
			3	1 = No record found.
			4	1 = Illegal ID specified.
			5	1 = Duplicate record sensed.
			6	1 = Overflow area full.
			7	1 = Record retrieved from overflow area.
	IJHCHNDV	31 (1F)		Highest level index device type. X'00' = 2311 X'01' = 2314/2319 X'04' = 3330 X'08' = 3340 general X'09' = 3340 35MB X'0A' = 3340 70MB.

DTFIS ADDRTR (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
	IJHCPNT	32 (20)		Relative position of the DSKXTN (logical unit, cell number) table (in words). This value is the length of the DTF table devided by 4.
		33–35 (21–23)		First prime data record in cylinder (HHR).
		36–37 (24–25)		Last prime data track in cylinder (HH).
		38 (26)		High record number on master index/cylinder index track (R).
	IJHCPDH	39 (27)		High record number on prime data track (R).
		40 (28)		High record number on overflow track (R).
	IJHCSTH	41 (29)		High record number on shared track (R).
	IJHCTIH	42 (2A)		High record number on track index (TI) track (R).
	IJHCRTR	43 (2B)		Retrieval byte.
			0	1 = WORKR area specified.
			1	1 = WORKS area specified.
			2	Overflow switch.
			3	1 = Read.
			4	1 = First record being processed (after issuing SETL macro).
			5	1 = Output.
			6	1 = Write key.
			7	1 = PUT macro issued.
		44–50 (2C–32)		Prime data lower limit (MBBCCHH).
	IJHCCIS	51–57 (33–39)		Cylinder index lower limit (MBBCCHH).
	IJHCMIS	58–64 (3A–40)		Master index lower limit (MBBCCHH).
	IJHCILN	65 (41)		Switches.
			0	1 = From WAITF routine
			1	1 = Seek check from WAITF.
			2	1 = Data track held.
			3	1 = Index track held.
			4	1 = RPS type device (data).
			5	1 = RPS type DTF.
			6	0 = Cylinder index. 1 = Master Index.
			7	1 = RPS type device (index).
&Filename.H	IJHCCLPA	66–73 (42–49)		Last prime data record address (MBBCCHHR).
	IJHCRESZ	74–75 (4A–4B)		Logical record length (RECSIZE).

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
	IJHCKYSZ	76-77 (4C-4D)		Key length (KEYLEN).
	IJHCBLSZ	78-79 (4E-4F)		Block size (logical record length times number of records).
	IJHCRL10	80-81 (50-51)		Overflow record length (logical record length + 10).
	IJHCBFAC	82-83 (52-53)		Blocking factor (number of logical records in block (NRECDS)).
		84-85 (54-55)		Index entry length (key length + 10).
	IJHCABCD	86-87 (56-57)		Prime data record length (key length plus physical record length (block size)).
		88-89 (58-59)		Overflow record length with key (key length + logical record length + 10).
	IJHCCMAX	90-91 (5A-5B)		Prime data record format length (key length + block size + 8).
		92-93 (5C-5D)		Overflow record format length (key length + logical record length + 18).
	IJHCKYLC	94-95 (5E-5F)		Key location (KEYLOC) for blocked records.
		96-97 (60-61)		Constant = 5.
		98-99 (62-63)		Constant = 10.
	IJHCATB2	100-101 (64-65)		Displacement of Part 2 of the DTFIS table from start of Part 1
	IJHCATB3	102-103 (66-67)		Displacement of Part 3 of the DTFIS table from start of Part 1
&Filename.S	IJHCSADR	104-113 (68-71)		Seek/search address area.
&Filename.W	IJHCBKCT	114-123 (72-7B)		Random/sequential retrieval work area.
&Filename.P	IJHACPRC	124-127 (7C-7F)		Prime data record count.
	IJHACSTI	128 (80)	 0-1 2 3-5 6 7	Status indicators. Not used. 1 = File closed. Not used. 1 = Last prime data track full. 1 = Block complete.
	IJHACLTA	129-133 (81-85)		Last track index normal entry address (CCHHR).

DTFIS ADDRTR (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
	IJHACLCA	134–138 (86–8A)		Last cylinder index entry address (CCHHR).
	IJHACLMA	139–143 (8B–8F)		Last master index entry address (CCHHR).
	IJHACLOA	144–151 (90–97)		Last independent overflow record address (MBBCCHHR).
&Filename.I	IJHACOTC	152–153 (98–99)		Number of independent overflow tracks.
&Filename.A	IJHACOFC	154–155 (9A–9B)		Number of full cylinder overflow areas.
&Filename.O	IJHACORC	156–157 (9C–9D)		Overflow record count.
	IJHACOLL	158–164 (9E–A4)		Independent overflow area lower limit (MBBCCHH).
	IJHACOUP	165–171 (A5–AB)		Independent overflow area upper limit (MBBCCHH).
	IJHAHRAA	172–175 (AC–AF)		A(&Filename.D) – Address of work area for cylinder overflow control record (COCR).
		176–179 (B0–B3)		A(&Filename.D+8) – Address of work area for the current track index normal entry count field.
		180–183 (B4–B7)		A(&Filename.D+16) – Address of work area for current track index overflow entry count field.
		184–187 (B8–BB)		A(&Filename.D+24) – Address of work area for current prime data record count field.
		188–191 (BC–BF)		A(&Filename.D+32) – Address of work area for current overflow record count field.
		192–195 (C0–C3)		A(&Filename.D+40) – Address of work area for track index normal entry data field.
	IJHADLNK	196–199 (C4–C7)		A(&Filename.D+50) – Address of work area for current overflow record sequence-link field.
	IJHAARAD	200–203 (C8–CB)		A(&IOAREAL) – Address of IOAREAL, the I/O area used for adding records to a file.
	IJHACUSE	204–207 (CC–CF)		A(&WORKL) – Address of WORKL, work area containing user data records to be added to the file.
	IJHADKEY	208–211 (D0–D3)		A(&Filename.K) – Address of the ADD key area.

DTFIS ADDRTR (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
		212-215 (D4-D7)		A(&IOAREAL+8) - Address of key position in IOAREAL.
	IJHAKLN8	216-219 (D8-DB)		A(&IOAREAL+8+&KEYLEN) - Address of data position in IOAREAL.

DTFIS ADDRTR (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename.2	IJHCASAD	0-3 (0-3)		A(&Filename.S+3) - Address of the seek/search address area+3.
		4 (4)	0	1 = Seek check indicated.
			1-5	Not used.
			6	1 = Over/under seek has occurred.
			7	1 = An error has been found, but a seek check is indicated.
		5-7 (5-7)		A(&Filename.W) - Address of the random/sequential retrieval work area.
	IJHSIOAR	8-11 (8-B)		Address of IOAREAS, I/O area used for sequential retrieval.
	IJHCRARA	12-15 (C-F)		Address of IOAREAR, I/O area used for random retrieval or address of IOAREA2 (if specified) for sequential retrieval.
	IJHCRKEY	16-19 (10-13)		Address of KEYARG, field containing user-supplied key used for random READ/WRITE operations and sequential retrieval initiated by key.
	IJHCRWOR	20-23 (14-17)		Address of WORKR, work area used for random retrieval.
	IJHSDB1	24-27 (18-1B)		Current sequential I/O area address.
	IJHSLIOR	28-31 (1C-1F)		1. L IOREG,*-4 - Load I/O register for sequential or 2. 4-byte NO-OP instruction for random.
	IJHSLMIT	32 (20)		X'00' = No Verify; X'40' = Verify.
		33 (21)		X'00' = Blocked; X'08' = Unblocked.
		34 (22)		R = First prime data record on shared track.
		35-39 (23-27)		Limits for sequential (CCHHR).

DTFIS ADDRTR (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
	IJHSINIT	40-41 (28-29)		H'0' = Blocked records. H'2' = Overflow record. H'8' = Unblocked records.
		42 (2A)		X'C7' = 2311, 2314, or 2319; X'FF' = 3330, 3340.
		43-47 (2B-2F)		Initial values for sequential.
&Filename.H	IJHSCADR	48-55 (30-37)		Current sequential DASD address (MBBCCHHR).
	IJHSCOVF	56-63 (38-3F)		Current overflow DASD address (MBBCCHHR).
	IJHSRCNT	64-65 (40-41)		Sequential record count.
	IJHSTICU	66-67 (42-43)		Current track index entry for sequential (HR).
&Filename.T		68-69 (44-45)		Number of records tagged for deletion.
	IJHRREGS	70-71 (46-47)		LR &IOREG,0 for random (or 2-byte NO-OP for sequential).
&Filename.G	IJHRIDSV	72-79 (48-4F)		DASD address save area for random retrieval (MBBCCHHR).
	IJHRADSV	80-83 (50-53)		Record pointer within I/O area for write (for random retrieval)
&Filename.R	IJHROVCN	84-87 (54-57)		Nonfirst overflow record count.
The following information is generated if the cylinder index in core option is specified. Bytes 88-91 (58-5B) are not used.				
	IJHCORST	92-95 (5C-5F)		A(&INDAREA) - Starting address of main storage area specified for cylinder index.
		96-97 (60-61)		AL2(&INDSIZE) - Number of bytes in main storage available for cylinder index.
		98-105 (62-69)		Next cylinder index entry to be read (MBBCCHHR).
		106-110 (6A-6E)		Last cylinder index entry (CCHHR).
	IJHCORBT	111 (6F)		Core index byte.
			0	1 = First time through $$BINDEX.
			1	1 = End of cyl. index reached.
			2	1 = Index skip option specified.
			3	1 = Suppr. index in-core option and read cylinder index.
			4-7	Not used.
	IJHCORKY	112-115 (70-73)		Pointer to key (stored by module).

DTFIS ADDRTR (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
The following information is generated if the prime data in-core add function is specified. This information is aligned on a double word boundary. If both cylinder index in-core and prime data in-core add functions are specified, the following information is found in bytes 116-131 (74-83).				
	IJHPSIZE	116-117 (74-75)		Size of IOAREAL.
	IJHPMAX	118-119 (76-77)		Maximum number of prime data records in main storage.
	IJHPDSP1	120-123 (78-7B)		Address of write CCWs.
	IJHPDSP2	124-127 (7C-7F)		Address of read CCWs.
	IJHPSW	128(80)		Switch byte.
			0	1 = EOF (Bits 1-7 not used).
		129 (81)		Reserved.
	IJHDCWRK	130-131 (82-83)		Work field for I/O module.

DTFIS ADDRTR (cont...)

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
&Filename.B		0-7 (0-7)		X'07', &Filename.S+1, X'40', 6 - Long seek CCW with command chaining.
	IJHCCCW	8-63 (8-3F)		Channel program build area.
		64-127 (40-7F)		Channel program build area for add function only.
&Filename.D	IJHACOCR	128-135 (80-87)		Cylinder overflow control record (COCR).
	IJHACTNA	136-143 (88-8F)		Current track index normal entry count field.
	IJHACTOA	144-151 (90-97)		Current track index overflow entry count field.
	IJHACRID	152-159 (98-9F)		Current prime data record count field.
	IJHACFID	160-167 (A0-A7)		Current overflow record count field.
	IJHACTIN	168-177 (A8-B1)		Track index normal entry data field.
	IJHACLNK	178-187 (B2-BB)		Current overflow record sequence-link field.

DTF Assembly Label	Module DSECT Label	Bytes	Bits	Function
	IJHACTIA	188-197 (BC-C5)		Current track index overflow entry data field.
	IJHAGATE	198 (C6)		X'01' - Add to EOF. X'02' - Add to independent overflow area.
		199-201 (C7-C9)		Overflow control bytes (CCH).
	IJHAOCOH	202-203 (CA-CB)		High HR on overflow track.
		204-211 (CC-D3)		Volume upper limit for prime data records (MBBCCHHR).
	IJHAICOM	212-217 (D4-D9)		CLC 0(&KEYLEN,13),0(6) - Unblocked CLC 0(&KEYLEN,13),&KEYLOC-1(6) - Blocked Utility CLC for key.
	IJHAISKY	218-223 (DA-DF)		MVC 0(&KEYLEN,13),0(12) - Unblocked MVC 0(&KEYLEN,13),&KEYLOC-1(12) - Blocked Utility MVC for key.
&Filename.E		224-227 (E0-E3)		First entry in DSKXTN table (logical unit, cell number). (Note 1)
		232-235 (E8-EB)		4X'FF' - End of DSKXTN table. (Note 2)
&Filename.K		236+ (EC-end)		Key area for add only. Number of bytes depends on key length, KEYLEN.

Notes:

1. Each entry in the DSKXTN table is four bytes long. The minimum number of entries is two. There is one entry per extent.
2. Location of the end-of-table indicator depends on length of DSKXTN table.

Numbers in parentheses are displacements in hexadecimal notation.

DTF Assembly Label	Bytes	Bits	Function
&Filename	0–15 (0–F)		Command Control Block (CCB). The CCW address in bytes 9–11 (09–0B) is changed by OPEN to point to a DTF Extension in the user virtual area. CLOSE restores it.
	16 (10)	0	1 = Dequeue old volume extents.
		1	1 = Dummy OPEN to obtain extents from label track.
		2	1 = File assigned 'IGN' (COBOL).
		3	1 = Track hold option specified.
		4	1 = DTF relocated by OPENR.
		5	1 = Input trailer labels to be processed at close time (COBOL only).
		6	1 = Spanned processing.
			1 = COBOL end–of–extent option specified.
	17–19 (11–13)		Address of logic module. OPEN changes this address to point to the logic module in the system virtual area. Close restores it.
	20 (14)		DTF type for OPEN/CLOSE (X'20' = sequential access DASD files).
	21 (15)	0	1 = Not Used.
		1	1 = Blocked file.
		2	1 = Work file.
		3	1 = Work area specified.
		4	1 = Not a Version 1 type table.
		5	1 = Open, 0 = Closed.
		6	1 = Input, 0 = Output.
		7	1 = User labels specified.
	22–28 (16–1C)		Filename (DTF Name).
	29 (1D)		Device Type Code: X'00' = 2311 X'09' = 3340 35MB X'01' = 2314, 2319 X'0A' = 3340 70MB X'04' = 3330–1, –2 X'0B' = 3375 X'05' = 3330–11 X'0C' = 3380 X'07' = 3350 X'90' = FBA device X'08' = 3340 general Note: In previous versions, last byte of filename contains device type code.
	30–35 (1E–23)		Address of Format 1 label in VTOC (BCCHHR or PBN).
	36–37 (24–25)		Volume sequence number.
	38 (26)		Open communications byte. Input File
		0	1 = No more extents.
		1	1 = Update file.
		2	1 = Process trailer labels.
		3	1 = Exit to user's EOF routine.
		4	1 = Next extent on new volume.
		5	1 = Return to close routine.
		6	1 = Process header labels.
		7	1 = Extent switch.

DTFSD (Data Files) (cont...)

DTF Assembly Label	Bytes	Bits	Function
	38 (26) (cont.)		Output File
		0	1 = No more extents.
		1	1 = Extents needed at close time.
		2	1 = Process trailer labels.
		3	1 = Process header labels.
		4	1 = Next extent on new volume.
		5	1 = Extents entered via console.
		6	1 = Process trailer labels at close.
		7	1 = Check extent for minimum of two tracks.
	39 (27)	0	1 = Extent bypassed before file is opened (Input only).
		1	1 = FEOVD has been issued (Input only).
		0-7	Sequence number of current extent opened (Output only).
	40 (28)		Sequence number of last extent opened.
	41-43 (29-2B)		Address of user's label routine.
	44 (2C)	0	Not used.
		1	1 = Device supports RPS.
		2	1 = Version 3 DTF.
		0-7	Not used.
	45-47 (2D-2F)		Address of IOAREA1.
	48-51 (30-33)		Address of user's label track (X'80000000') CCHH for CKD and device address for FBA.
	52-53 (34-35)		Lower head limit (HH); zeros for FBA.
	54-57 (36-39)		Extent upper limit (CCHH) for CKD; device address for FBA.
&Filename.S	58-59 (3A-3B)		Seek address (BB): X'0000' if a disk device; zero for FBA.
	60-63 (3C-3F)		Search argument (CCHH); physical block number for FBA.
	64 (40)		Record number: 0 for FBA; 0 for output, 1 for input for CKD.
	65-67 (41-43)		EOF address if input file. Key length and data length if output file.

	68-71
	(44-47)
			CCHH = X'00C80009' if 2311 - type 1
			CCHH = X'00C80013' if 2314 or 2319 - type 1
			CCHH = X'01940012' if 3330 - type 1
			CCHH = X'03280012' if 3330-11 - type 1
			CCHH = X'015C000B' if 3340 35MB
			CCHH = X'02B8000B' if 3340 70MB
			CCHH = X'022B001D' if 3350 - type 1
			CCHH = X'03B2000B' if 3375
			CCHH = X'0375000E' if 3380
			PBN = maximum block number if FBA
			Note: The last two digits of CCHH are replaced by the current head number when the type 128 version of the device is specified.

DTFSD (Data Files) (cont...)

DTF Assembly Label	Bytes	Bits	Function
	72 (48)		For fixed length records only: Number of records per track (input), number of records per track minus one (output).
	73 (49)		Switch byte used by the logic modules for various switching purposes. Functions indicated are for the ON condition (1) of the respective bit.
			Fixed Length Record Modules
		0	Not first entry after Open (INPUT and UPDATE). Not first write after Open (OUTPUT)
		1	Short record (INPUT and UPDATE without truncation).
		2	Partial block written (OUTPUT).
		3	ERROPT=SKIP (INPUT), TRUNCS=YES (OUTPUT).
		4	End-of-file record written (OUTPUT). End of extent (UPDATE).
		5	Truncation not specified (used by OPEN routines).
		6	Write block of records (UPDATE).
		7	End of file (UPDATE).
			Variable Length Record Modules
		0	Not first entry after OPEN (INPUT and UPDATE). Write record (OUTPUT).
		1	Wrong length record (INPUT), TRUNCS=YES (OUTPUT). Second GET operation performed (UPDATE).
		2	Return to close routine (OUTPUT). Update specified (UPDATE).
		3	Not first entry after OPEN (OUTPUT).
		4	New extent required by CLOSE,
		5	Capacity of I/O area exceeded (OUTPUT). Second GET required (UPDATE).
		6	Not first read (INPUT). Second GET issued (UPDATE).
		7	Unnecessary to read (INPUT). Track capacity exceeded (OUTPUT).Save record count (UPDATE)
			Undefined Length Record Modules
		0	Not first entry after OPEN (All modules).
		1	Save record count (UPDATE).
		2	Return to close routine (OUTPUT).
		3	Second GET issued (UPDATE).
		4	Not used.
		5	PUT command issued (UPDATE).
		6	End of file reached (UPDATE).
&Filename.B	74-75 (4A-4B)		Block size minus one.
	76-80 (4C-50)		Extent lower limit and record number. Initialize with the current PBN/track address: CCHHR for CKD, track address for FBA.
	81 (51)	1	1 = FEOVD has been issued (output only).
	81-83 (51-53)		Address of user wrong-length record routine if input file. Track capacity counter if output file.

DTFSD (Data Files) (cont...)

DTF Assembly Label	Bytes	Bits	Function
	84-87 (54-57)		Instruction to load user's register IOREG. Note: This field is a NOP unless blocked records are processed in one I/O areas, or two I/O areas are specified and records are processed in the I/O areas.
only if BLKSIZE=MAX specified			
	88-91 (58-5B)		Address of current available Input/Output area.
	92-95 (5C-5F)		Logical record size.
	96-99 (60-63)		Address of end of input/output area.
	100 (64)		Logical indicators
		0	1 = ERROPT = address.
		1	1 = ERROPT = IGNORE.
		2	1 = ERROPT = SKIP.
		3	1 = VERIFY = YES.
		4	1 = Two I/O areas.
		5	1 = WLRERR = address (Fixed length and variable records.) 1 = Output file (Undefined length records).
		6	1 = Fixed length records. 0 = Variable or undefined length records.
		7	Control parameter specified.
	101-103 (65-67)		Address of user's read error routine.
The following section is added to the DTFSD table for fixed-length record output files.			
	If CONTROL is not specified:		
	160-163 (A0-A3)		End-of-extent routine address (primarily used by COBOL compiler).
	If CONTROL = YES:		
	184-187 (B8-BB)		End-of-extent routine address (primarily used by COBOL compiler).
The following section is added to the DTFSD table for variable-length record, and spanned record input files.			
If UPDATE is not specified:			
	If CONTROL = YES: *		
	176-179 (B0-B3)		Logical record length.
	180-183 (B4-B7)		RX type instruction.

DTFSD (Data Files) (cont...)

DTF Assembly Label	Bytes	Bits	Function
	184 (B8)	0	Not used.
		1	1 = Skip segment.
		2	1 = Spanned first time.
		3	Not used.
		4	Not used.
		5	Not used.
		6	Not used.
		7	Not used.
	185-187 (B9-BB)		Pointer in logical record.
*These bytes are always generated when spanned processing is specified. The following section is added to the DTFSD table for variable-length spanned record update files.			
	216-219 (D8-DB)		Logical record length.
	220-223 (DC-DF)		RX type instruction. Load record
	224 (E0)	0	Not used.
		1	1 = Skip segment.
		2	1 = Spanned first time.
		3	1 = Null segment.
		4	1 = Spanned PUT return.
		5-6	Not used.
		7	1 = No update.
	225-227 (E1-E3)		Pointer in logical record.
	228-235 (E4-EB)		Not used.
	236-239 (EC-EF)		Extent status save area.
The following section is added to the DTFSD table for variable-length spanned record output files.			
	160-163 (A0-A3)		Space remaining in output area.
	164-165 (A4-A5)		Track capacity.
	166-169 (A6-A9)		Instruction to load user's register VARBLD. (If VARBLD is not specified, instruction is NO-OP.).
	If CONTROL = YES: *		
	170-172 (AA-AC)		Not used.
	173-175 (AD-AF)		End-of-extent routine address (primarily used by COBOL compiler).

DTFSD (Data Files) (cont...)

DTF Assembly Label	Bytes	Bits	Function
The following section is added to the DTFSD table for variable-length spanned record output files.			
	200–203 (C8–CB)		Logical record length.
	204–207 (CC–CF		RX type instruction. Store a record size.
	208 (D0)	0	Not used.
		1	Not used.
		2	1 = Leading segment.
		3	1 = Output block truncated.
		4	1 = End of track.
		5	1 = Track truncated.
		6	1 = Save count.
		7	1 = Volumes spanned.
	209–211 (D1–D3)		Pointer in logical record.
	212–219 (D4–DB)		Not used.
	220–223 (DC–DF)		Extent status save area.
The following section is added to the DTFSD table for undefined length record output files.			
	160–161 (A0–A1)		Track capacity.
	If CONTROL = YES:		
	162–164 (A2–A4)		Not used.
	164–167 (A4–A7)		End-of-extent routine address (primarily used by COBOL compiler).

Numbers in parantheses are displacements in hexadecimal notation.

DTF Assembly Label	Bytes	Bits	Function
&Filename	0–15 (0–F)		Command Control Block (CCB). Note: The CCW address in bytes 9–11 (09–0B) is changed by OPEN to point to DTF Extension in the user virtual area. CLOSE restores it.
	16 (10)	0–1	Not used.
		2	1 = File assigned 'IGN' (COBOL).
		3	1 = Track hold option specified.
		4	1 = DTF relocated by OPENR.
		5–7	Not used.
	17–19 (11–13)		Address of logic module. (OPEN changes this address to point to the logic module in the SVA. CLOSE restores it.)
	20 (14)		DTF type for OPEN/CLOSE (X'20' = sequential access disk files).
	21 (15)	0	0 = Disk device.
		1	1 = CLOSE macro is not to delete format-1 and format-3 file labels.
		2	1 = Work file.
		3	Type of open: 1 = Point, 0 = Normal.
		4	1 = Routine entered from close routine.
		5	1 = File opened. 0 = File closed.
		6	Not used.
		7	1 = Reentry to close routine.
	22–28 (16–1C)		Filename (DTF Name).
	29 (1D)		Device Type Code: X'00' = 2311 X'01' = 2314, 2319 X'04' = 3330–1, –2 X'05' = 3330–11 X'07' = 3350 X'08' = 3340 general X'09' = 3340 35MB X'0A' = 3340 70MB X'0B' = 3375 X'0C' = 3380 X'90' = FBA device Note: In previous versions, last byte of filename contains device type code.
	30–31 (1E–1F)		Track capacity counter.
	32–35 (20–23)		Address of format-1 label in VTOC. CCHHR for CKD; PBN address for FBA.
	36 (24)		Extent sequence number.
	37 (25)		Open communications byte.
		0	Not used.
		1	1 = Device supports RPS.
		2	1 = Version 3 DTF.
		3	1 = Symbolic unit in DTF.
		4	1 = Next extent on new volume.
		5	1 = Extent opened.
		6–7	Not used.

DTFSD (Work Files) (cont...)

DTF Assembly Label	Bytes	Bits	Function
	38 (26)		Lower head limit for CKD; zero for FBA.
	39 (27)		Upper head limit for CKD; zero for FBA.
&Filename.L	40–41 (28–29)		Record length.
	42–45 (2A–2D)		Initial extent lower limit for CKD; Initial extent lower limit PBN for FBA.
	46–49 (2E–31)		Current extent lower limit for CKD; Current extent lower limit PBN for FBA.
	50–53 (32–35)		Extent upper limit for CKD; Current extent upper limit PBN for FBA.
&Filename.S	54–55 (36–37)		Seek address (BB=X'0000') for CKD; not used for FBA.
	56–59 (38–3B)		Search address (CCHH) for CKD; PBN for FBA.
	60 (3C)		Record number for CKD; zero for FBA.
	61 (3D)		Switch byte used by logic module.
		0	1 = First write entry indicator.
		1	1 = Write update indicator.
		2	1 = POINTS macro issued.
		3	Not first record on track. (RECFORM=UNDEF).
		4	1 = Track upper limit reached.
		5	Not used.
		6	1 = Check after read/write.
		7	Not used.
&Filename (Note 1)	62–63 (3E–3F)		Maximum record length.
	64 (40)		Not used.
	65–67 (41–43)		Address of user's EOF routine.
	68 (44)		Logical indicators.
		0	1 = ERROPT = address.
		1	1 = ERROPT = IGNORE.
		2	1 = Fixed-length unblocked records.
		3	1 = Verify specified.
		4	1 = ERROPT = SKIP.
		5	1 = Reread after read error.
		6–7	Not used.
	69–71 (45–47)		Address of user read/write error routine.

Note: Only if BLKSIZE=MAX specified

Numbers in parantheses are displacements in hexadecimal notation.

DTFMT Data Area (Common Part for all Record Formats)

Offset	Field Name	Bytes and Bit Pattern	Description
0 (0)	IJJTDTF	5C	
0 (0)	DTFCCB	10	Command Control Block CCB use structure IORBADR to locate data in the CCB Note that the address of the channel program will be replaced by the address of the DTF Extension when the file is opened.
8 (8)	DTFOFFS	1	Buffer offset length ASCII
16 (10)	DTFFL1	1	Flag Byte 1
	DTFFIRST	X'80'	first time entered IOMOD
	DTFOPEN	X'40'	1=file is opened
	DTFCBIGN	X'20'	COBOL open ignore option
	DTFCBUSA	X'10'	ANSI COBOL only if std or nonstd files
	DTFRELO	X'08'	DTF table address constants relocated by $$BOPENR
	DTFSPN	X'04'	spanned record format
	DTFASCII	X'02'	1=ASCII 0=EBCDIC

	DTFLNGCK	X'01'	if ASCII input: length check
			if ASCII output: buffer offset length=4

17 (11)	DTFIOMOD	3	address of logic module
20 (14)	DTFTYPE	1	DTF file type X'10'=workfile X'11'=nonstd or unlabeled X'12'=std labeled output X'13'=std labeled input bwd X'14'=std labeled input fwd X'15'=std labeled work files
21 (15)	DTFFLAG2	1	Flag Byte 2
	DTFFRST2	X'80'	0=first time entry 1=not first time entry
	DTFBLK	X'40'	1=blocked
	DTF2IO	X'20'	0=1 I/O area 1=2 I/O areas
	DTFWA	X'10'	1=Workarea
	DTFINP	X'08'	1=Input
	DTFBWD	X'04'	1=Backwards also used for POINTW
	DTFCHKPT	X'02'	1=Checkpoint records
	DTFTRUNC	X'01'	1=TRUNC required during close
22 (16)	DTFFILNM	7	Symbolic file name
29 (1D)	*	1	Not used
30 (1E)	DTFCCWCD	1	Command Code for CCW
31 (1F)	DTFFLAG3	1	Flag byte 3
	DTFVDISP	xxxxx..	displacement to locate the variable fields in DTF
	DTFV21	X'04'	1=DTF Version 2.1 onwards
	DTFCBREW	X'02'	used by COBOL files for rewind set to '1' by COBOL
	DTFDHDR	X'01'	1=Header label and EOV info wanted

DTFMT Data Area (Common Part for all Record Formats) (cont...)

		Bytes and	
Offset	Field Name	Bit Pattern	Description
32	DTFFLAG4	1	Flag Byte 4
(20)	DTFSTDLB	X'80'	1=STD Labels
	DTFNSTD	X'40'	0=unlabeled tape file
			1=nonstandard labeled file
	DTFREWUN	X'20'	1=rewind unload modified by COBOL
	DTFNOREW	X'10'	modified by COBOL on close
			0=REWIND 1=NO REWIND
	DTFBWD2	X'08'	1=BACKWARDS
	DTFULAB	X'04'	1=User label routine specified
	DTFNOTM	X'02'	1=TM option not specified
	DTFSOEOF	X'01'	EOF-EOV switch (IBM Sort)
33 (21)	DTFULBRT	3	address of user label routine
36	DTFFLAG5	1	Flag Byte 5
(24)	DTFPH	X'80'	1=DTF build by DTFPH
	DTFCBLBL	X'40'	COBOL label indicator
			1=user label routine on input file
	DTFINP2	X'20'	1=input
	DTFEOVF	X'10'	1=force EOV switch
	DTFEOFV	X'08'	EOF-EOV switch (output)
			1=EOF
			0=EOV
	DTFOPENO	X'04'	1=file is opened (for VSAM)
	DTFVSPN	X'02'	1=variable or spanned records
	DTFUTIL	X'02'	this bit is modified by any unknown utility
	DTFUDEF	X'01'	1=undefined records
37 (25)	DTFUEOF	3	address of user EOF routine
20	DTFBLKCT	4	block count
(28)			for workfiles initialized to
			00000000 for read forward
			00400000 for read backward
44	DTFVARX	0	start of variable part of DTF
(2C)			use structures DTFFIX, DTFVAR, DTFSPAN, and DTFUNDEF
44 (2C)	*	12	Not used
56 (38)	DTFCCW0	8	CCW
56 (38)	DTFCCWC	1	CCW command code
57 (39)	DTFIOA1	3	address of IOAREA1
64	DTFIOA2	4	address of IOAREA1 or address
(40)			of IOAREA2 if any
68 (44)	*	20	Not used
88	DTFXPH	4	address of DTF extension for
(58)			DTFPH files

Numbers in parantheses are displacements in hexadecimal notation.

DTFMT Data Area (Variable Part for Record Format=FIXED)

Offset	Field Name	Bytes and Bit Pattern	Description
44 (2C)	DTFFIX	40	Length variable part
44 (2C)	*	8	Not used
52 (34)	DTFFIOR	4	instruction to load/store addr. of record from/into users 'IOREG' reg.
53 (35)	DTFFIOR2	1	IOREG regs in the leftmost 4 bits
57 (39)	*	16	Not used
72 (48)	DTFFRECS	4	F'RECSIZE' for forward F'-RECSIZE' for backward
77 (4D)	*	2	Not used
80 (50)	DTFFBLKS	2	block size
82 (52)	*	2	Not used
84 (54)	DTFFREC1	2	record size-1
86 (56)	*	2	Not used
88 (58)	DTFFWLR	4	WLRERR specification for all input files
92 (5C)	DTFFERR2	4	ERROPT specific. for all inp. files and not STD labeled output files
96 (60)	*	2	Not used
104 (68)	DTFFERR3	4	ERROPT specification for STD labeled output files

DTFMT Data Area (Variable Part for Record Format=VARIABLE)

Offset	Field Name	Bytes and Bit Pattern	Description
44 (2C)	DTFVAR	4C	Length of variable part
44 (2C0	DTFVRBLD	4	Instruction to load the length of the remaining space in the output area into the users VARBLD register
45 (2D)	DTFVRBL2	1	VARBLD regs in the leftmost 4 bits
48 (30)	*	4	Not used
52 (34)	DTFVIOR	4	instruction to store the address of record into the users 'IOREG" regs.
53 (35)	DTFVIOR2	1	IOREG regs in the leftmost 4 bits
56 (38)	*	12	Not used
68 (44)	DTFVBLKS	4	block size
72 (48)	*	12	Not used
84 (54)	DTFVRBEG	4	address of next record
88 (58)	*	4	Not used

DTFMT Data Area (Variable Part for Record Format=VARIABLE) (cont...)

Offset	Field Name	Bytes and Bit Pattern	Description
92 (5C)	DTFVERR1	4	ERROPT specification for not STD labeled output files (Version 3 downward)
96 (60)	*	4	Not used
100 (64)	DTFVWLR	4	WLRERR specification for all input files
104 (68)	DTFVERR3	4	ERROPT specification for not STD labeled output files (Version 3 onwards) and for all input files
108 (6C)	*	8	Not used
116 (74)	DTFVERR4	4	ERROPT specification for STD labeled output files

Numbers in parantheses are displacements in hexadecimal notation.

DTFMT Data Area (Variable Part for Record Format=SPANNED)

Offset	Field Name	Bytes and Bit Pattern	Description
44 (2C)	DTFSPAN	4C	Length of variable part
44 (2C)	*	8	Not used
52 (34)	DTFSRECS	4	Instruction to load/store the recordsize from/into users 'RECSIZE' register (output/input)
	DTFSRECO	1	not used op-code of instr
53 (35)	DTFSREC2	1	RECSIZE register in the leftmost 4 bits
54 (36)	*	2	Not used
56 (38)	*	12	Not used
68 (44)	DTFSBLKS	4	block size
72 (48)	*	12	Not used
84 (54)	DTFSRBEG	4	address of next record
88 (58)	*	4	Not used
92 (5C)	DTFSERR1	4	ERROPT specification for not STD labeled output files (Version 3 downward)
96 (60)	*	4	Not used
100 (64)	DTFSWLR	4	WLRERR specification for all input files
104 (68)	DTFSERR3	4	ERROPT specification for not STD labeled output files(Version 3 onwards) and for all input files
108 (6C)	*	8	Not used
116 (74)	DTFSERR4	4	ERROPT specification for STD labeled output files

DTFMT Data Area (Variable Part for Record Format=UNDEFINED)

Offset	Field Name	Bytes and Bit Pattern	Description
44 (2C)	DTFUNDEF	3C	Length of variable part
44 (2C)	*	4	Not used
48 (30)	*	4	Not used
49 (31)	DTFURECI	1	RECSIZE register in the
50 (32)	*	2 not used	instruction to store address of
52 (34)	DTFUIOR	4	record into the users 'IOREG' register (input)
53 (35)	DTFUIOR2	1	IOREG reg. in the leftmost 4 bits
54 (36)	*	2	Not used
56 (38)	*	20	Not used
76 (4C)	DTFUBLKS	2	block size
78 (4E)	*	3	Not used
81 (51)	DTFURECO	1	RECSIZE register in the rightmost 4 bits. (for output)
82 (52)	*	2	Not used
84 (54)	DTFUERR1	4	ERROPT specification for not STD labeled output and all input files
88 (58)	DTFUWLR	4	WLRERR specific. for all inp. files
92 (5C)	*	8	Not used
100 (64)	DTFUERR3	4	ERROPT specification for STD labeled output files

Numbers in parantheses are displacements in hexadecimal notation.

DTFMT Data Area (Variable Part for Workfiles)

Offset	Field Name	Bytes and Bit Pattern	Description
16 (10)	DTFWRKF	24	Length of variable part
	DTFWFLG1	1	Flag byte 1
17 (11)	*	4	Not used
21	DTFWFLG2	1	Flag byte 2
(15)	DTFWNREW	X'80'	1=no rewind
	DTFWRUN	X'40'	1=rewind unload
	DTFWWRK	X'20'	1=work file
	DTFWBWD	X'10'	1=backward
	DTFWWRT	X'08'	1=write (output file)
22 (11)	*	2	Not used
24 (18)	DTFWRECL	2	record length
26 (1A)	DTFWBLKS	2	maximum blocksize
28 (1C)	*	1	Not used
29 (1D)	DTFWEOF	3	address of users EOF routine
32	DTFWCCW	8	CCW
(20)	DTFWCCWC	1	CCW command code

Numbers in parantheses are displacements in hexadecimal notation.

DTFMT Data Area (Variable Part for Workfiles) (cont...)

Offset	Field Name	Bytes and Bit Pattern	Description
33 (21)	DTFWCCWA	3	CCW data address
40 (28)	*	4	Not used
44	DTFWERR1	4	address of 'ERROPT' routine
(2C)	DTFWERRA	X'80'	ERROPT routine specified
	DTFWERRI	X'40'	ERROPT = IGNORE specified
	DTFWHDR	X'20'	1=header label and EOV info wanted (STD labeled)
	DTFWRECF	X'10'	record fixed unblocked
48 (30)	DTFWRL1	2	record length (STD labeled)
50 (32)	DTFWBKS1	2	maximum blksize (STD labeled)

Numbers in parantheses are displacements in hexadecimal notation.

Offset (Hex)	Field Name	Bytes and Bit Pattern	Description
0	IJJTDTFX	8A0	
	DTFXID	8	Control block ID 'DTFX'
8	DTFCCW	28	room for 5 CCW'S
	DTFCWCMD	1	CCW command code
9	DTFCWADD	3	data address
C	DTFCWFL1	1	flag byte
C	DTFCWCD	X'80'	1=data chaining
	DTFCWCC	X'40'	1=command chaining
	DTFCWSLI	X'20'	1=suppress incorrect length
	DTFCWSK	X'10'	1=suppress data transfer
E	DTFCWCNT	2	CCW count
30	DTFMAIN	4	address of Mother DTF
34	DTFXLEN	2	length of DTF Extension Block
36	DTFXOP	1	actual/last operation against DTF
37	DTFTID	1	ID of active task
38	DTFNXDTF	4	DTF list pointer
3C	DTFODL	4	pointer to ODL entry
40	DTFOAT	4	pointer to OAT entry
44	DTFXOPS	1	temp save area for DTFXOP
45	DTFLIO1	1	save area for LIOCSCOM byte 1
46	*	2	Not used
File Type Specifications			
48	DTFFLGS	A	indicators
48	DTFFLG1	1	indicators and switches
	DTFXOPEN	X'80'	1=the file is open
	DTFXSTD	X'40'	1=standard labels
	DTFXNSTD	X'20'	0=unlabeled tape 1=nonstandard labeled tape
	DTFXBWD	X'10'	1=backward
	DTFXINP	X'08'	1=input file
	DTFXWRK	X'04'	1=work file
	DTFXASC	X'02'	1=ASCII file

Numbers in parantheses are displacements in hexadecimal notation.

DTFMT (DTF Extension) (cont...)

Offset (Hex)	Field Name	Bytes and Bit Pattern	Description
Record Format Specifications			
49	DTFFLG2	1	indicators and switches
49	DTFXBLK	X'80'	1=blocked record format
	DTFXFIX	X'40'	1=fixed record format
	DTFXVAR	X'20'	1=variable record format
	DTFXSPN	X'10'	1=spanned record format
	DTFXUND	X'08'	1=undefined record format
	DTFX2IO	X'04'	1=2 IOAREAS specified
	DTFXWA	X'02'	1=WORKAREA=YES specified
	DTFXCHK	X'01'	1=checkpoint records on file
Miscellaneous Specifications			
4A	DTFFLG3	1	indicators and switches
4A	DTFFRST	X'80'	1=first time indicator initialized by 'OPEN' to 0
	DTFXCPR	X'20'	1=user CCB is translated
	DTFXNOTM	X'10'	1=TPMARK option not specified
	DTFXULR	X'08'	1=user label routine specified
	DTFCHKP	X'04'	1=checkpoint in progress
	DTFEXCP	X'02'	1=EXCP initiated
4B	DTFFLG4	1	indicators and switches
4B	DTFANSI3	X'80'	1=ANSI level 3 file
	DTFXUSCO	X'40'	1=ANSI COBOL file
	DTFBFULL	X'20'	1=block full write it on tape
	DTFRFULL	X'10'	1=record full give it the user
	DTFULBL	X'08'	1=all user labels written
Flags used during 'OPEN'			
4C	DTFFLG5	1	open flags 1
4C	DTF1VOL1	X'80'	VOL1 record found
	DTFEHDR1	X'40'	empty HDR1 record found
	DTFFHDR1	X'20'	HDR1 record on tape
	DTF1TM	X'10'	first record on tape is a TM
	DTFEOD	X'08'	'/*' read on SYSIN/SYSRDR
	DTFUNEX	X'04'	overwrite unexpired file
	DTFNEWTP	X'02'	operator has mounted new tape
	DTFIGNLB	X'01'	ignore further label processing
4D	DTFFLG5A	1	open flags 2
4D	DTFINILP	X'80'	1=tape was at load point
	DTFLTLBL	X'40'	1=// TLBL record found
	DTFWVOL1	X'20'	1=VOL1 label written on tape
	DTFLPTM	X'10'	1=tape mark at load point
	DTFFHDR2	X'08'	1=HDR2 record on tape
	DTFUNL	X'04'	1=tape is unloaded
FLAGS USED DURING 'CLOSE' AND 'EOF/EOV'			
4E	DTFFLG6	2	'CLOSE' and 'EOF' flags
4E	DTFXEOF	X'80'	EOF
	DTFXEOV	X'40'	EOV
	DTFXEOFQ	X'20'	quick EOF
	DTFXEOVQ	X'10'	quick EOV
	DTFXEOVF	X'08'	force EOV
	DTFXEOVS	X'04'	EOV reached during writing EOF
	DTFWLR	X'02'	wrong length record
	DTFTRU	X'01'	actual block is truncated

DTFMT (DTF Extension) (cont...)

Offset (Hex)	Field Name	Bytes and Bit Pattern	Description
Other Flags used during OPEN/CLOSE/EOF/IOCS			
50	DTFFLG7	1	
50	DTFMFIL	X'80'	multifile volume
	DTFMVOL	X'40'	multivolume file
	DTFCHDIR	X'20'	FSL/BSL changed read direction
Flags for DEBUG			
51	DTFFLG8	1	flag byte
51	DTFDBG1	X'80'	reserved
	DTFDBG2	X'40'	reserved
	DTFDBG3	X'20'	reserved
	DTFDBG4	X'10'	reserved
	DTFDBG5	X'08'	reserved
	DTFDBG6	X'04'	reserved
	DTFDBG7	X'02'	reserved
	DTFDBG8	X'01'	reserved
52	*	2	Not used
Interface to User Routines			
54	DTFUSRIF	8	
54	DTFUCODE	4	user code in register 0
56	DTFUCOD1	2	2-bytes code for EOF/EOV
57	DTFUCOD2	1	1-byte code for OPEN/CLOSE
58	DTFUPTR	4	pointer to data
5C	DTFUIF	8	interface to user routine
5C	DTFUADTF	4	address of users DTF
60	DTFUABLK	4	address of block in error
64	DTFUR00S	4	save area for register 0
68	DTFUR01S	4	save area for register 1
6C	DTFUR13S	4	save area for register 13
70	DTFUR15S	4	save area for register 15
CCB'S CCW'S Sense Bytes			
74	DTFUIOMD	4	users I/O module address
78	DTFUXPH	4	save area for DTFPH
7C	*	4	reserved
80	DTFOCCB	10	origin CCB copied from DTF
90	DTFXCCB	10	CCB copied from DTF used for OPEN/CLOSE/EOV
A0	DTFPCCB	10	CCB to be used for SYSLST
B0	DTFSNS	18	room for 24 sense bytes
C8	DTFPUBE	8	saved PUB entry for actual tape unit - use structure PUBADR to locate data
D0	DTFPUB	4	address of PUB entry
D4	DTFPUB2	4	address of PUB2 entry
D8	DTFNTAP	4	number of assigned tapes
DC	DTFNTAPA	4	number of actual tape
E0	*	4	reserved
Addresses and Registers (User)			
E4	DTFIOA1X	4	address of I/O area 1
E4	DTFIOA1A	X'80'	1= I/O area 1 is active
E5	DTFIOAR1	3	pointer to IOAREA1
E8	DTFIOA2X	4	address of I/O area 2
E8	DTFIOA2A	X'80'	1= I/O area 2 is active
E9	DTFIOAR2	3	pointer to IOAREA2

DTFMT (DTF Extension) (cont...)

Offset (Hex)	Field Name	Bytes and Bit Pattern	Description
EC	DTFIOREG	1	specified register for IOREG=(X)
ED	DTFRECSR	1	specified register for RECSIZE=(X)
EE	DTFRVRBL	1	specified register for VARBLD=(X)
F0	DTFAEOF	4	address of users EOF routine address in DTF
F4	DTFAURTN	4	address of error routine to be invoked
F8	DTFAERR	4	ptr to ERROPT specification
FC	DTFAWLR	4	ptr to WLRERR specification
100	DTFERRFL	1	flag byte for error routines
100	DTFWLRAC	X'80'	WLRERR routine active
	DTFERRAC	X'40'	ERROPT routine active
101	*	3	reserved
104	DTFALBRT	4	address of users label routine address in DTF
108	DTFCHKAC	4	ptr to access check routine
Physical Block Description			
11C	DTFPHYS	20	
11C	DTFBBEG	4	pointer to active physical block
120	DTFBEND	4	ptr to last byte+1 in actual physical block
124	DTFBSIZ	4	blocksize as specified in DTF
128	DTFBLEN	4	length of actual physical block
12C	DTFBSPC	4	number of unused bytes in actual IOAREA (remaining space)
130	DTFBCNT	4	number of physical blocks read/written
134	DTFBOFF	4	length of block header initialized by 'OPEN' to: —> 4 if EBCDIC file —> 4 if ASCII file + LENCHK=YES —> 0 IF ASCII file + LENCHK=NO
138	DTFBREC	4	number of records in block
Logical Record Description			
13C	DTFLOG	18	
13C	DTFRBEGW	4	address of record in I/O area or workarea
140	DTFRBEG	4	for input: pointer to actual logical record for output: address of area where the next record can be build
144	DTFRLEN	2	length of actual logical record
146	DTFRLENB	2	length of actual logical record in physical block
148	DTFRLENW	2	number of bytes moved from the users workarea
14A	DTFRTYP	2	type of segment (spanned record)
14C	DTFRACT	2	actual record number
14E	*	2	reserved
150	DTFRSIZ	4	RECSIZE as specified in DTF
Miscellaneous Variables			
154	DTFMISC	10	
158	DTFRCNT	4	residual count
15C	DTFNREC	4	number of logical records in physical block
160	DTFNACT	4	number of actual record

DTFMT (DTF Extension) (cont...)

Offset (Hex)	Field Name	Bytes and Bit Pattern	Description
// TLBL Card Info			
164	DTFTLBL	50	
164	*	1	Not used
165	DTFTFNM	7	file name
16C	*	1	Not used
16D	DTFTFID	11	file ID
17E	DTFTVSER	6	file (volume) serial number modified in: IJJTOPN.OPSTDO IJJTOPN.PVOLSER IJJTOPN.HFLESER
184	DTFTVSQF	4	volume sequence number modified in: IJJTOPN.PVOLSER
	DTFTVSEQ	4	volume sequence number modified in: IJJTOPN.HVOLSEQ
188	DTFTFSQF	4	file sequence number
	DTFTFSEQ	4	file sequence number modified in: IJJTOPN.OPSTDO IJJTOPN.HFLESEQ
18C	DTFTGEN	4	generation number
190	DTFTVERS	2	version number
192	DTFTCDAT	6	creation date
198	DTFTEDAT	6	expiration date
19E	DTFTFSEC	1	file security
19F	DTFTBCNT	6	block count
1A5	DTFTSYS	D	system code
1B3	DTFTFLG	1	flag bits
1B3	DTFTNEW	X'80'	DISP=NEW
	DTFTOLD	X'40'	DISP=OLD
	DTFTMOD	X'20'	DISP=MOD
VOL Label Record Save Area			
1B4	DTFVOL1	50	VOL1 record save area
1B4	DTFVOLN	4	label identifier
1B4	DTFVOLID	3	label identifier 'VOL'
1B7	DTFVOLNR	1	volume label number
1B8	DTFVSER	6	volume (file) serial number
1BE	DTFVSEC	1	volume security
1BF	*	D	reserved
1CC	DTFVSYS	D	system that created the vol. labels
1D9	DTFVOWN	E	volume owner
1E7	*	1C	reserved
203	DTFVANSL	1	ANSI label level
HDR Label Record Save Area			
204	DTFHDR1	50	header label record
204	DTFH1N	4	label identifier
204	DTFH1ID	3	label identifier 'HDR'
207	DTFH1NR	1	file label number
208	DTFHFID	11	file ID
219	DTFHFSER	6	file (volume) serial number
21F	DTFHVSQF	4	volume sequence number
21F	DTFHVSEQ	4	volume sequence number
223	DTFHFSQF	4	file sequence number
223	DTFHFSEQ	4	file sequence number
227	DTFHGENF	4	generation number
227	DTFHGEN	4	generation number
22B	DTFHVERF	2	version number
22B	DTFHVERS	2	version number
22D	DTFHCDAT	6	creation date (YYYDDD)
233	DTFHEDAT	6	expiration date (YYYDDD)
229	DTFHSEC	1	file security code
23A	DTFHBCNT	6	block count
240	DTFHSYS	D	system code

DTFMT (DTF Extension) (cont...)

Offset (Hex)	Field Name	Bytes and Bit Pattern	Description
HDR2 Label Record Save Area			
254	DTFHDR2	50	header label record
254	DTFH2N	4	label identifier
254	DTFH2ID	3	label identifier 'HDR'
257	DTFH2NR	1	file label number
258	DTFHRECF	1	record format 'F'= fixed length 'D'= variable length 'S'= spanned
259	DTFHBLKS	5	blocksize
25E	DTFHRECS	5	record size
263	*	23	reserved
286	DTFHBOFF	2	buffer offset length
Message Writer ($IJJGMSG) Interface			
2A4	DTFMSGS	4	address of message writer save area
2A8	DTFMSGRC	4	return code from message writer
2AC	DTFMSGSA	48	register save area
2F4	DTFMNPL	30	monitor parameter list
324	DTFMPL	20	message parameter list
344	DTFFILL	28	message filler
// TLBL Processing PLIST			
36C	DTFLPL	15	label PLIST
ITRA Buffer Anchor			
384	DTFITRA	34	ITRA buffer constants
384	DTFITRF1	X'80'	ITRA buffer allocated
	DTFITRF2	X'40'	ITRA buffer entries to be done
385	DTFITRB	3	start address of ITRA buffer
388	DTFITRL	4	address of last possible entry in ITRA buffer
38C	DTFITRLG	4	length of ITRA buffer
390	DTFITR1	4	address of first possible entry in ITRA buffer
394	DTFITRN	4	address of actual entry
398	DTFITRX	20	at least one ITRA entry if
398	DTFITRBL	4	pointer to begin of trace for actual request
Save Areas			
3B8	DTFUSAVE	58	users PP save area -it-does-not-have-the-same layout than that in the users PP save area
3B8	DTFUPGNM	8	users program name
3C0	*	4	Not used
3C4	DTFUPSW	4	address part of PSW
3C8	DTFURGS	40	users register from 0-15
	DTFUR08	24	users register from 0-8
3EC	DTFUR9F	1C	users register from 9-15
408	DTFMSGNR	4	last given message number
40C	DTF4191	4	reason code when MSG=4191I
410	DTF13SAV	28	register 13 save area
438	DTFESAVE	2D0	register save area for 9 nested external calls
438	*	14	Not used
44C	DTFEREG0	4	register 0
708	DTFELVL	4	level of external call
708	*	3	Not used

DTFMT (DTF Extension) (cont...)

Offset (Hex)	Field Name	Bytes and Bit Pattern	Description
70B	DTFELVLC	1	level of call
70C	DTF14SAV	38	link register save areas indexed by DTFCLVL
744	DTFCLVL	4	level of subroutine
744	*	3	Not used
747	DTFCLVLC	1	level of subroutine
748	DTFSRVF	4	function indicator for service module IJJTSRV
74C	DTFSVC50	4	reason code for SVC 50
754	*	4	reserved
Work Areas			
758	DTFWORK	18	doubleword bdy for 'CVD' instr
770	DTFWORK1	A0	WORKAREA 1
770	DTFIOBUF	50	
7C0	DTFIOBU2	50	
810	DTFWORK2	A0	WORKAREA 2
ATTENTION: DTFEURTN must be the last variable in DTF Extension **			
8B0	DTFEURTN	0	Starting at this point the DTF extension contains the executable code to setup the linkage to the applications ERROPT/WLRERR routine and to transfer control to it. These routines will return here by using the ERET macro. This executable code will be moved into this field by the transient phase $$BOTSVA during OPEN processing
3C8	DTFUREG	40	declaration for
3C9	DTFUREGS	3	save area
3C9	*	1	reserved

DTFMT Open Anchor Table (OAT)

Offset (Hex)	Field Name	Bytes and Bit Pattern	Description
0	OAT	6C	
0	OATNAME	4	Control block name 'OAT'
4	OATATOP	4	address of IJJTTOP
8	OATAOPN	4	address of IJJTOPN
C	OATACLS	4	address of IJJTCLS
10	OATAEOF	4	address of IJJTEOF
14	OATALOG	4	address of IJJTLOG
18	OATASRV	4	address of IJJTSRV
1C	OATAMSG	4	address of IJJTMSG
20	OATAMSGT	4	address of $IJJGMSG
24	OATAXIT	4	address of $IJJTXIT
28	*	4	reserved
2C	*	4	reserved
30	OATODL	4	address of first ODL
34	OATCOMRG	4	address of PP COMREG
38	OATFLAG	1	Flag byte
38	OATSEC	X'80'	system is secured
39	OATYYDDD	6	todays date (.YYDDD)
39	*	1	blank in 20th century
3A	OATYY	2	YY
3C	OATDDD	3	DDD
3F	*	1	to force boundary
40	OATTRTAB	4	address of ASCII translate table
44	OATSPID	8	subpool-ID for GETVIS
4C	OATPBDY	14	begin and end of partition
4C	OATPBEG	4	begin of partition
50	OATPEND1	4	end of partition
54	OATPEND	4	end of partition (incl. GETVIS)
58	OATRBEG	4	begin of real partition
5C	OATREND	4	end of real partition
60	OATOPN	4	number of open DTFs in partition

DTFMT Open DTF List (ODL)

Offset (Hex)	Field Name	Bytes and Bit Pattern	Description
0	ODL	80	
0	ODLNAME	4	Control block name ('ODL ')
4	ODLFWD	4	pointer to next ODL or 0
8	ODLNOPN	4	number of open DTF's in partition
C	ODLNENT	4	number of entries in this ODL
10	ODLDTF	70	14 entries
80	ODLEND	0	end of ODL

DTFMT - ODL Entry

Offset (Hex)	Field Name	Bytes and Bit Pattern	Description
0	ODLENTRY	8	STRUCTURE FOR AN ODL ENTRY
0	ODLDTFP	4	POINTER TO DTF
4	ODLFLAG	1	FLAG BITS
	ODLOPEN	X'80'	FILE IS OPENED
	ODLEOF	X'40'	FILE IS CLOSED DURING EOF
5	ODLDTFX	3	DTFX POINTER

DTF Type Code

DTF Type Code (Byte 20) of DTF Table	DTF	Description
X'00'	DTFCD	Combined files
X'02'	DTFCD	Reader and 3881 Optical Mark Reader files
X'03'	DTFCN	Console
X'04'	DTFCD	Punch files
X'05'	DTFCD	Reader files on 2560, 5424/5425
X'07'	DTFPR	Printer files on 2560
X'08'	DTFPR	Printer files
X'09'	DTFOR	Optical Reader files except 3881 and 3886 files
X'0A'	DTFOR	Optical Reader files (HEADER=YES)
X'0B'	DTFMR	Magnetic Ink Character Recognition (MICR) and Optical Reader/Sorter files
X'0C'	DTFDR	3886 Optical Character Reader files
X'10'	DTFMT	Unlabeled tape work files
	DTFCP	Unlabeled tape work files (compiler). (Note 1)
X'11'	DTFMT	Nonstandard or unlabeled tape files
X'12'	DTFMT	Standard labeled, output tape files
	DTFPH	Standard labeled, output tape files (physical IOCS)
X'13'	DTFMT	Standard labeled, input tape files (read backw)
X'14'	DTFMT	Standard labeled, input tape files (read forw)
X'15'	DTFMT	Standard labeled tape work files
X'1A'	DTFDU	Diskette Input/Output Unit files
X'20'	DTFSD	Sequential DASD work files and data files
	DTFCP	DASD work files (compiler)
X'21'	DTFPH	Sequential DASD files, MOUNTED=SINGLE (physical IOCS)
X'22'	DTFDA	Direct access files
X'23'	DTFPH	Direct access files, MOUNTED=ALL (physic. IOCS)
X'24'	DTFIS	Indexed sequential, LOAD file
X'25'	DTFIS	Indexed sequential, ADD file
X'26'	DTFIS	Indexed sequential, RETRVE file
X'27'	DTFIS	Indexed sequential, ADDRTR file
X'28'	ACB	Access Method Control Block for VSE/VSAM
X'30'	DTFCP	Compiler file for DOS Version 1 (Note 1)
X'31'	DTFCP	Compiler file for DOS Versions 2 onward
X'32'	DTFCP	Compiler file for DOS Vers. 2 onward (Note 2)
X'33'	DTFDI	Device independent system unit files
X'40'	DTFBT	Basic Telecommunications Access Method - Extended Support (BTAM-ES) file (Notes 3 and 4)
X'60' - X'67'		

Notes:

1. DTF type is X'30' except for tape or DASD assigned to units SYS000 to SYSnnn. In this case, the DTFCP open phases change the DTF type to X'10' for tape work files, or X'20' for DASD work files.

2. DTF type is X'32' except for DASD assigned to units SYS000 to SYSnnn. In this case, the DTFCP open phases change the DTF type to X'20' for DASD work files.

3. The following control unit codes are ORed into the low-order 4 bits of the DTF type code.

Control Unit	Code
7770	1
2848	3
2701	4
2702	5
2703	6

4. The DTF tables for BTAM-ES files are not documented in this manual. They are documented in the BTAM-ES publications.

ISAM RPS or DAM DASD Device Independent Extension Work Area

0 (0) Channel Program (Variable Length)	
Work Space	172 (AC) (Except ISAM) Sector values (up to 4)
176 (B0) Address of original channel program	180 (B4) Address of original logic module
184 (B8) 72 Byte Register Save Area	
256 (100) Additional Work Space 256 bytes for DAM 128 bytes for ISAM	

CHAPTER 4. SUPERVISOR CONTROL BLOCKS AND AREAS

SUPERVISOR STORAGE ALLOCATION

Generation Macro	Macros Called	Generated Code	Base Regs. Used
SUPVR		None, this macro only sets globals	–
FOPT		None, this macro only sets globals	–
IOTAB	SGEND	DSECTS, EQUATES	–
	SGLOWC	HW/SW interface (PSWs, log-out areas, etc.)	–
		Various constants and tables must be below 4K. CRTGEN, PIB tables, exit tables, I/O tables, foreground communication regions, etc., having Y-type address pointers in low storage, must be below 32K.	–
	SMICR	External interrupt handler	R14
		C-transient, B-transient, and A-transient area	– –
	SGEFCH	Temporary library control blocks and TFIX table for pageable FETCH routines.	R9
	ASYCODE	Asyncronous operator communication routines.	R9
	ASYTAB	Asyncronous operator communication tables	R9
	SGATAB	Tables having A-type address pointers in low storage (CRTSAV, SDAGDT, ISTAVT, DTSVECTB, SCYVECTB).	–
	DISP	Task selection	R6
	SGNUC	Interrupt handler, job accounting in-line routine.	R13
	SGPCK	Program check handler	R13
		(DTSMCIC) ICCF Monitor intercept routine	R14
	SGAFCH	Fetch data section (CCWs, control blocks)	R11
	SGDFCH	Fetch overall logic and directory search	R9
	SGCCWT	CCW translation for 370 mode	R8,R9
	SGCCWF	CCW analysis and fixing routine for ECPS:VSE mode	R8,R9
	SGSVC	Various SVC routines	R13
	SGSVCX	Various SVC routines	R13
	MCRAS	Machine/Channel Check handler, RTA	R15
	SGSTAR	System track algorithm routines	R9

Supervisor Storage Allocation (cont...)

Generation Macro	Macros Called	Generated Code	Base Regs. Used
IOTAB (cont)	SGIOS	SVC0 (EXCP) and SVC15 (SYSIO) routines	R13
		(SGSCHED) Channel scheduler routine	R13
		(IOINTER) I/O interrupt handler	R9
		(SGMIH) Missing interrupt handler	R13
		(SGDSK) Disk error recovery routine	R13,R14
		(SGSERI) Service task interface and data	R12
	SGCFCH	Fetch SVC routines	R13
	SGERP	Interface to ERP transients	R13
	SGAP	Asynchronous processing SVC routines	R13
	SGTINF	Tasking interface routines	R12,R13
	DTSSVCIC	ICCF SVC intercept routine	R14
	DTSSVCIN	ICCF SVC routine	R14
	SGRM	Resource management SVC routines	R13
		Tasking control blocks	R12
	SGLOCK	LOCK, UNLOCK routines	R13
	SGAM	CDLOAD, GETVIS, and FREEVIS routines	R14
		(SGAMSUBR) Subroutines of SGAM	R14
	SGNPGR	Allocate programmer logical units (LUBS)	R13
	SGBFCH	Input buffer, program fetch and I/O processing	R9
	SGSER	Automatic Volume Recognition and related SVC routines	R13
		SGSLDUP, SLD update routine, Disk sharing only	R14
	SGACF	Security and Audit support	R13
	SGXECB	Cross partition common SVC routines	R13
	SGACCT	GETJA SVC routine Change/Display Priority SVC routines	R13
	SGINF	Logical SV/PP common SVC routines	R12
	SGIUCV	IUCV-VCNA connection	R13
	SGPREAL	Get/free processor storage for 370	R9
	SGPMR	Page manager	R9
		(SGPSVC) VIOPOINT service	R9
		(SGPLLEV) Load leveler	R15
		(SGPFIX) Fixing routines	R9
		(SGPOPT) Page in SVCs	R9
		(SGPDATA) Data for page manager	R8
		IPL initialization routines CCW translation copy buffers	R7,R9

VSE SUPERVISOR CALL TABLE

SVC Code		Imperative Macro that issues the SVC	Activation option to be specified	Function
DEC	HEX			
0	00	EXCP	none	Execute channel program
1	01	FETCH	none	Fetch a phase, except a transient phase
2	02		none	Fetch a logical transient phase ($$B.....)
3	03		none	Quisce I/O
4	04	LOAD/SLOAD	none	Load a phase
5	05	MVCOM	none	Modify the partition communication-region
		if issued by ERP-Task	none	Fetch a physical transient ($$A.....)
6	06	CANCEL	none	Cancel a problem program or a task
7	07	WAIT	none	Wait for the posting of a control block (CCB, IORB, ECB, TECB)
8	08		none	Transfer control from a logical transient to a problem program
9	09	LBRET	none	Return from the problem program to the logical transient which issued the SVC 8
10	0A	SETIME	none	Set interval timer
11	0B		none	Final return from a logical transient
12	0C		none	Reset switches in the partition communication region (COMREG)
13	0D		none	Set switches in the partition communication region (COMREG)
14	0E	EOJ	none	Terminate a job and go to job control for end of job step processing
15	0F	SYSIO	none	Head queue I/O request and execute the channel program
16	10	STXIT PC	none	Establish/reset linkage to user's PC routine for program check interrupts
17	11	EXIT PC	none	Return from the user's PC routine
18	12	STXIT IT	none	Establish/reset linkage to user's IT routine for interval timer interrupts
19	13	EXIT IT	none	Return from the user's IT routine
20	14	STXIT OC	none	Establish/reset linkage to user's OC routine in case of attention MSG command
21	15	EXIT OC	none	Return from the user's OC routine
22	16		none	SEIZE or RELEASE the system; enable or disable for external and I/O interrupts; set the key in a user's PSW
23	17		none	Store the LOAD ADDRESS of a phase at a defined user address

VSE Supervisor Call Table (cont...)

SVC Code		Imperative Macro that issues the SVC	Activation option to be specified	Function
DEC	HEX			
24	18	SETIME	none	Set TIMER INTERVAL and establish accessability to user's TECB
25	19	HALTIO	none	Issue an HDV for a telecommunication device or for any device if issued by OLTEP.
26	1A		none	Validate address limits
27	1B		none	Issue an HDV for a telecommunication device without dequeueing the CHANQ entry
28	1C	EXIT MR	MICR=type in SUPVR	Return from user's stacker select routine
29	1D	WAITM	none	Wait for the posting of one of the control blocks specified
30	1E		none	Reserved
31	1F		none	Reserved
32	20		none	Reserved
33	21	COMREG	none	Force task selection
34	22	GETIME	none	Provide the time and update
35	23		TRKHLD=YES in FOPT	Hold a track for exclusive use by the requesting task
36	24	FREE	TRKHLD=YES in FOPT	Free a track held by the requesting task
37	25	STXIT AB	none	Establish/reset linkage to user's AB routine for abnormal termination of a task
38	26	ATTACH	none	Initialize a subtask and establish its processing priority
39	27	DETACH	none	Terminate a subtask; free resources that might be held by the subtask
40	28	POST	none	Indicate occurrence of an event and ready any waiting task
41	29	DEQ	none	Indicate that a previously enqueued resource is available again
42	2A	ENQ	none	Prevent two or more tasks from simultaneously manipulating a shared resource (e.g. data area)
43	2B		none	Reserved
44	2C		none	Force a unit check record to be written onto the recorder file
45	2D		none	Reserved
46	2E		none	Allow OLTEP to run in supervisor state
47	2F	WAITF	MICR=type in SUPVR	Support the multiple wait macro WAITF for MICR type I/O routines
48	30		none	Fetch a CRT-transient phase
49	31		none	Allow ACF/VTAM to initiate the execution of a channel program

VSE Supervisor Call Table (cont...)

SVC Code		Imperative Macro that issues the SVC	Activation option to be specified	Function
DEC	HEX			
50	32		none	Used by LIOCS to channel user indicating illegal SVC
51	33		none	Make directory entry information for a phase available to the requesting task
		HIPROG	none	Calculate the highest address of an overlay structure of phases or of one phase only and store it in the COMREG
52	34	TTIMER	none	Return the remaining time interval or cancel a time interval
53	35		none	Allow ACF/VTAM to schedule a user exit in an application program
54	36		none	Release page frames to selection pool (applies only to 370 mode of operation)
55	37		none	Allow SDAID to acquire processor storage needed for program initialization (applies only to 370 mode of operation)
56	38	CPCLOSE	MODE=VM or MODE=370 in SUPVR	Support the VSE/POWER-CP interface when VSE operates under VM/370
57	39	GETPRTY	none	Return partition priorities to the requesting task
		SETPRTY	none	Change partition priorities as specified
58	3A	INVPART	none	Initialize partition
59	3B	INVPAGE	none	Initialize table or invalidate pages
60	3C	GETDADR	none	Return the virtual equivalent of a real I/O area plus offset
61	3D	GETVIS	none	Request allocation of storage within the same partition or within the SVA
62	3E	FREEVIS	none	Free storage requested through a GETVIS macro
63	3F	USE	none	Indicate system resource is in USE
64	40	RELEASE	none	RELEASE a system resource
65	41	CDLOAD	none	Load a phase in the requesting partition's GETVIS area unless that phase is already in the SVA
66	42	RUNMODE	none	Return the system's operating mode
67	43	PFIX	none	FIX pages in processor storage
68	44	PFREE	none	FREE pages in processor storage
69	45	REALAD	none	Return the REAL address corresponding to a given virtual address

VSE Supervisor Call Table (cont...)

SVC Code DEC	HEX	Imperative Macro that issues the SVC	Activation option to be specified	Function
70	46	VIRTAD	none	Return the virtual addr. corresponding to a given real address
71	47	SETPFA	none	Establish or terminate linkage to a user Page Fault Appendage routine
72	48	GETCBUF	none	GET copy buffer for IDAL of tape ERP
		FREECBUF	none	FREE copy buffer for IDAL of tape ERP
73	49	SETAPP	none	Allow linkage to channel-end appendage routine
74	4A	PFIXREST	none	FIX page(s) in processor storage for restart
		PFIXCHPT	none	Build parameter list for PFIXREST during checkpointing
75	4B	SECTVAL	RPS=YES in FOPT	Calculate a sector value for a disk device with the RPS feature
76	4C		none	Initiate recording on VM recorder file
77	4D	TRANSCSW	none 370 mode only	Returns the virtual address of an ERP CCW address copied from the pertinent CSW
78	4E	CHAP	none	Change the processing priority of the requesting task
79	4F		none	Reserved
80	50	SETT	TTIME=part-id in FOPT	Set task time interval
81	51	TESTT	TTIME=part-id in FOPT	Return remaining task time interval or cancel the time interval
82	52		none	Set monitor call and/or branch, for ICCF
83	53	ALLOCATE	none	Allocate real or virtual partitions
84	54	SETLIMIT	none	Set partition sizes
85	55	RELPAG	none	Release the contents of one or more pages
86	56	FCEPGOUT	none	Force a page-out operation for more pages
87	57	PAGEIN	none	Request a page-in operation for more pages
88	58	TPIN	none	Start TP balancing
89	59	TPOUT	none	Stop TP balancing
90	5A	PUTACCT	JA=YES in IPL SYS-CMD	Provide interface with VSE/POWER for additional user-provided account information
91	5B		JA=YES in IPL SYS-CMD	Provide interface with VSE/POWER for standard account information
92	5C	XECBTAB	none	Define, delete, or check an entry in the cross-partition ECB table
93	5D	XPOST	none	Set the traffic bit in a cross-partition ECB and ready any waiting task
94	5E	XWAIT	none	Wait for a cross-partition ECB to be posted

VSE Supervisor Call Table (cont...)

SVC Code		Imperative Macro that issues the SVC	Activation option to be specified	Function
DEC	HEX			
95	5F	EXIT AB	none	Return from a user's abnormal termination routine
96	60	EXIT TT	TTIME=part-id in FOPT	Return from a user's task timer exit routine
97	61	STXIT TT	TTIME=part-id in FOPT	Establish/reset linkage of user task's timer exit routine for task time interval end
98	62	EXTRACT	none	Extract system control information
		MODCTB	none	Modify a PUB2 table entry
99	63	GETVCE	none	Return a specific volume characteristics and/or track balance information
100	64	PFIX PFREE	none (ECPS:VSE mode only)	Fix or free a page in the SYSTEM GETVIS area
101	65	MODVCE	none	Update the volume characteristics table
102	66	GETJA	JA=YES in IPL SYS-CMD	Update the fields in the requesting partition's job accounting table
103	67		none	Execute I/O operations for SYSFIL on FBA device, if FBA supported
104	68	EXTENT	none	Add, return, or delete Disk extent information
105	69	SUBSID	none	Accept, return, and delete subsystem identification information
106	6A		none	Set the storage key for a specific area to the value in Reg. 0 (ICCF)
107	6B		none	Release a device that was 'in use'
		DEVUSE		Force a device to be set 'in use'
		GETFLD		Retrieve task related information
		MODFLD		Modify task related information
		RLOCK		Obtain access to a specified resource or wait for it
		SENTER		Enter a subsystem
		SLEAVE		Leave a subsystem
		TREADY		Post or cancel a task
		TSTOP		Deactivate current task or partition
		VIO POINT		Point to VIO control block (VIORB)
108	6C	SECHECK	SEC=nn in IPL SYS-CMD	Check user's authority for accessing the specified resource
109	6D	PAGESTAT	none	Return status of a page or a set of pages
110	6E	LOCK/UNLOCK	none	Protect or release a serially re-usable resource against concurrent access of two or more tasks
111	6F		none	Reserved

VSE Supervisor Call Table (cont...)

SVC Code		Imperative Macro that issues the SVC	Activation option to be specified	Function
DEC	HEX			
112	70	MSAT	none	Build, return, or delete stored assignment information
113	71	XPCC	none	Cross-partition communication services
114	72	VIOC	none	Allocate, deallocate or extent VIO file
115	73	PWROFF	none	Software initiated power-off for 4361
116	74	NPGR	none	Allocate or reallocate programmer LUB's
117	75		none	Reserved
118	76	CPCOM	MODE=VM or MODE=370 in SUPVR	CP command interface (CPCOM macro)
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
140	8C		none	Reserved
141	8D	VSIUCV	MODE=VM in FOPT	Provide subsystem support for VM/VCNA (VTAM Communication Network Application)
142	8E		none	Reserved
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
255	FF		none	Reserved

CANCEL CODE TO MESSAGE CODE CROSS REFERENCE

Cancel Code	Message Code	Descriptive Part of Message (or Condition)
X'00'	-----	In all cases default value except those listed
08	0V16I	Cancel request from subsystem
09	0V15I	Cancel request from LIOCS
0A	0S21I	Processing error in access control
0B	0S20I	Access control violation
0C	0S19I	Execution failure in ICCF interactive partition
0D	0V13I	Program check in subsystem or appendage
0E	0V14I	Page fault in subsystem or appendage
0F	0P80I	Invalid 'Read from/or write to' system file on FBA device
10	-----	Normal EOJ
11	0V07I	No channel program translation for unsupported device
12	0V06I	Insufficient buffer space for channel program translation
13		Reserved
14	0V04I	Page pool too small
15	0V02I	Page fault in disabled program
16	0V11I	Error in privately translated CCW
17	0S02I	Program request (Same as 23 but causes dump because subtasks were attached when maintask issued CANCEL macro)
18	-----	Eliminates cancel message when task issues DUMP macro with subtasks attached
19	0P74I	I/O operator option
1A	0P73I	I/O Error
1B	0P82I	Channel failure
1C	0S14I	CANCEL ALL macro
1D	0S12I	Maintask termination
1E	0S13I	I/O error on lock file
1F	0P81I	CPU failure
20	0S03I	Program check
21	0S04I	Illegal SVC
22	0S05I	Phase not found
23	0S02I	Program request
24	0S01I	Operator intervention (cancel)
25	0P77I	Invalid address
26 *	0P71I	SYSxxx not assigned (unassigned LUB code)
27	0P70I	Undefined logical unit (invalid LUB code in CCB)
28	0S35I	Phase too long (does not fit in LTA or partition)
29	0P92I	Invalid Sub-library structure
2A	0V10I	I/O error on page data set
2B	0P84I	I/O error during fetch from private core image library
2C	0V09I	Illegal parameter passed by PHO routine
2D	0P881	Failing storage block (program cannot be executed)
2E	0S16I	Invalid resource request (possible deadlock)
2F	0V03I	More than 255 PFIX requests for 1 page
30	0P72I	Reading past /& statement (on SYSRDR or SYSIPT)
31		Reserved
32	0P76I	Invalid DASD address
33	0P79I	Invalid first CCW
34	0P93I	GETVIS space exhausted
35	0P85I	Job control open failure
36	0V08I	Program check or page fault in I/O appendage routine
37		Reserved
38	0V11I	Wrong privately translated CCW
39	0V12I	Invalid CCW chain for SYSLOG
3A	0V17I	Spool request out of sequence

Cancel Code to Message Code Cross Reference (cont...)

Cancel Code	Message Code	Descriptive Part of Message (or Condition)
X'40'	0V95I	ACF/VTAM error (termination of task)
41	0V96I	ACF/VTAM error (invalid condition code)
42	0P86I	Violated DASD file protection
FF		Multiple cancel condition (see SYSLST for details)
XX	0P78I	Unrecognized cancel code
	0P83A **	Supervisor catalog failure
	0P87A **	IPL failure

* If the CCB/IORB is unavailable, the logical unit is SYSxxx.

** This cancel code is not significant in case of a supervisor catalog or IPL failure, because the system is placed in a WAIT state without any further processing by the terminator.

XX Any other DIGITS

SAVE AREAS

The addresses of the various save areas allocated by the system can be found in the appropriate TCB table.

The LTA save area can be identified via -eye catcher- 'CNCLINFO in the supervisor or via TIBATAB -64.
If TIBFLAG=LTAACT and TERMACT, the LTA save area will be found in SAVARPTR, and the PP save area pointer in SAVARPT2.

Layout of Problem Program (PP) and LTA Save Area

0 (0) Program Name (Logical Transient Name) 7 (7)				
Program Status Information (- Return PSW -)				
8(8)	9(9)	10(A)	11(B) 12(C)	13(D) 15(F)
X'40' PER X'04' DAT X'02' I/O X'01' EXT	Protection Key and Mask (CMWP) Bits	(Note 1)	Zero	Instruction Address
16 (10) General Register save area (Reg. 9 through Reg. 8) 79 (4F)				
80 (50) Reserved 81 (51)	82 (52) (Note 2) (Note 3) 87 (57)			
88 (58) Floating Point Reg. save a. (Reg. 0 through 6) 119 (77)				

Notes:

1. Byte 10
   - bits 0-1 = Reserved (zero)
   - bits 2-3 = Condition Code
   - bits 4-7 = Program Mask
2. Bytes 82 - 87 (PP Save Area)
   - main task: Date of job begin
   - subtask: 82 (52) - 83 (53) : Reserved
   - 84 (54) - 85 (55) : Task id
   - 86 (56) : Key of ICCF pseudo-partition
   - 87 (57) : Reserved
3. Bytes 82 - 87 (LTA Save Area)
   - Reserved

Layout of User-Exit-Save Area (STXIT)

Interrupt Status Information				
0(0)	1(1)	2(2) 3(3)	4(4)	5(5) 7(7)
Reserved	Protection Key and Mask bits from PSW byte 1	Interruption Code	(Note 1)	Instruction Address
8 (8) General Register save area (Reg. 0 through 15) 71 (47)				

Note:

1. Byte 4
   - bits 0-1 = Instruction Length Code
   - bits 2-3 = Condition Code
   - bits 4-7 = Program Mask

The address of the save area specified by the user in the STXIT macro parameter is stored in the appropriate table (TCB, PCB or TTTAB).

TASK AND PARTITION KEY DEFINITIONS

Storage Protection Key

Each partition in VSE is assigned a unique storage protection key. It's the hexadecimal representation of the value 16*n, where
0 <= n <= number of partitions.

Storage protection keys are assigned depending on the number of partitions according to the following scheme:

		PIK Value in COMREG										
Part.	Part.	Number of Partitions										
id	name	12	11	10	9	8	7	6	5	4	3	2
00	SYS	00	00	00	00	00	00	00	00	00	00	00
01	BG	10	10	10	10	10	10	10	10	10	10	10
0C	F1	C0	B0	A0	90	80	70	60	50	40	30	20
0B	F2	B0	A0	90	80	70	60	50	40	30	20	
0A	F3	A0	90	80	70	60	50	40	30	20		
09	F4	90	80	70	60	50	40	30	20			
08	F5	80	70	60	50	40	30	20				
07	F6	70	60	50	40	30	20					
06	F7	60	50	40	30	20						
05	F8	50	40	30	20							
04	F9	40	30	20								
03	FA	30	20									
02	FB	20										

PARTITION IDENTIFICATION

Normally a partition is identified by its unique storage protection key. Due to its additional use, a special storage protection key value is often called a 'Partition Identification Key' (PIK). In some cases a partition is identified by a 'Partition Identifier' (PID) value, which is just the value PIK/16.

Note: The PID values are contained in the first digit of a storage protection key.

Task Identification

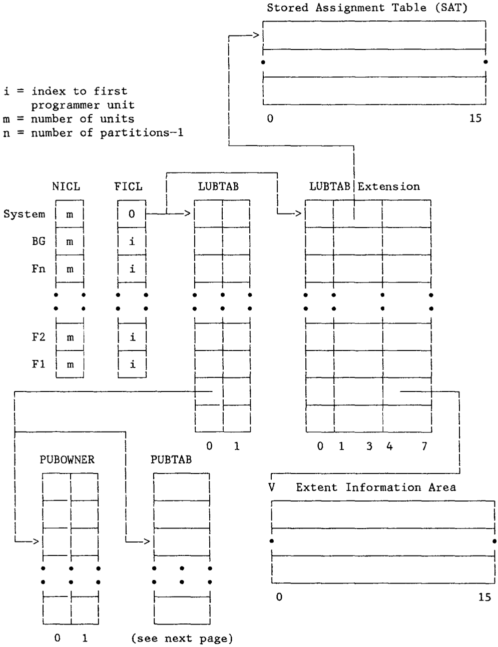
Tasks are identified by hexadecimal numbers 1 to X'FF'. The following table shows the task identifier (TID) values and their assignments to particular tasks:

System Task			Main Task		Sub Task	
TID			TID		TID	
00	Unused		20	AR	30	**
01	SNS	- CCH/MCAR task to issue	21	BG	31	**
		SENSE command	22	F1 *	32	**
02	DSK	- resident disk error	23	F2 *	33	**
		recovery task	24	F3 *	34	**
03	RAS	- CCH/MCAR maintask	25	F4 *	35	**
04	PMR	- page manager task	26	F5 *	..	
05	Unused		27	F6 *	nn	**
06	PGN	- page in task	28	F7 *		
07	SUP	- fetch task	29	F8 *		
08	DIR	- directory read task	2A	F9 *		
09	CRT	- display operator console	2B	FA *		
		support task	2C	FB *		
0A	ASY	- asynchronous operator	2D	Unused		
		communication support task	2E	Unused		
0B	ERP	- error recovery task	2F	Unused		
0C	LCK	- lock service task				
0D	Unused					
0E	LOG	- logger task				
0F	SVT	- automatic volume recognition				
		task				
10	Unused					
..						
1F	Unused					
20	AR	- attention routine task				

* Depending on the number of partitions, all or some of these identifiers may be unused (in descending order of values).

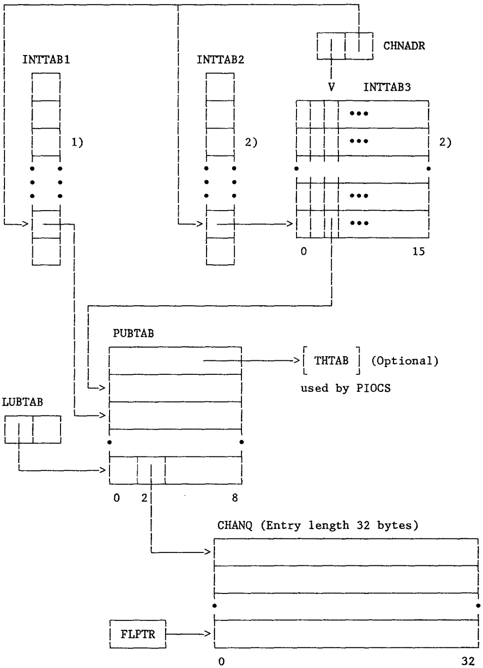
** A pool of subtasks is created and maintained by the supervisor. The size of this pool is given by the maximum number of subtasks active at the same time.

I/O TABLE INTERRELATIONSHIP



i = index to first
programmer unit
m = number of units
n = number of partitions-1

(see next page)

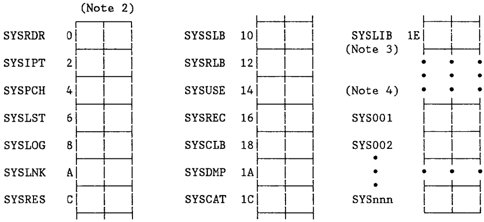


1) Initialized by IPL.
2) Optionally allocated and initialized by IPL.

Logical Unit Block Entry (Note 1):

Bytes Dec	 Hex	Description
0	0	PUB index of device assigned to this logical unit X'FF' if no PUB is assigned or X'FE' if I/O is to be ignored for this log. unit
1	1	Reserved

Logical Unit Block Table (LUBTAB)



Notes:

1. Null entries X'FFFF' are generated at supervisor generation time.
2. There are 14 externally known system LUBs and one internally used for label access method.
3. System LUBs used by dynamic assignments.
4. The total number of system LUBs is a constant.

LUBTAB Extension Table

Bytes Dec	 Hex	Label	Description
0	0	LUBXFLG	Flag Byte
		LUBXPA	X'80' Perm. altern. assignment stored
		LUBXTA	40 Temp. altern. assignment stored
		LUBXPE	20 Permanent assignment stored
			10 -01 Reserved
			(LUBXPA and/or LUBXTA is on)
1-3	1-3	LUBXADR	Pointer to first Stored Assignment Table entry (SAT)
			(LUBXPA and LUBXTA both off)
1	1		Reserved
2-3	2-3	LUBXPER	Stored permanent assignment
OPTIONAL DASDFP=YES 4-7	 4-7	LUBXEPT	Pointer to first EXTENT INFORMATION chain entry or zero if no EXTENT INFORMATION available

The start address of the LUB Extension table is stored by IPL in bytes 168-171 (X'A8-AB') of the Partition Communication Region.

Stored Assignment Table Entry (SAT)

Bytes			
Dec	Hex	Label	Description
0	0	SATFLG	Flag byte
			X'80' Reserved
			40 Reserved
		SATPE	20 Permanent Assignment saved in this entry
			10 Reserved
			08 Reserved
			04 Reserved
			02 Reserved
			01 Reserved
1–3	1–3	SATNEXT	Pointer to next assign entry in the chain
4	4	SATEOCH	Offset within SATSAV of next free entry
5	5	SATEOPCH	Offset within SATSAV of saved permanent assignment
6–7	6–7	SATSAV	Space for saving permanent assignment (max. of 5)
•••	•••		
14–15	E–F		

The address of the Stored Assignment Table Entry (SAT) can be found in LUBTAB Extension Byte 1.

Extent Information Entry

The LUB extension table entry contains a pointer (4-7) to a chain of Extent entries for DASD File Protection.

Bytes			
Dec	Hex	Label	Description
0	0	EXBFLG	Flag Byte
		EXBREAD	X'80' Allow READ access only (no multi–track operation)
		EXBSHORT	40 Extent information is CC only
			20 Reserved
			10 Reserved
			08 Reserved
			04 Reserved
			02 Reserved
			01 Reserved
1–3	1–3	EXBNXT	Pointer to next Extent entry in the chain or zero if this is the last Extent entry
4–7	4–7	EXBHI	High Extent Limit
			CKD Device Cylinder+Head No.
			FBA Device Physical Block No.
8–11	8–B	EXBLOW	Low Extent Limit
			CKD Device Cylinder+Head No.
			FBA Device Physical Block No.
12–13	C–D	EXBCOUNT	Usage count for this extent
14–15	E–F		Reserved

Command Control Block (CCB)

Count	Transmission Information	CSW Status Bits	Type Code and Logical Unit	Used by LIOCS or 3895 PIOCS	CCW Addr.	Used by Physical IOCS	CCW Address in CSW	Optional Sense CCW
0 1	2 3	4 5	6 7	8	9 11	12	13 15	16 23

Byte(s)	Description		set on by
0–1 RESIDUAL COUNT	Number of bytes that have not been processed by the channel BTAM (370 mode only): Number of needed copy blocks		
2–3 TRANSMITTING INFORMATION between PIOCS and PROBLEM PROGRAM (Pr.Pr.)	Byte 2		set on by
	Bit 0:	Traffic Bit (WAIT). (Note 5)	PIOCS
	Bit 1:	End–of–File, (Note 2) PRT1–UCSB Parity Check.	PIOCS
	Bit 2:	Irrecoverable I/O error was encountered.	PIOCS
	Bit 3:	Prevent Cancelation on Irrecoverable I/O error.	Pr.Pr.
	Bit 4:	Return DASD and/or DISKETTE Data Checks, Return 5424/5425 not ready, Indicate action–type messages for DOC.	Pr.Pr.
	Bit 5:	Post at Device End. (Note 5)	Pr.Pr.
	Bit 6:	Return TAPE or DASD Read Data Check, (Note 3) Return 2560 Data Check, Return 2520, 2540, 2560, 3881, or 5424/5425 Equipment check, Return 3505, or 3525 Permanent error, (Note 7) Return 3203 or PRT1 errors, Return 3895 errors. (Note 9)	Pr.Pr.
	Bit 7:	User handles I/O errors. . (Note 8)	Pr.Pr.
	Byte 3		set on by
	Bit 0:	DASD Data Check in Count Area, 3330, 3340 or 3350 permanent error, 1287/1288 Data Check, 1419D SCU Not Operational, 3203 or PRT1 print check/equipment check, 3540 Special Record transferred.	PIOCS
	Bit 1:	DASD Track Overrun, 1419 Intervention required, 1287 Keyboard Correction in Journal Tape Mode, PRT1 Print Quality/Equipment check	PIOCS
	Bit 2:	DASD End–of–Cylinder, 1419, 1287 ,1288 Hopper Empty (Note 4) PRT1/2245 Line position error.(Note 6)	
	Bit 3:	1287, 2520, 2540 or 3881 Equipment Equipment Check, 2560, 3203, 5424/5425 Data/equipment check, 3505 or 3525 Permanent Error, (Note 7) TAPE Read Data check, DASD Data Check, PRT1 Print Check/Data Check, Diskette Data Check.	PIOCS

Note: Pr.Pr. stands for Problem Program

Command Control Block (CCB) (cont...)

Count	Transmission Information	CSW Status Bits	Type Code and Logical Unit	Used by LIOCS or 3895 PIOCS	CCW Addr.	Used by Physical IOCS	CCW Address in CSW	Optional Sense CCW
0 1	2 3	4 5	6 7	8	9 11	12	13 15	16 23

Byte(s)	Description	
2–3 (cont.)	Byte 3	set on by
	Bit 4: CARD Unusual command sequence, DASD No Record Found, 1287/1288 Document Jam or Torn Tape, PRT1 UCSB, PRT1 UCSB Parity Check (Command retry), 5424/5425 not ready.	PIOCS
	Bit 5: user does not expect NO RECORD FOUND condition,	Pr.Pr.
	Bit 6: PRINTER Carriage Channel 9 Overflow, DASD Verify error; 1287 Late Stacker select (Doc. Mode), 1288 End of Page.	PIOCS
	Bit 7: Channel Program is not retryable (Command Chain – Retry will be started from failing CCW).	Pr.Pr.
4–5 CSW STATUS BYTES	Byte 4	Byte 5 (Note 1)
	Bits:	Bits:
	0 (32): Attention	0 (40): Program Controlled Interruption
	1 (33): Status Modifier	1 (41): Incorrect Length
	2 (34): Control Unit End	2 (42): Program Check
	3 (35): Busy	3 (43): Protection Check
	4 (36): Channel End	4 (44): Channel Data Check
	5 (37): Device End	5 (45): Chan. Contr. Check
	6 (38): Unit Check	6 (46): Interf. Contr. Chk
	7 (39): Unit Exception	7 (47): Chaining Check
6–7 TYPE code and LOGICAL UNIT	Byte 6	
	B'1x00x00x' = User-translated CCB B'x1x0x00x' = BTAM CCB B'0x10x00x' = System-translated CCB B'xxx0100x' = CCB for physical unit B'xxx00001' = CCB for program logical unit B'xxx00000' = CCB for system logical unit	
	Byte 7	
	Hexadecimal representation of SYSnnn:	
	SYSRDR = 00 SYSSLB = 07 SYSLUB = 0e-ff	
	SYSIPT = 01 SYSRLB = 08 SYS000 = 00	
	SYSPCH = 02 SYSUSE = 09 SYS001 = 01	
	SYSLST = 03 SYSREC = 0A SYS002 = 02	
	SYSLOG = 04 SYSCLB = 0B .	
	SYSLNK = 05 SYSDMP = 0C .	
	SYSRES = 06 SYSCAT = 0D SYS255 = FF	
8 LIOCS Information	Buffer Offset:	
	ASCII Input Tapes X'00' — X'63'	
	ASCII Output Tapes Fixed X'00'	
	Variable X'00' or X'04'	
	Undefined X'00'	
	2501 Read ahead support X'80' (2501 Read ahead support is active)	
	SNS task I/O request X'80' (I/O error on alternate channel)	
	3895 Error information (Note 9)	

Note: Pr.Pr. stands for Problem Program

Command Control Block (CCB) (cont...)

Count	Transmission Information	CSW Status Bits	Type Code and Logical Unit	Used by LIOCS or 3895 PIOCS	CCW Addr.	Used by Physical IOCS	CCW Address in CSW	Optional Sense CCW
0 1	2 3	4 5	6 7	8	9 11	12	13 15	16 23

Byte(s)	Description
9-11 CCW ADDRESS	Virtual or real addr. of CCW associated with this CCB (Byte 6 bit 0 = 1 Address is a REAL address) (Byte 6 bit 0 = 0 Address is a VIRTUAL address)
12 PIOCS Information	X'80' CCB is used by ERP X'40' Channel Appendage Routine present X'20' Sense Information desired (Note 8) X'10' Reserved X'08' Reserved X'04' OLTEP Appendage available X'02' TAPE ERP Read Opposite Recovery in progress X'01' Reserved
13-15 CCW ADDRESS from CSW	Address of CCW pointed to by CSW at Channel End, (Byte 6 bit 0 = 1 Address is real) (Byte 6 bit 0 = 0 Address is virtual) or address of the appendage routine.
OPTIONAL 16-23 Sense CCW	8 bytes appended to the CCB when Sense Information is desired.

Notes:

1. Bytes 4 and 5 contain the status bytes of the CSW (Bits 32-47). If byte 2, bit 5 is ON and Device End occurs as a separate interrupt, bytes 4 and 5 will contain the accumulated status information. A tape read-backward I/O operation reading into loadpoint will force the UNIT EXCEPTION (Bit 47) to be turned on and the unit check bit to be reset (assuming byte 2 bit 7 and byte 12 bit 2 are both off).
2. Indicates /* or /& statement read on SYSRDR or SYSIPT. Byte 4, bit 7 (Unit Exception) is also on.
3. DASD data checks on count not returned.
4. For 1255/1259/1270/1275/1419, disengage. For 1275/1419D, I/O error in external interrupt routine (Channel Data Check or Bus-out check).
5. The traffic bit (Byte 2, bit 0) is normally set on at channel end to signify that the I/O was completed. If byte 2, bit 5 has been set on, the traffic bit and bits 2 and 6 in byte 3 will be set on at device end. See also Note 1.
6. This error occurs as an equipment check, data check or FCB parity check. For 2245, this error occurs as a data check or FCB parity check.
7. Byte 2, bit 6 must be set on to allow you to accept 3505, 3525 permanent errors. This bit is forced on by LIOCS if the user specified ERROPT for his input or output files. Byte 3, bit 3 is set on if a permanent error was encountered.
8. If User Error Routine is specified and the user needs the sense information to further process the error, byte 12, bit 2 must also be set. Otherwise, the supervisor error routine will clear off the status on return and the sense information is not available.
9. 3895 error codes are returned in CCB byte 8. Refer to 3895 Document Reader/Inscriber Machine and Programming Description for information on these error codes.

Input/Output Request Block (IORB)

Bytes Dec	Bytes Hex	Description		
0- 1	1- 1	Residual count, Number of bytes which where not transferred by the channel		
2	2	Communication Byte 1 Set by Physical IOCS:		
		X'80'	WAIT Bit, Traffic Bit	(Note 1)
		X'40'	End-of-File on SYSRDR or SYSIPT, /* or /&	(Note 2)
		X'20'	Irrecoverable I/O error encountered	
			Set by Problem Program	
		X'10'	Prevent cancelation in case of irrecoverable I/O Error	
		X'08'	Reserved	
		X'04'	User wants to be posted at Device End	(Note 1)
		X'02'	Reserved	
		X'01'	Skip system error recovery (no Recovery Action)	
3	3	Communication Byte 2 Reserved for ERP return information.		
4	4	Device Status Information		(Note 3)
		X'80'	Attention	
		X'40'	Status modifier	
		X'20'	Control unit end	
		X'10'	Busy	
		X'08'	Channel end	
		X'04'	Device end	
		X'02'	Unit check	
		X'01'	Unit exception	
5	5	Channel Status Information		(Note 3)
		X'80'	Program controlled interrupt	
		X'40'	Incorrect length	
		X'20'	Program check	
		X'10'	Protection check	
		X'08'	Channel data check	
		X'08'	Channel control check	
		X'02'	Interface control check	
		X'01'	Channel Chaining check	
6	6	IORB and device identification information		
		X'80'	Reserved	
		X'40'	Reserved	
		X'20'	Copied IORB (370 mode only)	
		X'10'	Reserved	
		X'08'	Device is identified by PUB entry number	
		X'04'	Control Block is an IORB	
		X'02'	Reserved	
		X'01'	Device is identified by programmer Logical Unit	
7	7	LUB or PUB entry number in the appropriate table		

Byte 6 bit 4 off + 7 off		Byte 6 Bit 4 off 7 on	Byte 6 Bit 4 on
SYSRDR=00	SYSRLB=08	SYS000=00	PUB entry No. 00
SYSIPT=01	SYSUSE=09	SYS001=01	.
SYSPCH=02	SYSREC=0A	SYS002=02	.
SYSLST=03	SYSCLB=0B	.	.
SYSLOG=04	SYSDMP=0C	.	.
SYSLNK=05	SYSCAT=0D	.	.
SYSRES=06	SYSLUB=0E-FF	.	.
SYSSLB=07		SYS255=FF	FF

Input/Output Request Block (IORB) (cont...)

Bytes Dec	 Hex	Description
8	8	Reserved for Logical Input Output Control System (LIOCS)
9-11	9- B	Virtual address of the CCW associated with this IORB
12	C	Reserved for physical Input Output Control System (PIOCS)
		X'80' IORB is used by Error Recovery Procedure
		X'40' Reserved
		X'20' This IORB has an extension
		X'10' Reserved
		X'08' Reserved
		X'04' Reserved
		X'02' Tape ERP read opposite recovery in progress
		X'01' Reserved
13-15	D- F	Address+8 of last CCW that was executed
16	10	Fix Flag
		X'80' Fix List is already in compressed format (Each page to be fixed for channel program execution is covered only once within the FIXLIST)
		X'40' All pages are FIXED (The user has already fixed all the pages need for channel program execution)
		X'20' Reserved
		X'10' Reserved
		X'08' Reserved
		X'04' Reserved
		X'02' Reserved
		X'01' Reserved
17-19	11-13	Address of FIXLIST
20-21	14-15	IORB Version identification code
22-23	16-17	Special processing flags set by LIOCS
		Bit 0 SYSFIL request for FBA Device
		Bits 1-15 Reserved
OPTIONAL		
24	18	Parameter ID:
		Bit 0 Identifies the last optional Parameter
		Bits 1-7 Parameter ID
		B'0000000' ECB ID
		B'xxxxxxX' Reserved
25-27	19-1B	Address portion of optional Parameter
⋮	⋮	⋮
		Parameter ID:
		Parameter ID:

Notes:

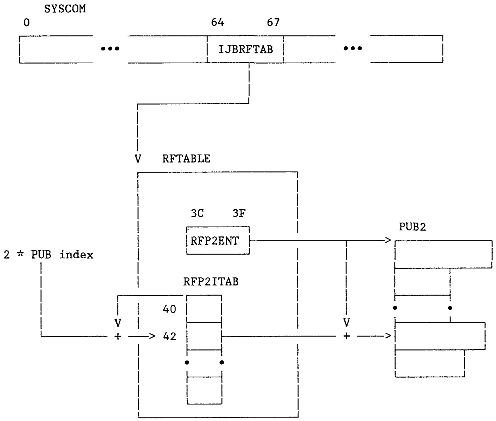
1. The WAIT Bit (byte 2, bit 0) is normally set on at Channel End to to signify that at least the data transfer is completed. If byte 2, bit 5, has been set on, the WAIT Bit is set at Device End.

2. Unit Exception (Byte 4, bit 7) is also turned on.

3. Bytes 4 and 5 contain the status bytes of CSW (Bits 32-47) which is always the accumulated status information received so far.

PHYSICAL UNIT BLOCK TABLES (PUBTAB, PUB2, PUBOWNER)

PUB2 Relationship



Physical Unit Block Table 2 (PUB2)

Bytes			
Dec	Hex	Label	Description
0 - 3	0 - 3	P2USAGE	Usage count (number of non-ERP SIO)
4	4	P2FLAGS	Flag byte common to all PUB2 entries
		P2INTSM	X'80' Device is in intensive mode
		P2DIAGM	40 Device is in diagnostic mode
		P2NORCM	20 No recording mode
		P2STAT2	10 Call statistics transient 2
		P2NAMEF	08 Use PUB2 name completion field
		P2OPEN	04 Volume opened on this device
			02 Reserved
			01 Reserved
5	5	P2LIMIT	CE mode limit byte
		P2BBMASK	CE mode byte/bit mask
6	6	PUB2EXT	End of basic PUB2

Physical Unit Block Table 2 Extensions

Unit Record and Unsupported Device Extension

Bytes			
Dec	Hex	Label	Description
6	6	P2UNITX	Start of unit record PUB2
6 - 11	6 - B	SDRUNITR	SDR counters for unit record devices
12	C	P2UNITE	End of unit record PUB2

3540 Diskette Extension

Bytes			
Dec	Hex	Label	Description
6	6	P23540X	Start of PUB2 extension
6 - 13	6 - D	SDR3540	SDR counters
14 - 15	E - F	P23540R	Reserved
16	10	P23540E	End of 3540 PUB2

3211 Printer Extension

Bytes			
Dec	Hex	Label	Description
6	6	P23211X	Start of PUB2 extension
6 - 11	6 - B	SDR3211	SDR counter area
12	C	P23211E	End of 3211 PUB2

3800 Printer Extension

Bytes			
Dec	Hex	Label	Description
6	6	P23800X	Start of PUB2 extension
6	6	PB2SDR1	Channel data checks
7	7	PB2SDR2	Cont forms stacker misfolds
8	8	PB2SDR3	Burster/trimmer jams
9	9	PB2SDR4	No burst check
10	A	PB2SDR5	Burster/stacker jams
11	B	PB2SDRE	End of counters area
11	B	PB2DFLG	Default flags
		PB2DBRST	X'80' Default spec.=burst
12 - 15	C - F		Reserved
16 - 19	10 - 13	PB2DFCB	Default fcb id
20 - 23	14 - 17	PB2DCHAR	Default char. arrangement table id
24 - 27	18 - 1B	PB2DMDFY	Default copy modific. id
28 - 31	1C - 1F	PB2DFLSH	Default forms overlay frame id
32 - 35	20 - 23	PB2DFORM	Default paper forms id
36	24	PB2DFTE	End of default area
36 - 39	24 - 27	PB2WCGMS	Character sets presently load
40	28	PB2WMOD	WCGM# with modified character sets
		PB2WMOD0	X'80' WCGM0 contains a modified character set
		PB2WMOD1	40 WCGM1 cont. a mod.chr set
		PB2WMOD2	20 WCGM2 cont. a mod.chr set
		PB2WMOD3	10 WCGM3 cont. a mod.chr set
41	29	PB2FLAG1	First byte of flags
		PB2BURY	X'30' Burst = Y last specified
		PB2BURN	10 Burst = N last specified
		PB2UDCHK	08 DCHK=U was specified

3800 Printer Extension (cont.)

Bytes Dec	Bytes Hex	Label	Description
42	2A	PB2FLAG2	Second byte of flags
		PB2TRCY	X'30' TRC=Y was specified
		PB2TRCN	10 TRC=N was specified
		PB2DEBTR	0E Debug = trac last specified
		PB2DEBDU	0A Debug = dump last specified
		PB2DEBTE	06 Debug = term last specified
		PB2DEBNO	02 Debug = none last specified
43	2B		Reserved
44 - 47	2C - 2F	PB2FCB	Currently loaded FCB id
48 - 63	30 - 3F	PB2CHAR	Character arrangement tables (CAT)
48 - 51	30 - 33	PB2CHAR1	Id of 1st CAT currently loaded
52 - 55	34 - 37	PB2CHAR2	Id of 2nd CAT currently loaded
56 - 59	38 - 3B	PB2CHAR3	Id of 3rd CAT currently loaded
60 - 63	3C - 3F	PB2CHAR4	Id of 4th CAT currently loaded
64 - 67	40 - 43	PB2CMCHR	Id of CAT used when loading current copymode
68 - 71	44 - 47	PB2CPMOD	Id of copymode currently loaded into the printer
72 - 75	48 - 4B	PB2FORMS	Id of paper form currently loaded
76 - 79	4C - 4F	PB2FLASH	Id of current forms overlay frame
80 - 87	50 - 57	PB2COPYG	Eight copy group count last received by setprint
88	58	PB2CINDX	Copy group id (received by setprint)
89	59	PB2FLSHC	Flash count last received by setprint
90 - 91	5A - 5B		Reserved
92	5C	P23800E	End of 3800 PUB2

3886 Optical Character Reader Extension

Bytes Dec	Bytes Hex	Label	Description
6 - 25	6 - 19	SDR3886	SDR counter area
26	1A	P23886E	End of 3886 PUB2

3890 Document Reader Extension

Bytes Dec	Bytes Hex	Label	Description
6	6	P23890X	Start of PUB2 extension
6 - 15	6 - F	SDR3890	SDR counter area
16	10	P23890E	End of 3890 PUB2

Disk Device Extension

Bytes Dec	Bytes Hex	Label	Description
6	6	P2DISKX	Start of PUB2 extension
6	6	P2DFLG	Disk flags
		P2SDERRQ	X'80' Soft DASD error is queued
		P2DLOG	40 ERP requests error logged
7 - 8	7 - 8		Reserved
9	9	P2DMOD	Physical module identifier
10 - 15	A - F	P2DVOL	Volume serial number
16	10	P23330E	End of 3330 PUB2
16	10	P23340E	End of 3340 PUB2
16	10	P23350E	End of 3350 PUB2
16	10	P2fBAE	End of FBA PUB2
16 - 23	10 - 17	SDRDISK	SDR counters for 23xx
24	18	P2DISKE	End of 23xx PUB2

Tape Device Extension

Bytes			
Dec	Hex	Label	Description
6	6	P2TAPEX	Start of PUB2 extension
6 - 7	6 - 7	P2TNAME	Name of ERP that wants control
8	8	P2TFLG1	Tape flags 1
		P2TUNSOL	40 Unsolicited interrupt for tapes
		P2TERP	20 ERP is in control
		P2TREPO	10 ERP requests repositioning
		P2TIEORG	08 Use original tie byte; if off the opposite tie is used
		P2TECPT	04 Intercept next SIO request
		P2TROR	02 ERP read opposite request
		P2TREST	01 Restart users CCW chain
9	9	P2TFLG2	Tape flags 2
10	A	P2TFLG3	Tape flags 3
11	B	P2TEMPR	Temporary read count
12	C	P2TEMPW	Temporary write count
13	D	P2NOISE	Noise record count
14 - 15	E - F	P2ERG	Erase gap count
16 - 17	10 - 11	P2CLEAN	Cleaner action counts
18	12	P2PRD	Permanent read errors
19	13	P2PWT	Permanent write errors
20	14	P2ORGTIE	Tie original direction
21	15	P2OPPTIE	Tie opposite direction
22	16	P2ECTR0	ERP counter
23	17	P2ECTR1	ERP counter
24 - 31	18 - 1F	P2TWORKA	ERP work area
32 - 37	20 - 25	P2TVOL	Tape serial number
38 - 39	26 - 27	P2TBLK	Block length
40 - 43	28 - 2B	P2CCWAD	
44	2C	P2CSWRES	
45 - 47	2D - 2F		Reserved
48 - 63	30 - 3F	P2RUNSAV	Save area for run ERP
48 - 55	30 - 37	P2TSCSW	For CSW in error
56 - 57	38 - 39	P2TSSNS0	For sense bytes 0,1
58	3A	P2TSSNS5	For sense byte 5
59 - 63	3B - 3F		Reserved
.... 2400 Extension........			
64 - 73	40 - 49	SDR2400	2400 SDR area
74 - 75	4A - 4B		Reserved
4C	4C	P22400E	End of 2400 PUB2
.... 3420 Extension........			
64 - 83	40 - 53	SDR3420	3420 tape drive counter area
84	54	P23420E	End of 3420 PUB2

Physical Unit Block Ownership Table (PUBOWNER) Entry

Bits	Description
0	Device is owned by ACF/VTAM
1–2	Reserved
3	Device is owned by the system (e.g. contains PDS extent)
4–15	Identifier of Partition owning the PUB

Bit setting	Partition owning the PUB if number of partitions is										
	2	3	4	5	6	7	8	9	10	11	12
X'000'	UNASSIGNED										
001	BG	BG	BG	BG	BG	BG	BG	BG	BG	BG	BG
002	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB
004		F1	F2	F3	F4	F5	F6	F7	F8	F9	FA
008			F1	F2	F3	F4	F5	F6	F7	F8	F9
010				F1	F2	F3	F4	F5	F6	F7	F8
020					F1	F2	F3	F4	F5	F6	F7
040						F1	F2	F3	F4	F5	F6
080							F1	F2	F3	F4	F5
100								F1	F2	F3	F4
200									F1	F2	F3
400										F1	F2
800											F1

Bytes 120 - 123 (X'78'-X'7B') of the System Communication Region (SYSCOM) contain the address of the PUB Ownership Table. Label PUBOWNER identifies the first byte of the table.

Physical Unit Block Table (PUBTAB)

Bytes			
Dec	Hex	Label	Description
0	0	PUBCHANN	Channel number of device (Hex 0-F) X'FF' indicates end of PUBTAB
1	1	PUBDEVNO	Unit number
2	2	PUBCHQPT	Index to first CHANQ entry X'FF' indicates no request enqueued
3	3		Reserved
4	4	PUBDEVTY	Device type code
5	5	PUBOPTN	For TAPE devices: Tape Mode from ADD or ASSGN
			For Disk devices: Index of TRKHLD Table entry or X6
			For MICR devices: External line in use
			For 3704/3705: Type of channel adapter
			For 2560 or 5424/5425:
			X'80' Repositioning required (ERP)
			40 SYSPCH temporarily assigned to hopper 2
			20 SYSIPT temporarily assigned to hopper 2
			10 SYSRDR temporarily assigned to hopper 2
			08 Reserved
			04 SYSPCH permanently assigned to hopper 2
			02 SYSIPT permanently assigned to hopper 2
			01 SYSRDR permanently assigned to hopper 2
			For 3800:
			Bit 0-1
			00 3800
			01 3800 B
			10 3800 C
			11 3800 BC
6	6	PUBCSFLG	Channel Scheduler flags
		DEVBSY	X'80' Device is active
		SWITCH	40 Device is switchable
			20 Reserved
		QEDERR	10 I/O error queued for recovery
		OPINTV	08 Operator intervention required
		INTPEND	04 Interrupt was trapped by SDAID
		BRSDEV	02 Burst or overrunable device
		SVNTRK	01 7-track tape unit
7	7	PUBJCFLG	Job Control flags
			Bits 0-4: TAPE : Standard MODE assignment
			Not TAPE : All ones if device is up
			Device DOWN: All zeros
			5: Device supports RPS
			6: Primary path is not operational
			7: Secondary path is not operational

Note: A PUB entry must be added during IPL for any device of the installation.

Bytes 64-65 (X'40'-X'41') of the Partition COMREG contain the address of the PUB table. Label PUBTAB identifies the first byte of the table.

Channel Queue Table (CHANQ)

Bytes 37-39 (X'25'-X'27') of the System Communication Region (SYSCOM) contain the address of the Channel Queue Table. Label CHANQ identifies the first byte of the table.

Bytes Dec	Bytes Hex	Label	Description
0	0	CHQCHAIN	Index of next entry in free list or device queue. X'FF' indicates the last entry.
0–3	0–3	CHQCCBAD	Address of CCB/IORB associated with I/O request
4	4	REQID	PIK of service owner
5	5	CHQPROC	Logical processing flag required
		CHQDOINT	X'80' Interrupt not yet processed
		CHQDQUNC	40 Dequeue unconditional
		CHQNODEQ	20 Do not dequeue entry
		CHQPRCBF	10 Console buffering request
		CHQPROCF	08 OCCF request
		CHQDASFP	04 DASD file protect needed
		CHQFILE	02 SYSFIL on CKD device
		CHQSFFBA	01 SYSFIL on FBA device
6	6	CHQSLUB	System logical unit number associated with request X'FF' if this is a programmer unit (SYS000–SYS254)
7	7	TKREQID	Task ID (TID) of request owner
8	8	CHQCCSIO	SIO flag byte
		CHQCCACT	X'80' Device is running
		CHQCCALT	40 Alternate channel I/O
		CHQCCPRI	20 Primary channel I/O
		CHQCCLTE	10 Long time entry (Missing Interrupt Handler)
		CHQCCRUN	08 Condition Code 0
		CHQCCCSW	04 Condition Code 1
		CHQCCBSY	02 Condition Code 2
		CHQCCNOP	01 Condition Code 3
9	9	CHQCCBB1	Copied from byte 2 of CCB/IORB
10	A	CHQCCBB2	Copied from byte 3 of CCB/IORB
11	B	CHQCCBB3	Copied from byte 12 of CCB/IORB
12	C	CHQPFIX	Reserved for page fixing routine.
13–15	D–F	CHQPFIXL	Address of user specified or internal fixlist
16	10	CHQERRCT	Error retry count
17–18	11–12		Reserved
19	13	CHQPUBNO	PUB entry number
20	14	CHQFLG1	Flag byte
		CHQHQU	X'80' Unconditional request
		CHQHQA	40 Head queue request
		CHQCSBSY	20 Device busy status from PUB
		CHQCSQED	10 Device queued–in–error from PUB
		CHQDIDJA	08 Request was already accounted
			04 Reserved
		CHQFSIO2	02 Start on alternate channel only
		CHQFSIO1	01 Start on primary channel only
21	15	CHQGRP	Requestor flag
		CHQGROLT	X'80' OLTEP request
		CHQGRBTM	40 BTAM request
		CHQGRVTM	20 VTAM request (new interface)
			10 Reserved
		CHQGRRAS	08 RAS request
		CHQGRROK	04 Successful retry
			02 Reserved
			01 Reserved

Channel Queue Table (cont...)

Bytes Dec	Hex	Label	Description
22	16	CHQDEV	Device group indicator
		CHQDASD	X'80' CKD device or diskette
		CHQFBA	40 FBA device
		CHQTAPE	20 TAPE device
		CHQTP	10 TP (teleprocessing) device
		CHQCRT	08 2260 or 3277 device
		CHQURC	04 Unit record device
			02 Reserved
			01 Reserved
23	17	CHQIOINF	Delayed interrupt exit indicator
			X'00' Dispatcher (DISP)
			04 I/O initiator (INITRG)
			08 I/O interrupt handler (INTRTN)
			0C Error ignore routine (IGNORE)
			10 Cancel with code X'1A' (ERR1A)
			14 Reserved
			18 Dequeue routine (DEQUNCON)
			1C Post routine (PSTRESET)
			20 Emergency MSG writer (EMWINTRQ)
24	18	CHQCAWKY	Storage protect key
24-31	18-1F	CHQCSW	Accumulated status information from CSW

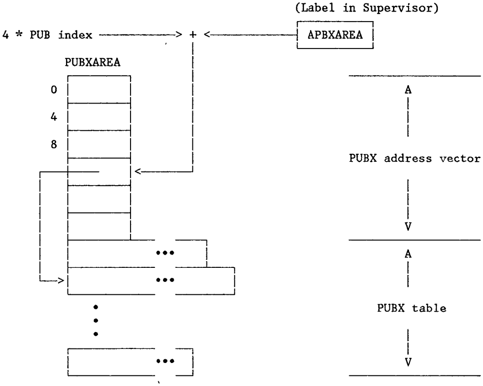
Channel Control Table (CHNTAB)

Label CHNTAB identifies the first byte of the Channel Control Table.

Bytes Dec	Hex	Label	Description
0	0	CHNTYPE	Channel Flag Byte
		NTOPCHN	X'80' Channel not operational or not present
			40 Reserved
		BLCKCHN	20 Block multiplexer channel
		MPXCHN	10 Byte multiplexer channel
			08 -04 Reserved
		BRSTCHN	02 Byte multiplexer running in burst mode
		BMPXCHN	01 Byte multiplexer with burst devices attached
1	1	CHNTERR	Number of unit checks pending on this channel
2	2	CHNTFLG1	Processing Flag Byte
		CHNRSTRT	X'80' Channel must be restarted
		CHNRSDEV	40 At least one device busy during restart
			20 -10 Reserved
		CHNISBSY	08 Channel is busy
			04 -01 Reserved
3	3	CHNTFLG2	Channel ID (Channel No.)
4-7	4-7	CHNTPUBF	Address of first PUB on channel
8-11	8-B	CHNTPUBL	Address of next PUB to be started on channel
12	C	CHNTCNT	Channel Balance Count
13-15	D-F	CHNTPUBB	Address of PUB that needs channel exclusively

Physical Unit Block Extension (PUBX)

A PUBX entry is addressed via address table APBXAREA at offset 4*PUB index (see Figure below).



PUBX (Physical Unit Block Extension)

Bytes			
Dec	Hex	Label	Description
0	0	PBXFLG	Flag byte
		PBXDASD	X'80' DASD device
		PBXTAPE	40 Tape device
		PBXUR	20 Unit record device
			10 - 02 Reserved
		PBXSLOG	01 SYSLOG device
1	1	PBXFLAG1	Flag byte
		PBXSHR	X'80' Partition sharable device Is on for DASD devices, for the SYSLOG device and for unit record devices, which are sharable as POWER dummy devices
		PBXMTFLG	40 Mount request pending
			20 - 01 Reserved
2- 3	2- 3	PBXCUU	CUU address
4	4	PBXPUBCD	VSE device type code
5-11	5- B	PBXSNSID	Sense device type information
5	5		X'FF' If entry is valid
6- 7	6- 7	PBXCUTYP	Control unit type number
8	8	PBXCUMOD	Control unit model number
9-10	9- A	PBXDVTYP	Device type number
11	B	PBXDVMOD	Device type model number
12-13	C- D	PBXOWNER	PIK of partition owning the device, if applicable
14-15	E- F		Reserved
			(if PBXSHR OFF)
16-19	10-13	PBXUSCNT	Device usage counters
20-23	14-17	PBXJACNT	Job Accounting SIO counters
			(if PBXSHR ON)
16-19	10-13	PBXUSOFF	Offset of usage counters within partition string
20-23	14-17	PBXJAOFF	Offset of SIO counters within partition string
24-27	18-1B	PBXERBLK	Addr. of Error Entry for this device
28	1C	PBXCLNG	End of common section
		End of section	common to all devices
28-31	1C-1F	PBXCCW	DASD devices: Address of Set File Mask CCWs TAPE devices: Addr. of Set Mode CCWs
32	20	PBXTLNG	End of tape device section
32	20	PBXDLNG	End of DASD device section

ERBLOC Area

Bytes Dec	Bytes Hex	Label	Description
0–7	0–7	SVC5NM	Name of first/next ERP Transient to be fetched
8–11	8–B	YRETRY	Continuation address for retry I/O request (INITRG)
12–15	C–F	YIGNORE	Continuation address to ignore I/O error (IGNORE)
16–19	10–13	ACANCEL	Continuation address to cancel I/O request (ERR1A)
20–23	13–17	YERPEXIT	Common DSK/ERP return address (ERPEXIT)
24–75	18–4B	ERQ1	Area to pass recovery and recording information to the ERP. Its layout is the same as for a single error block, except for the 8–byte header (See note)
76–111	4C–6F	SNSSDAID	Sense data saved by SDAID
112–119	70–77	ERCHNOFT	Chain header offset table, used to address the following error chains
120–123	78–7B	RASERCHN	Address of first RAS error entry
124–127	7C–7F		Pointer to RAS TIB
128–131	80–83	ERPERCHN	Address of first ERP error entry
132–135	84–87		Pointer to ERP TIB
136–139	88–8B	DSKERCHN	Address of first DSK error entry
140–143	8C–8F		Pointer to DSK TIB
144–147	90–93	SNSERCHN	Address of first SNS error entry
148–151	94–97		Pointer to SNS TIB

Bytes 0-3 (X'00 - X'03') of the System Communication Region contain a pointer to the ERBLOC area.

There is one I/O error block for each device. Field PBXERBLK in the PUBX contains a pointer to this block.

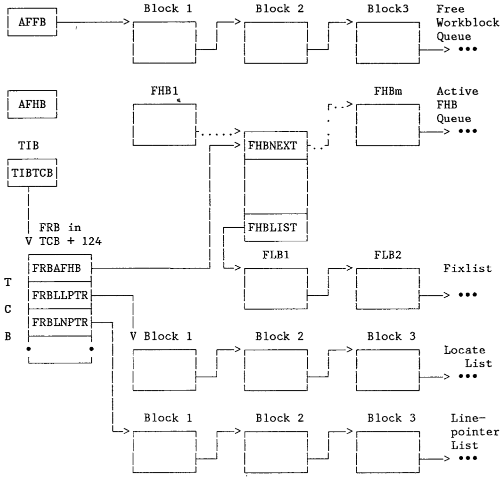
An additional error block exists for some system tasks. The address of this block is contained in field TCBERBLK of the system task TCB.

Bytes			
Dec	Hex	Label	Description
0-3	0-3	ERBLKPTR	Pointer to next error block or 0
4	4	ERBLKFLG	Flag byte
		HQERBLK	X'80' System task error block
		ALTCHANN	40 Error on alternate channel
		ERSNSDAV	20 Sense data available
		ERACTIVE	10 Error block active
		ERQUEUED	08 Error block is enqueued in some error chain
			04 - 01 Reserved
5	5	ERBLKFLG1	Flag byte
			X'80 - 10' Reserved
		NEEDSNS	08 Must be processed by SNS task
		NEEDDSK	04 Must be processed by DSK task
		NEEDERP	02 Must be processed by ERP task
		NEEDRAS	01 Must be processed by RAS task
6	6		Reserved
7	7	ERBLKSNL	Number of sense bytes
••• End of error block header ••••••• Layout of UNIT CHECK entry •••			
8-15	8-F	ERRQCSW	CSW of I/O error
16-17	10-11	ERRQPUB	PUB pointer of affected device
18	12	ERRQFLG	Flag Byte
		TRUNRF	X'80' No record found on DASD
			40 Interv. required (set by ERP)
			20 Pass back error information (set by ERP)
		IGNERR	10 Channel program is not retryable (IGNORE)
		SUCCESS	08 Error successfully recovered (IGNORE)
		RTYERR	04 Channel program is retryable
		RECONLY	02 Error to be recorded only
		OCCUP	01 Error block is in use (set only for error block in ERBLOC area)
19	13	ERRQMSG	Message Code
20-23	14-17	ERRQSEK	Used for disk devices only
			CKD: Failing Seek address
			FBA: OS device type codes
24	18	ERRQCQPT	Index of channel queue entry
			X'FF' for unsolicited error
24-27	18-1B	ERRQCCB	CCB pointer (addr. = 0 if not avail.)
28-...	1C-...	ERRQSNS	Sense data
••• End of UNIT CHECK entry •••••••••• Layout of RECORDING entry •••			
8-11	8-B	ERQAEADR	SD record address
		ERQAEINF	SD record information
12	C	ERQAELEN	Length of SD record
13	D	ERQAETYP	Type of SD record
14	E	ERQAESW1	Record dependent switch 1
15	F	ERQAESW2	Record dependent switch 2
16-17	10-11	ERQAEPUB	PUB pointer of affected device
18	12	ERQAEFLG	Flag Byte
			X'80' SD record is TFIX-ed
			02 Must be 0 for recording info.
		OCCUP	01 Error block is in use (set only for error block in ERBLOC area)
19	13	ERQAEMSG	Contains X'AE' for Alternate Entry
20-23	14-17	ERQAETIB	TIB of requesting task
24-27	18-1B		Reserved
28-...	1C-...	ERQAECOM	Communication information
••• End of RECORDING entry •••••• Layout for CHANNEL CHECK entry •••			
8-31	8-1F		See Note.

Note:
Bytes 8-31 same as ERPIB control block.

Relationship of Control and Work Blocks for CCW fixing(VSE:EXCP)

Figure below shows the relationship of the control and work blocks for the CCW fixing function.



Fix Request Block

The Fix Request Block (FRB) serves as a dynamic save area and work area. It is located in the TCB work area. (CCW fixing part) Since a fixing request may be interrupted (for example by a page fault, wait), the fixing routine has to be partially re-enterable to enable the handling of several requests simultaneously.

Fixlist Header Blocks (FHB)

0	Flag Byte 1 (General Fixing Function)	Flag Byte 2 (Reserved)	TID of Requester
4	Pointer to next active FHB		
8	BA1 *	EA1 *	
12	BA2	EA2	
16	BA3	EA3	
20	BA4	EA4	
24	BA5	EA5	
28	BA6	EA6	
32	Pointer to next fixlist block		

Note:
Bytes 120-123 (X'78-7B') (Label FRBAFHB) in the CCW fixing work area of the Task Control Block (TCB) contain the address of the Fixlist Header Block (FHB).

Layout of Fixlist Header Block (FHB) for General Fixing Function

0	Flag Byte 1 (General Fixing Function)	Flag Byte 2 (Fast Fixing Support)	TID of Requester
4	Saved queue forward pointer		
8	Saved queue backward pointer		
12	Pointer to replica or zero		
16	Pointer to next active FHB		
20	BA1 *	EA1 *	
24	BA2	EA2	
28	BA3	EA3	
32	Pointer to next fixlist block		

* BA = Page number multiplied by 8 of first page to be TFIXED.

* EA = Page number multiplied by 8 of last page to br TFIXED.

Layout of Fixlist Header Block (FHB) for Fast Fixing Support

The meaning of the flag bytes is as follows:

Flag Byte 1 (General Fixing Function):

```
  Bit 0=1: Fixing function request complete.
  Bit 1=1: At least one page is fixed for
           this task or the fixing request
           is pending.
  Bit 2=1: Fixing of pages required.
  Bit 3-7: Reserved.
```

Flag Byte 2 (Fast Fixing Support):

```
  Bit 0=1: Fast fixing in progress.
  Bit 1=1: FHB belongs to saved FHB queue.
  Bit 2=1: IORB request.
  Bit 3-7: Reserved.
```

Fixlist Block (FLB)

0	BA1	EA1
4	BA2	EA2
8	BA3	EA3
12	BA4	EA4
16	BA5	EA5
20	BA6	EA6
24	BA7	EA7
28	BA8	EA8
32	Pointer to next fixlist block or zero	

Layout of Fixlist Block (FLB)

Note: Bytes 32 - 35 (X'20 - 23') of the Fixlist Header Block (FHB) contain the address of the Fixlist Block (FLB).

LTID (Logical Transient Owner)

The LTID, a halfword (LIK) at displacement 88 in the system communication region (SYSCOM) contains the same value as the TID when the Logical Transient Area (LTA) is in use and, therefore, identifies the owner of the LTA. When the LTA is free, the LTID is zero. The SVC 2 (X'02') routine sets the LTID, and the SVC 11 (X'0B') routine resets it to zero.

Notes:

1. Do not use this interface any more.
2. Any logical transient routine may find its own task identifier by using the TID field.

Logical Transient Key (LTK)

The logical transient key, a halfword (LTK) at displacement 110 in each partition communication region (COMREG), has a zero value in the high-order byte and a key value in the low-order byte. In a foreground communication, the key value in the LTK is not significant. The LTK in the background communication region (BG-COMREG) has the same value as the PIK of the partition of the task that owns the LTA, or contains zeros when the LTA is free. When the LTA is occupied by the task, therefore, the BGCOMREG has the same value in its LTK as in its PIK when the owning task is active.

Note: This LTK interface should not be used anymore.

I/O Requestor's Partition or System Task ID (REQID)

The REQID is a one-byte identifier in the Channel Queue (CHANQ) entry, used for storing the service owner identification. When a background or foreground program requested the I/O operation, the REQID has the value of the partition identification key. When a system task requested the I/O operation, the REQID contains the partition identification key of the service owner. The REQID is set by the Channel Scheduler Routine.

Locate List Block

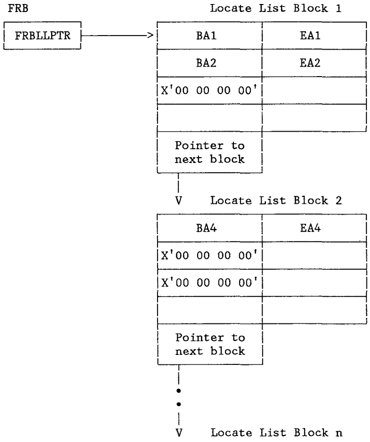
A locate list is built during the scanning of the channel program of an EXCP request that uses a CCB. Each locate list block consists of four 8-byte entries, and an entry contains a begin (BA) and an end address (EA) which define an area of the channel program whose CCWs have been checked already.

After completion of the scanning procedure the locate list defines those areas of a channel program that have to be TFIXed. The entries describe isolated areas; they are not adjacent or overlapping, and are arranged in ascending sequence.

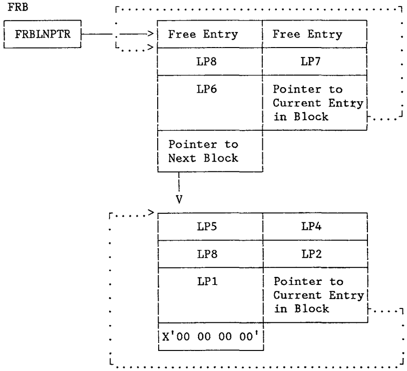
Line Pointer Block

Line pointer blocks are used for storing addresses of channel program areas during the scanning procedure.
The line pointer blocks ensure that all lines of the channel program will be checked for fixing.

Layout of Locate List Blocks



Layout of Line Pointer Blocks

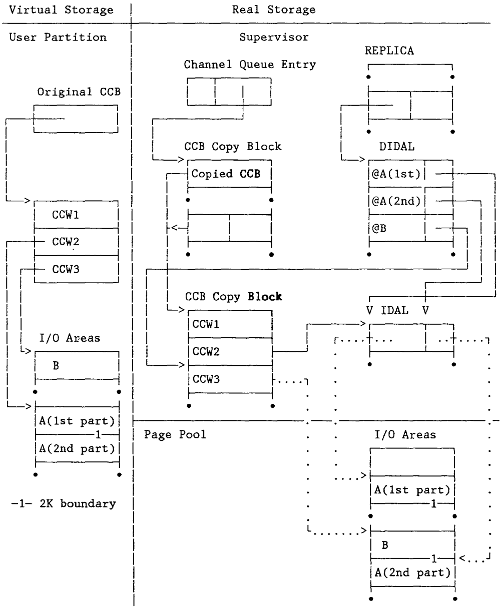


Translation Control and Copy Blocks

The following control and copy blocks are used to copy and translate a CCB and channel program for a virtual-mode I/O request:

- A CCB copy block. The user CCB and sense CCW (if any) are copied into this block. The CCB copy block also contains information about the copied and translated channel program.
- A translation control block (CCWTCB). This is a work and save area, located in the CCW translation work area of the task control block (TCB) and used during translation.
- CCW copy blocks. Each block contains copy locations for up to 7 contiguous CCWs and queueing information.
- IDAL blocks used for building Indirect Data Address Lists for data areas which cross page boundaries.
- Fix information blocks containing the page frame numbers of pages freed for this request.

Control Block Structure for Fast CCW Translation (/370 Mode)



CCB Copy Block

	0	1	2	3	4	5	6	7	
0	CCBCNT		CCB COM1	CCB COM2	CCB STA1	CCB STA2	CCB CLS*	CCB LNO	A Copied CCB
8	CCBCCW Address of first CCW				CCBBY3	CCBCSWW			| V
16	CCBSENS Sense CCW if any								
24	TID TASKID		CCB Flag**	Unused	CCBVA Virtual address of CCB				
32	CCBACB Address of first CCW copy block in channel program with lowest VBA				CCBICB Address of first IDAL block in channel program				
40	CCBXINF (Fix information) Real page numbers of TFIXed pages								
64	CCBXPTR Address of additional fix information block				*** X'80'	CCBNEXT Address of next CCB copy block			

- * - Bit 2 is set (X'20') to indicate copied CCB
- ** - Legend CCBFLAG:

Bits Description

0: Indicates that CCW-translation of this request is complete; indicator is set before I/O request is enqueued in channel queue.

1: Indicates that control has been transferred to TFIX routine at least once during CCW translation; if 0, scan through CCBXINF for freeing pages is skipped; indicator is set immediately before control is passed to TFIX routine.

2: Reserved.

3: Indicates that the next CCW translation request from BTAM is from BTAM channel appendage. This indicator is set immediately after the first time a request from BTAM has been completed.

4: Indicates that the channel program is valid for fast CCW translation (CCWs are contiguous, the requestor is not BTAM and it is not a system task request with an I/O area in the SVA).

5: Indicates that this CCB copy block is on the saved CCB queue.

6: Indicates that the pages containing I/O areas for this channel program require fixing.

7: Reserved.

- *** - 'Block in use' indicator.

Note: If the fast CCW translation option is active, bytes 56-67 of the CCB copy block have a different meaning, as shown below:

Bytes	Description
56 - 59:	The address of the REPLICA block associated with this channel program.
60 - 63:	Pointer to the next CCB in the saved CCB queue used by the fast CCW translation routines.
64 - 67:	Pointer to the saved CCB queue.

The fix information normally held in these locations is not required when fast CCW translation is active, as it is kept in the REPLICA for the channel program.

Translation Control Block (CCWTCB)

CCWTCB is a work and save area, located in the task control block (TCB) and used during translation. The format of the CCWTCB is shown in the "CCW translating work area (/370 mode)" part of the TCB.

CCW Copy Block

	0	1	2	3	4	5	6	7
0	1st Copy location for CCW							
8	2nd Copy location for CCW							
16	3rd Copy location for CCW							
24	4th Copy location for CCW							
32	5th Copy location for CCW							
40	6th Copy location for CCW							
48	7th Copy location for CCW							
56	X'80' *	X'000000'			Virtual address of first CCW in copy block (VBA)			
64	X'88' **	X'000000'			*** X'80'	Addr. of next CCW copy block in chain (ANB)		

Notes:

1. * X'80' indicates the end of the CCW copy locations in the block. It is replaced by a TIC (Transfer in channel command) if the 7th copy location contains a copied CCW with data- or command chaining. Bytes 57-59 will then point to the copy location of the CCW following the CCW in the 7th copy location. Bytes 56-59 will not be changed if the CCW in the 7th copy location is a TIC.

2. ** X'88' indicates the last 8-byte entry in the block. It is replaced by a TIC if the CCW in the 7th copy location is a status modifier CCW. Bytes 65-67 will then point to the copy location of the second CCW following the status modifier CCW.

   The CCW copy blocks for a translation are queued in order of increasing VBAs with the lowest one being pointed to by the field CCBACB in the CCB copy block.

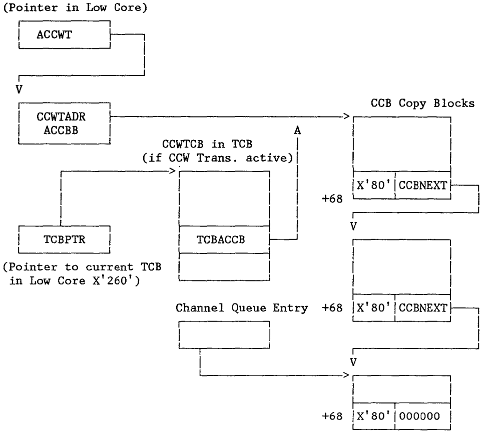
3. *** X'80' 'Copy block in use' indicator.

IDAL Blocks

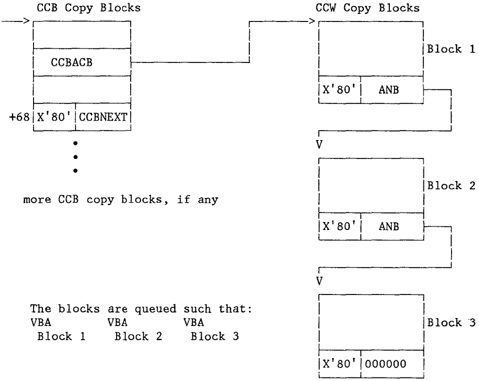
CCWs whose data areas cross 2K boundaries must have an IDAL (Indirect Data Address List) in the copied channel program.

If a data area crosses a 2K boundary, the CCW is changed to show that an IDAL is used (bit 37 of the copied CCW is set) and the address of the IDAL is placed in the data address of the CCW.

Locating CCB Copy Blocks

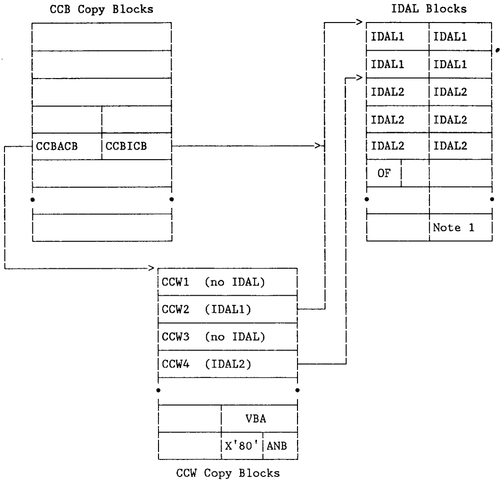


Locating CCW Copy Blocks



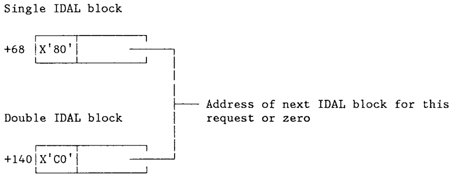
The blocks are queued such that:
VBA Block 1 VBA Block 2 VBA Block 3

Relation of IDAL Blocks to other Blocks



Notes:

1.

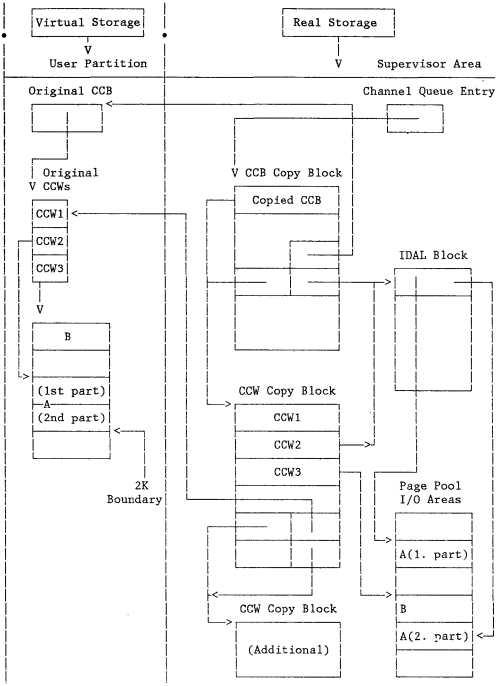


the contents of X'C0' being:
X'80' Block in use
X'40' Double copy block

2. The X'0F' in the first byte of the 11th IDAW indicates the end of the IDAWs for the block. In this case, the IDALCNT field in the CCWTCB would show seven free copy locations.

3. The data area of CCW2 crosses three 2K boundaries (may be up to 8K) and the data area of CCW4 crosses five 2K boundaries (may be up to 12K).

Schematic Representation of Channel Program Translation



Doubleword Indirect Data List (DIDAL)

DIDAL Block

ENTRY 1	
ENTRY 2	
ENTRY 3	
ENTRY 4	
ENTRY 5	
ENTRY 6	
ENTRY 7	
ENTRY 8	
Reserved	CHAIN POINTER **

DIDAL Entry

Virtual address	Flag byte*	Pointer to real loc. ***
0 3	4	5 7

- * Flag Byte

Bits	Description
0	Indicates that TFIXING is not necessary because the page has already been TFIXED for this request.
1–6	Reserved.
7	Indicates that TFIX request for this entry has been completed.

- ** Pointer to next DIDAL (X'80000000' in last DIDAL).
- *** Real location (copied ir IDA word) that should contain the translated I/O area address.

REPLICA Control Blocks

Main REPLICA Block

VCCBA	RCCBA		
TIMEST	REPPIK	REPLCNT	CCWSTRL
REPDIDAL			
CCB			
	CCW1		
CCW1	CCW2		
CCW2	CCW3		
CCW3	REPFPT		
REPBPT	X'80'	REPNEXT	

Additional REPLICA Block

CCW4		
CCW5		
{		{
REPBPT	X'80'	REPNEXT

VCCBA	Virtual CCB address.
RCCBA	Address of copied CCB.
TIMEST	Timestamp.
REPPIK	Partition identification key.
REPLCNT	The number of tasks currently testing this REPLICA for a match with their channel program.
CCWSTRL	Length of CCW string (number of CCWs).
REPDIDAL	Address of DIDAL block.
REPFPT	Forward pointer used for chaining REPLICAs.
REPBPT	Backward pointer, used for chaining REPLICAs.
REPNEXT	Pointer to (next) additional REPLICA block.

REPLICA Blocks

REPLICA blocks are needed to save the channel program and related information. The first two REPLICA blocks are REPLICA Header Blocks (RHB1+2).

Layout of REPLICA Header Blocks for a CCB

RHB1

0	(See Note) Flag Byte	Number of Tests on REPLICA	PIK of Requester
4	Forward pointer in partition's REPLICA queue		
8	Backward pointer in partition's REPLICA queue		
12	Address of virtual CCB		
16	Saved CCB		
32	Pointer to RHB2		

RHB2

0	Pointer to RHB1
4	Pointer to associated FHB (fixlist)
8	Time stamp
12	Length of saved channel program
16	Reserved
20	Reserved
24	Saved user SENSE
32	Pointer to next REPLICA block

REPLICA Block

0	CCW1
8	CCW2
16	CCW3
24	CCW4
32	Pointer to next block or zero

Note:
Bit 0=1 Freeing of REPLICA requested.
Bits 1-7 Reserved.

Layout of REPLICA Header Blocks for an IORB

RHB1

0	(See Note) Flag Byte	Number of Tests on REPLICA	PIK of Requester
4	Forward pointer in partition's IORB REPLICA queue		
8	Backward pointer in partition's IORB REPLICA queue		
12	Reserved		
16	First user fixlist entry		
24	Second user fixlist entry		
32	Pointer to RHB2		

RHB2

0	Pointer to RHB1
4	Pointer to associated FHB (fixlist)
8	Time stamp
12	Length of saved channel program
16	Third user fixlist entry
24	Fourth user fixlist entry
32	Pointer to next REPLICA block

REPLICA Block

0	Fifth user fixlist entry
8	Sixth user fixlist entry
16	Seventh user fixlist entry
24	Eighth user fixlist entry
32	Pointer to next block or zero

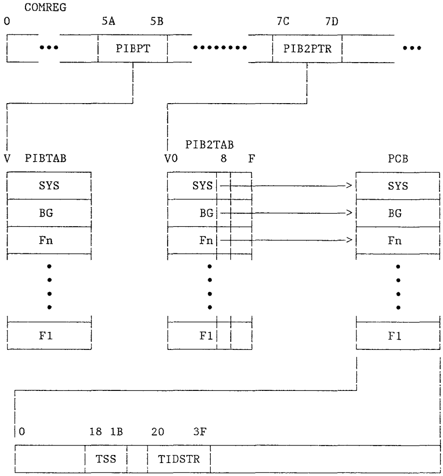
Notes:

- Bit 0=1 Freeing of REPLICA requested.
- Bits 1-7 Reserved.

PARTITION CONTROL BLOCKS (PIB, PIB2, PCB)

PIB (Partition Information Block)
PIB2 (Partition Information Block Extension)
PCB (Partition Control Block)

Partition Control Blocks Interrelationship



Notes:

1. Fn = Foreground partition "n"
2. @ = Address
3. @(FnPIB) = @(PIBTAB) + FnPIK
4. @(FnPIB2) = @(PIB2TAB) + FnPIK
5. @(@(FnPCB)) = @(FnPIB2) + 8

The PIB, the PIB2 and the PCB contain static and dynamic status information about the system and about partitions. There is one set of these control blocks for the system and one for each partition generated (NPARTS specification).

Partition Information Block (PIB)

Bytes			
Dec	Hex	Label	Description
0	0	PIBSTATE	Partition status byte
		STOPPED	X'82' Partition is stopped
		INACT	80 Partition is inactive/unbatched
			00 Partition is active
1	1	PIBOLDST	Maintask status at operator cancel time
2–3	2–3	PIBLOGID	SYSLOG ID (AR,BG,F1,...,Fn)
4	4	PIBFLAG0	Flag byte
		TRAM	X'80' Partition running in virt. mode
			40 Reserved
			20 Reserved
			10 Reserved
			08 Reserved
			04 Reserved
			02 Reserved
			01 Reserved
5–7	5–7	PIBPBEG	Begin address of virtual partition
8	8		Reserved
9–11	9–B	PIBSAV2	Problem Program save area of LTA owner (system PIB)
12	C	PIBFLAG1	Flag byte
			X'80' Reserved
		APPEN	40 Channel appendage allowed
			20 Reserved
			10 Reserved
			08 Foregr. assignments to be hold
			04 Reserved
			02 Reserved
			01 Reserved
13	D	PIBLUBID	Number of System LUBs (AR PIB: Number of BG system LUBs)
14	E	PIBLUBNO	Number of Progr.LUBs (AR PIB: Number of BG progr. LUBs)
15	F	PIBFLAG2	Flag byte
			X'80' Reserved
			40 Reserved
		JOBDUN	20 End of Job indicator
			10 Partition stopped (set by Job Control)
			08 Reserved
			04 Reserved
			02 Reserved
			01 Reserved

The address of PIBTAB can be found in bytes 90-91 (X'5A'-X'5B') of the Partition Communication Region.

The first entry of the table belongs to the system.

Partition Information Block Extension (PIB2)

Bytes Dec	 Hex	Label	Description
0-1	0-1	PIBCOMRA	Address of Communication Region (AR PIB: BG COMREG)
2-3	2-3	SYSLUBX	Index of System LUBs relative to LUB table (always 0)
4-5	4-5	PIBMTID	Task ID of main task
6-7	6-7		Reserved
8-11	8-B	APCB	Address of Partition Control Block (PCB)
12-13	C-D	PIBPIK	PIK of partition (0 for AR PIB2)
14-15	E-F		Bytes 2 and 3 of ECB for ATTACH limit within partition
15	F	PIBFLG3	Extension flags

The address of the PIB2TAB can be found in bytes 124-125 (X'7C'-'X7D') of the partition communication region (COMREG).

The first entry of the table belongs to the system.

The PIB/PIB2 for a given partition is found by adding the PIK of this partition to the begin address of the appropriate table.

Partition Control Block (PCB)

Bytes Dec	Bytes Hex	Label	Description
0-1	0-1	PCBLNGTH	Length of PCB
2	2	PCBFLAG	Flag byte
		BALANCED	X'80' Balanced partition
		PERACT	40 Reserved
		SUPPRPFH	20 Suspend page fault handling (load leveller)
			10 Reserved
			08 Reserved
			04 Reserved
			02 Reserved
		PWSRVFLG	01 Some task within partition waiting for POWER
3	3		Reserved
4-7	4-7	PCBPMASK	Partition priority mask
8-11	8-B	RUNTIME	Time counter for part.balancing and job accounting
12-15	C-F	PBALTIME	Initial value of part.balancing time
16-19	10-13	PCBJAPTR	PCB pointer for time accounting
20-21	14-15	PCBPIK	PIK of partition
22-23	16-17	PCBLCTSS	Active length code for TIDSTR
23	17	PCBSUBS	Number of attached subtasks
24-27	18-1B	TSS	Task selection bit string
28-31	1C-1F		Reserved
32-63	20-3F	TIDSTR	TID's of attached tasks in priority order
64-67	40-43		Reserved
68-69	44-45	PCBNTASK	Counter of used subtasks
70-71	46-47	CDLDTID	TID of CDLOAD owner within partition
71	47	CDLDBYTE	Significant portion of CDLDTID
72-75	48-4B	PCBPSCB	SCB pointer of allocation space
76-79	4C-4F	PCBASCB	SCB pointer of active space
....		Begin of SMCB	....
80-81	50-51	SMAXPFIX	PFIX limit in pages (system PCB: SVA PFIX limit)
82-83	52-53	SMPFIX	PFIX count in pages (system PCB: SVA PFIX count)
84-87	54-57	SMPSAVE	Partition PCB: Address of main task save area
		PCBAPBEG	= Active partition begin address
88	58	SMVFLAG	Storage management flag byte
			X'80' SETLIMIT given indicator
			40 Reserved
			20 Reserved
			10 Reserved
			08 Reserved
			04 Reserved
			02 Reserved
			01 Reserved
88-91	58-5B	SMVGVIS	Partition PCB: Addr. of GETVIS area
		SMSGVIS	System PCB: Address of system GETVIS area
92-95	5C-5F	SMVPBEG	Partition PCB: Begin of virtual partition
		SMSVABEG	System PCB: Begin of SVA
96-99	60-63	SMVPEND	Partition PCB: End of virtual partition + 1
		SMSVAEND	System PCB: End of SVA + 1
100-103	64-67	SMRPBEG	Partition PCB: Begin of real part. System PCB: Begin of real area for system PFIX
104-107	68-6B	SMRPEND	Partition PCB: End of real part. + 1 System PCB: End of real area for system PFIX + 1
(28)	(1C)	SMCBLNG	Length of SMCB
....		End of SMCB	....

Partition Control Block (PCB) (cont...)

Bytes			
Dec	Hex	Label	Description
108-111	6C-6F	PCBAPEND	End address + 1 of user key area within partition
112	70	PCBSSCNT	Subsystem counter within partition
113	71		Reserved
114	72	PCBSSFL1	Subsystem flag byte
			X'80' Reserved
			40 Reserved
			20 Reserved
			10 Reserved
			08 Reserved
			04 Reserved
			02 Reserved
		NPDA	01 NPDA partition
115	73	PCBSSFLG	Subsystem flag byte
		PWR	X'80' POWER partition
		VTAM	40 VTAM partition
		ICCF	20 ICCF partition
		CICS	10 CICS partition
		VCNA	08 VCNA partition
		OCCF	04 OCCF partition
		DS2	02 DS2 partition
		SSX	01 SSX partition
116-121	74-79	CHPTENT	Checkpoint PFIX entry
116-119	74-77	CHPTPAGE	First PFIXCHPT page not yet handled
120-121	78-79	CHPTCNT	Remaining number of PFIX-ed pages for PFIXCHPT
122-123	7A-7B		Reserved
124-127	7C-7F	PCBOCPTR	Address of OC exit routine (partition PCB only)
128-131	80-83	PCBOCSAV	Address of OC exit save area (partition PCB only)
132-139	84-8B	PCPUTIME	CPU time counter
140-147	8C-93	POVHTIME	Overhead time counter
148-155	94-9B	PBNDTIME	Allbound time counter
156-163	9C-A3	PCBRQ	Begin of PCB resource descriptors
		SRQGTV	GETVIS/FREEVIS resource queue header
162	A2	PCBRBGTV	GETVIS/FREEVIS resource byte
164-171	A4-AB	SRQCDL	CDLOAD resource queue header
170	AA	PCBRBCDL	CDLOAD resource byte
172-179	AC-B3	SRQPFX	PFIX resource queue header
178	B2	PCBRBPFX	PFIX resource byte
180-183	B4-B7	PCBCNT	Address of usage and SIO counters for partition sharable devices
184	B8	FIXTYPE	PFIX flag byte
		GTRBIT	X'80' GETREAL request
		RSTRTBIT	40 PFIXREST request
			20 Reserved
			10 Reserved
			08 Reserved
			04 Reserved
			02 Reserved
			01 Reserved
184-187	B8-BB	FIXTIB	TIB pointer of PFIX/GETREAL requestor
188-191	BC-BF	PFTERSVD	Address of reserved PFTE for PFIX/GETREAL
.... End of system PCB			
184-192	B8-C0	PCBPFIXL	Count of tasks with open VTAM ACBs
193-194	C1-C2		Reserved
195	C3	PCBVTCNT	Count of tasks with open VTAM ACBs
(196)	(C4)	PCBVMLNG	Length of PCB for VM
196-215	C4-D7	PHOTIB	Pseudo-TIB for PHO

Bytes			
Dec	Hex	Label	Description
0	0	TIBCHAIN	Wait chain indicator
			X'00' Task is enqueued in wait chain
			FF Last TIB in resource wait chain
1-3	1-3		Pointer to next TIB in a resource wait chain
4-7	4-7	TIBSTATE	Resource identifier within generic wait chain
8	8	TIBFLAG1	Flag byte
		PHOIND	X'80' Pseudo-TIB for PHO or VIO
		PHOACT	40 PHO initialized for this task
		PHOREQ	40 (PHO TIB only) PHO request enqueued
		EOTACT	20 EOT active
		VIOREQ	20 VIO pseudo-TIB (if PHOIND on)
		EOTINPR	10 EOT subsystem clean-up active
		LTAACT	08 LTA active
		LTAOWNER	04 LTA owner
		TERMACT	02 Terminator active
		SYSACT	01 System code active
		PRIVILEG	1B
8-11	8-B	TIBTCB	TCB pointer
		TIBPFAPP	PHO TIB: Address of PHO appendage
		TIBVIOTB	VIO TIB: VIOTAB pointer
12	C	PGQTYP	Type of page I/O request
		PGSEL	X'80' Page selection required
		PGNCNT	40 Counting already done
		PGO	10 Page-out request
		PGOWAIT	18 Page-out request with waiting task
		PGOPGIN	14 Page-out request with waiting Page-in
		PGOVIO	12 Asynchr. Page-out requ. for VIO
12-15	C-F	PGINF	Address of PDS device control block or of PFTE
16-17	10-11	TIBRTID	User tasks: Task ID
			Syst.tasks: Task ID of service owner
			PHO TIB: Task ID of PHO owner within partition
17	11	TIBRBYTE	Significant byte of TIBRTID
18-19	12-13	TIBRPIK	User tasks: PIK of owner partition
			Syst.Tasks: PIK of serviced partition
(20)	(14)	TIBPFLNG	Length of PHO/VIO TIB
.... End of Pseudo-TIB			
20-23	14-17	TIBPCB	Pointer to PCB of owner partition
24-27	18-1B	TIBPMASK	Priority mask of task within partition
28	1C	TIBFLAG	Dispatcher exit flags
		CSVRET	X'80' Return to supervisor routine
		RETRYSVC	40 Restart SVC pre-processing
		TIBDELMV	20 General delayed move processing
		FETCHEOJ	10 Task termination to be initialized
		ROLLOUT	08 ICCF inter.part. eligible for roll-out
		CDELEX	04 Delayed timer interrupt processing
		OCPEND	02 OC exit to be scheduled
		APSEXFLG	01 Call VTAM exit
48	30	TIBLNG	Length of TIB

Task Information Block (TIB) (cont...)

Bytes Dec	 Hex	Label	Description
29	1D	TIBFLAG2	Flag byte
		ICCFPP	X'80' ICCF Interactive Partition
		PWRMTASK	40 POWER main task
		OVHIND	20 Account CPU-time as overhead
		SVPCCNCL	10 Status saved in special save area
		OCCFACT	08 OCCF service request pending
		ASYOCACT	04 ASYNOC request pending
		VTOPEN	02 At least one VTAM ACB open
		LIBRSERA	01 Librarian service active
30	1E	TIBCNCL	First cancel code
31	1F	TIBCNCL2	Last cancel code
		TERMCNL	X'80' Terminator canceled
32	20	TIBRQID	Task status flag
33	21	TIBFLAG3	Flag byte
			X'80' Reserved
			40 Reserved
			20 Reserved
		SEIZEBIT	10 Task is seizing the system (see SVC-16)
			08 Reserved
			04 Reserved
			02 Reserved
			01 Reserved
34	22	TIBCNCL3	Terminator cancel code
35	23	TIBDMFLG	Del. move flag, used with TIBDELMV
		TIBCMVEX	X'80' Invoke CCB delayed posting
		TIBXPCEX	40 Invoke XPCC delayed move exit
		TIBSFLEX	20 Return to SYSFIL FBA processing
		TIBPERST	10 Invoke PER bit update
		TIBDMALL	F0 Invoke previous services
			08 Reserved
			04 Reserved
			02 Reserved
			01 Reserved
36–41	24–29	TIBITREQ	Significant part of timer interrupt
42–43	2A–2B	TIBITCHN	Address of Y-pointer to next TIB in IT chain
44–47	2C–2F	TIBSCB	Current SCB pointer for task
48	30	TIBLNG	Length of TIB

Task Control Block (TCB)

Bytes			
Dec	Hex	Label	Description
0-1	0-1	TCBLNGTH	Length of TCB
2	2	TCBAUTHF	Authorization flag
		TCBFLAG3	
			X'80' Reserved
			40 Reserved
		CICSMT	20 CICS 'maintask' from SUBSID
		DLIMT	10 DLI 'maintask' from SUBSID
		ISPFMT	08 ISPF 'maintask' from SUBSID
		FTPTSK	04 FTP task allowed to use CPCOM
			02 Reserved
			01 Reserved
3	3	TCBRID	RID saved on interrupt in supervisor service
4-5	4-5	FATHERID	Task ID of attaching task (user subtask only)
6	6	TCBFLAGS	Flag byte
		SYSRESW	X'80' DASD File Protect to be skipped
		SKIPMSG	40 OPTION=NODUMP for STXIT AB
		EARLYAB	20 OPTION=EARLY for STXIT AB
		ACLOSE	10 VSAM Automatic Close in progress
		VSMOPEN	08 VSAM ACB'S open in partition (set for main tasks and ICCF IP's)
			04 Reserved
		ICCFSVC	02 ICCF SVC screening flag
		OWNTIMER	01 Task Timer owner (main task only)
7	7	TCBFLAG2	Flag byte
		CNCLRTRN	X'80' Terminator to be reentered
			40 Reserved
			20 Reserved
		OPENSVA	10 OPEN active in SVA
		SELFTERM	08 Task terminating by itself (EOJ, CANCEL, DUMP, JDUMP, DETACH by user code)
			04 Reserved
		CNCLALL	02 CANCEL ALL request
		NOPAGING	01 No page faults allowed (system tasks)
8-11	8-B	TCBTIB	TIB pointer
12-15	C-F	TCBSAVE	Address of current save area
16-19	10-13	INTINFO	Saved interrupt information
19	13	SVCIC	SVC interruption code
20-23	14-17	AERREXIT	Address of cancel exit (used for system tasks only)
24-27	18-1B	TCBERBLK	Address of head queue error entry (system tasks only)
28-31	1C-1F	TCBSAV2	Address of second save area
.... End of TCB for system tasks without second save area			
24-103	18-67	TCBSSAVE	Second save area
104-251	68-FB		TCB work area (FETCH, CCW Translation, CCW fixing, SVC)
252-255	FC-FF	TCBCINF	Fetch cancel information
		TCBCALIB	Pointer to library name
256-259	100-103	TCBCASLB	Pointer to sublibrary name
260-263	104-107	TCBCANAM	Pointer to phase name
264-267	108-10B		Used for move mode
(268)	(10C)	TCBWLEN	Total length of TCB work area
.... End of TCB for system tasks without exits...........			

Task Control Block (TCB) (cont...)

Bytes			
Dec	Hex	Label	Description
268-299	10C-12B	TCBEXTAB	AB, IT, PC exit information
268	10C		AB exit flag byte
		EXITACT	X'80' AB exit routine active
268-271	10C-10F	TCBABPTR	Address of AB exit routine
272-275	110-113	TCBABSAV	Address of AB exit save area
276-283	114-11B	TCBABSEC	Address and save area of secondary AB exit
284	11C		PC exit flag byte
		EXITACT	X'80' PC exit routine active
284-287	11C-11F	TCBPCPTR	Address of PC exit routine
288-291	120-123	TCBPCSAV	Address of PC exit save area
292	124		IT exit flag byte
		EXITACT	X'80' IT exit routine active
		DELINT	40 IT interrupt processing delayed
292-295	124-127	TCBITPTR	Address of IT exit routine
296-299	128-12B	TCBITSAV	Address of IT exit save area
300-303	12C-12F	TCBEOTAD	Continuation address for End of Task clean-up
304	130	VTAMBGIN	AR TCB: Begin address of VTAM partition (set by VTAM)
		APSFLAG	Flag byte
			X'80 - 40' Reserved
		VTLTDLY	20 VTAM user exit delayed while task owns the LTA
			10 - 01 Reserved
305-307	131-133		User task TCB: Pointer to VTAM APT (set/used by VTAM)
308-311	134-137	VTAMEND	AR TCB: End address of VTAM partition (set by VTAM)
308	134	APSCNT	Count of open VTAM ACBs (maintained by VTAM)
309-311	135-137		Reserved for VTAM
312-315	138-13B	VTPCINF	Program check information (VTAM)
316	13C	VTAMFLG	Flag byte
		VTABEND	X'80' AR TCB: TPBAL issued (set by VTAM) User task TCB: Abnormal term. of a VTAM process
		VTSPSAV	40 PSW + registers in SVPCSAVE
		VTCDLY	20 Cancel delayed for VTAM
		VTAPDEL	10 VTAM AP exit delayed while terminator is active
		VTURX	08 VTAM user exit in control
		VTSVC	04 VTAM SVC active
		VTAPP	02 VTAM process active
		VTAMKO	01 Key 0 / supervisor state required for VTAM
317-319	13D-13F		Reserved
320-323	140-143	TCBECB	Address of ATTACH ECB (used only for user subtasks)
324-325	144-145	TCBSPOFF	Identification of dedicated GETVIS subpool
326-327	146-147		Reserved
328-331	148-14B	TCBCRCBC	Anchor of CRCB chain (XPCC exit)
(332)	(14C)	TCBLNG	Length of AR and main task TCB
.... End of user main task TCBs			
332-335	14C-14F	TCBSTADR	Address of system task deactivation routine
.... End of TCBs for system tasks with exits			
336-455	150-1C7	TCBUSAVE	Subtask save area in case of ATTACH without SAVE
(446)	(1C8)	STCBLNG	Length of subtask's TCB
.... End of subtask's TCB			

Bytes Dec	Hex	Label	Description
.... FETCH work area			....
104-143	68-8F	DFCBSAV	Save area (Registers 0-2,8-14)
144-147	90-93	DFWKLPNT	Phase load point
148-151	94-97	DFWKEPNT	Phase entry point
152-155	98-9B	DFWKUSEN	Pointer to user's directory entry
156	9C	DFCBSW1	Flag byte
		FIRSTDIR	X'80' First directory record
		INVUSEN	40 Invalid local list
		FLABMASK	20
		USERMASK	10 User task
		NODEVALD	08 No validation required
		PARTLOAD	04 Load into partition
		REALMASK	02 Request for real partition
		SYSAMASK	01 System task request
157	9D	DFCBSW2	Flag byte
		FIXPAG	X'80' Pageable FETCH part is fixed
		USERDUPD	40 Update user directory entry
		GENINT	20 CCW generation area exhausted
		FIXTXT	10 Area for phase read in is fixed
		ACTDIR	08 Directory entry active
		LASTTBL	04 Last text block is read in
		IDERR	02 ID mismatch during dir. read
		FLNKVIO	01 LNKEDT with option LINK
158	9E	DFWKRCOD	Return code
159	9F		Reserved
160-163	A0-A3	DFWKPHPT	Address of phase name
164	A4	DFWKFLAG	Option byte
		FLRETCOD	X'80' Return code required
		SVAUPD	40 Load/update SVA phase
		SDLUPD	20 Update SDL
			10 Reserved
		SDLFORM	08 Directory entry has SDL format
		SYSLIST	04 Search SYSLIB first
		DENTRY	02 Directory entry option
		NTXTNTRY	01 No text load option
165-167	A5-A7	DFWKLIST	Pointer to local list
168-175	A8-AF	DFWKNAME	Phase name
176	B0	DFWKEGEN	Reserved
177	B1		Phase attributes
178-179	B2-B3		Offset of PRBA-ADDR
180-183	B4-B7	DFWKERBA	Relative block address
184-185	B8-B9	DFWKECON	Number of contiguous blocks
186-187	BA-BB		Reserved
188	BC	DFWKESWT	Indicators
189	BD	DFWKEMVS	Status MOVE-MODE
190-191	BE-BF		Reserved
192-195	C0-C3	DFWKEPLN	Length of phase in bytes
196-199	C4-C7	DFWKELPL	Load point at LNKEDT time
200-203	C8-CB	DFWKEEPL	Entry point at LNKEDT time
204-207	CC-CF	DFWKEBGP	Part. start address at LNKEDT time
208-209	D0-D1	DFWKERLD	Number of RLD items
210-215	D2-D7	DFWKERDA	PRBA of RLD item
216-223	D8-DF		Reserved
224-227	E0-E3	DFWKEVLE	Entry point in SVA
228-231	E4-E7	DFWKLBID	Librarian identifier of phase
232-235	E8-EB	DFWKALIB	Address of Library Definition Table
236-239	EC-EF	DFWKASLB	Addr. of Sublibrary Definition Table
(240)	(F0)	DFWKEND	End of DE layout
240-243	F0-F3	DFWKANAM	Address phase name
244-247	F4-F7	DFWKCOMG	Pointer to actual COMREG
248-251	F8-FB	ANCSAV	Pointer to Anchor table
(148)	(94)	LFCHWORK	Length of fetch work area
...............End of FETCH work area			....

Task Control Block (TCB) (cont...)

Bytes			
Dec	Hex	Label	Description
.... CCW Translation work area (/370 mode)			
104	68	TCBFLAG	Flag byte
		TCBDC	X'80' Data chaining
		TCBRDS	40 Read/Sense command
		TCBRDB	20 Read Backward command
		TCBSM1	10 Status modifier command and command chaining
		TCBSM2	08 Status modifier command
		FXGETBL	04 Request for FIXINF block
		CHKSTM	02 Check status modifier 1287/3890
		GETDCBL	01 Double copy block request
105	69	ADBTAMCB	No. of addit. blocks needed by BTAM
106	6A		Reserved
107	6B	CCWTFLG2	CCW-Translation second flag byte
		CCWTFIDA	X'80' Fix IDAL request
		CCWTPAF	40 Page already fixed
			20 Reserved
			10 Reserved
			08 Reserved
			04 Reserved
			02 Reserved
			01 Reserved
108-111	6C-6F	DEVSTPTR	Pointer to status modifier list
112-115	70-73	DEVCDPTR	Pointer to control command list
116-119	74-77	LINEPTR	Pointer to next line
120-123	78-7B	BENDPTR	Block end pointer
124-127	7C-7F	TCBACCB	Address of copied CCB
128-131	80-83	IDALCNT	Number of free IDAL's
132-139	84-8B	DYNAREA1	Dynamic save area
140-147	8B-93	DYNAREA2	Dynamic save area
148-151	94-97	DYNAREA3	Dynamic save area
152-155	98-9B	DYNAREA6	Dynamic save area
136-159	9C-9F	DYNAREA7	Dynamic save area
160-163	A0-A3	FIXADDR	Address of last TFIX request
164-167	A4-A7	CCWTFREP	Address of free fix list entry
168-171	A8-AB	TCBDCB	Pointer to DIDAL block chain FASTTR
172-175	AB-AF	DDALBLAD	Address of current DIDAL block FASTTR
176	B0	TCBFLAG1	Flag byte FASTTR
		RFALG	X'80' REPLICA creation request FASTTR
			40 Reserved
		REPLCR	20 REPLICA block request FASTTR
		DIDALCR	10 Request for DIDAL block FASTTR
			08 Reserved
			04 Reserved
			02 Reserved
			01 Reserved
177	B1		Reserved FASTTR
178-179	B2-B3	DIDALCNT	No. of free DIDAL double words FASTTR
180-183	B4-B7	DIDAWAD	Address of current DIDAL FASTTR double word FASTTR
184-187	B8-BB	VCCWAD1	Stored virtual CCW address FASTTR
188-215	BC-D7	SAVEREG2	Save area for registers 2-8
216-219	D8-DB	SAVEREG9	Save area for register 9
220-223	DC-DF	SAVEREGA	Save area for register 10
224-227	E0-E3	SAVEREGB	Save area for register 11
228-231	E4-E7	SAVEREGC	Save area for register 12
232-235	E8-EB	SAVEREGD	Save area for register 13
236-239	EC-EF	SAVEREGD	Save area for register 14
240-243	F0-F3	SAVEREGD	Save area for register 15
(140)	(8C)	LCCWTAR	Length of CCW Translation work area
.... End of CCW Translation work area			

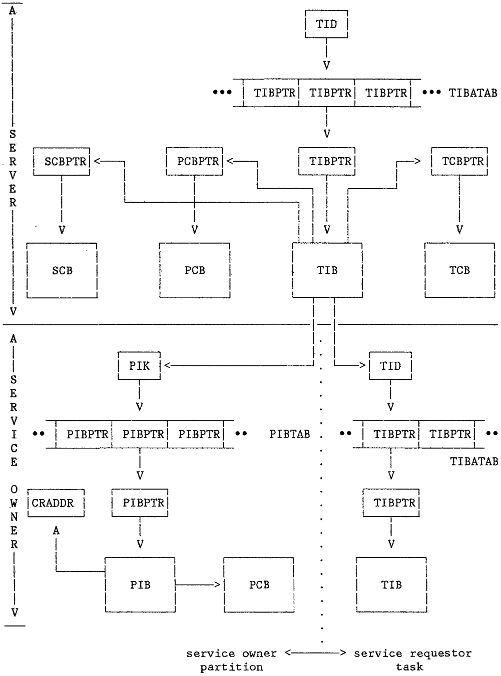
Task Control Block (TCB) (cont...)

Bytes		Label	Description
Dec	Hex		
			.... CCW Fixing work area (ECPS:VSE mode) ....
104	68	FRBFLAG1	Flag byte
		FRBDC	X'80' Data chaining specified
		FRBRDS	40 Read/Sense command
		FRBRDB	20 Read Backward command
		FRBSM1	10 Status modifier command and data chaining
		FRBSM2	08 Status modifier command and command chaining
		FRBSM3	04 Status modifier handling in process
			02 Reserved
		FRBDOIO	01 DOIO request
105	69	FRBFLAG2	Flag byte for FASTTR
		FRBRRQ	X'80' REPLICA creation required
		FRBCFL	40 Chained fixlist
		FRBVAL	20 Valid fixlist entry
		FRBRCS	10 Replica creation suppressed
			08 Reserved
			04 Reserved
			02 Reserved
			01 Reserved
106-107	6A-6B		Reserved
108-111	6C-6F	FRBSFADR	Address of SETFLAG routine
112-115	70-73	FRBSMPTR	Address of status modifier list
116-119	74-77	FRBCDPTR	Address of control command list
120-123	78-7B	FRBAFHB	Address of FHB (fixlist)
124-127	7C-7F	FRBLNPTR	Address of LINEPTR stack
128-131	80-83	FRBLLPTR	Address of Locate list
132-139	84-8B	FRBSAVLE	Save field for locate list entry
132-135	84-87	FRBALLE	Actual Locate list entry
136-139	88-8B	FRBWRK1	Work field 1
140-143	8C-8F	FRBWRK2	Work field 2
144-147	90-93	FRBWRK3	Work field 3
148-151	94-97	FRBWRK4	Work field 4
152-155	98-9B	FRBWRK5	Work field 5
156-159	9C-9F	FRBSAV	Save area for register 15
160-163	A0-A3	FRBSAV0	Save area for register 0
164-167	A3-A7	FRBSAV1	Save area for register 1
168-199	A8-C7	FRBSAV2	Save area for registers 2-9
200-203	C8-CB	FRBSAVA	Save area for register 10
204-211	CC-D3	FRBSAVB	Save area for registers 11-12
212-219	D4-DB	FRBSAVD	Save area for registers 13-14
(220)	(DC)	FRBEND	End of FRB
(116)	(74)	LCCWFAR	Length of CCW Fixing work area
			.... End of CCW Fixing area ....

Bytes		Label	Description
Dec	Hex		
			.... SVC work area ....
104-167	68-A7	SVCSV3	Save area for registers 9-8
168-215	A8-D7	SVCWORK	Work area
(112)	(70)	LSVCWORK	Length of SVC work area
			.... End of SVC work area ....

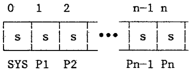
TASK SELECTION

Task Selection Control Block Interrelationship



Partition Selection String (PSS)

This bit string is located in low core (Label PSS in the Supervisor)



n = number of partitions
s equal 0 = no task of the partition is ready to run
s equal 1 = at least one task of the partition is ready to run
SYS,P1,P2,..,Pn = partition priorities

|——> priority order, high to low

Partition Priority Owner Table (PPRTYOWN)

(pointer to PPRTYOWN located in low core on fixed location X'2C4'.)

```
 0        4        8                          4*n
┌────────┬────────┬────────┬─     ─┬────────┬────────┐
│ pcbptr │ pcbptr │ pcbptr │  •••  │ pcbptr │ pcbptr │    fullword pointers
└────────┴────────┴────────┴─     ─┴────────┴────────┘
 SYS      P1       P2               Pn-1     Pn

|———> priority order, high to low

  pcbptr = pointer to partition control block of priority owner
  SYS,P1,P2,...,Pn = partition priorities
```

Task Selection String (TSS)

```
 0   1   2          31
┌───┬───┬───┬─    ─┬───┐
│ s │ s │ s │ ••• │ s │      A task is ready to run
└───┴───┴───┴─    ─┴───┘      when its status bit (s) is one.
 T0  T1  T2          T31

|————>  priority order, high to low
```

Task Identifier String (TIDSTR)

TIDSTR describes the priority of tasks within a partition.
It is located in the Partition Control Block (PCB).

```
┌─────┬─────┬─────┬─     ─┬─────┐
│ TID │ TID │ TID │  •••  │ TID │          1-byte entries
└─────┴─────┴─────┴─     ─┴─────┘
 T0    T1    T2            T31

|————>  priority order, high to low
```

TIB Address Table (TIBATAB)

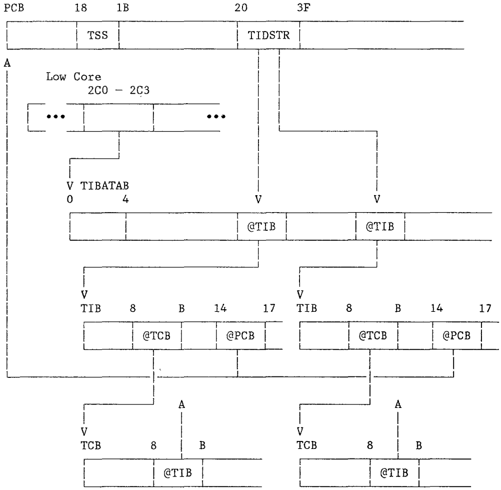
The address of the TIB Address Table (TIBATAB) can be found on fixed location X'2C0' in low core.

```
|<——System Tasks——>|<——————————User Tasks——————————>|
|                  |                                |
|                  |                                |
|0   4   8      128|                      192 |     |
┌───┬───┬───┬─   ─┬───┬───┬───┬─   ─┬───┬─   ─┬───┬─   ─┬───┐
│ 0 │TIB│TIB│ ••• │TIB│TIB│TIB│ ••• │TIB│ ••• │TIB│ ••• │   │
└───┴───┴───┴─   ─┴───┴───┴───┴─   ─┴───┴─   ─┴───┴─   ─┴───┘
| -  SNS DSK        |AR  BG  F1        Fn-1      |S1        Sm |
|                   |                            |             |
|                   |                            |             |
|                   |<————————Maintasks—————————>|<—Subtasks——>|
|———>4*TID (offset within TIBATAB)

Where:
      n =   number of partitions
      m =   number of subtasks          TIB =   Address of TIB
```

PCB Address Table (PCBATAB)

(See Partition priority owner PPRTYOWN)

Partition/Task Control Tables Relationship



@ = Address
@@ TIB = @ TIBATAB + TID x 4

SVC 107 (X'6B') Function Codes

Type	Value	Name	Usage
S	55	RSGTBND	Gate for real space segment table
S	56	SPFIXBND	Gate for PFIX in SVA processing
W	57	PWSRVBND	Gate for usage of POWER service
S	58	GQMGBND	Gate for usage of LOG queue manager
S	59	G117BND	Gate for usage of LOG service
S	5A	NPGRBND	Gate for usage of LUB allocation services
S	5B	VIOBND	Gate for virtual I/O support
0	5C	CONDRDY	Flag for conditional ready state
S	5D	IUCVBND	Gate for IUCV support for VCNA
S	5E	G108BND	Gate for usage of SVC–6C
S	5F	SATBND	Gate for usage of stored assign.table
S	60	CRTSVBND	Gate for CRTSAV usage
S	61	HCFCBBND	Gate for HC–file control block usage
S	62	ERQBND	Gate for error queue entry
S	63	G133BND	Gate for XPCC processing
S	64	OCFBND	Gate for operator comm. facility
S	65	OREBND	Gate for operator request element
S	66	EOTBND	Gate for EOT routine
C	67	SCYBND	Gate for security task
C	68	LCKBND	Gate for LOCK file I/O
C	69	PGFXBND	Gate for page to be freed
S	6A	GSMBND	Gate for ALLOCATE processing
S	6B	THTABBND	Gate for track hold table
C	6C	SFILBND	Gate for SYSFIL I/O
S	6D	SGTVSBND	Gate for GETVIS SVA
S	6E	LQBND	Gate for security logger queue
S	6F	CBFBND	Gate for console buffers
C	70	MICRBND	Gate for MICR I/O
S	71	GETRBND	Gate for GETREAL processing
S	72	FDIRBND	Gate for program fetch directory
S	73	SEIZEBND	Gate for SEIZE to be freed
S	74	CILBND	Gate for CIL update
S	75	BUFBND	Gate for copy blocks
C	76	ICCFBND	Gate for ICCF high priority task
S	77	PFRBND	Gate for page frames
S	78	PFGBND	Gate for page frames (occupied by TFIX)
S	79	CHQBND	Gate for channel queue entry
S	7A	DIBBND	Gate for DIB access
S	7B	CCWBND	Gate for CCW translation
W	7C	TRKBND	Gate for track to be freed
W	7D	AVRBND	Gate for AVR processing
S	7E	G41BND	Gate for ENQ/DEQ processing
S	7F	G92BND	Gate for XECB processing
C	80	NOTACT	Flag for inactive tasks
C	80	SYSBND	Flag for inactive system tasks
S	81	LTABND	Gate for LTA use
I	82	WAITBND	Gate for ECB/XECB (I/O or TIMER or POST)
0	83	READY	Flag for ready to run state
S	84	IDRABND	Gate for program fetch IDRA (old gate)
S	84	FPGMBND	Gate for program fetch IDRA (new gate)
C	85	FETCHBND	Gate for program fetch processing
W	86	PGIOBND	Gate for page I/O
C	87	PMRBND	Gate for page fault processing
I	88	ENQBND	Gate for RCB to be freed

Type:
0 = permanently opened gate
C = permanently closed gate
I = I/O chain with permanently closed gate
W = wait chain with permanently closed gate
P = partition chain with switchable gate,
P gates located in Partition Control Block (PCB)
S = system chain with switchable gate

Task Status Flags and Resource Gates (cont...)

Type	Value	Name	Usage
S	89	TERMBND	Gate for terminator processing
C	8A	PGINBND	Gate for page-in
S	8B	USEBND	Gate for LOCK/UNLOCK processing
C	8C	CNCLBND	Gate for subtask to be canceled
S	8D	SSIDBND	Gate for subsystem id processing
W	8E	RURBND	Gate for LOCK to be freed
S	8F	EXNTBND	Gate for EXTENT processing
P	90	GTVBND	Gate for partition GETVIS
P	91	CDLBND	Gate for CDLOAD
P	92	PFXBND	Gate for PFIX

Type: O = permanently opened gate
C = permanently closed gate
I = I/O chain with permanently closed gate
W = wait chain with permanently closed gate
P = partition chain with switchable gate,
P gates located in Partition Control Block (PCB)
S = system chain with switchable gate

SVC 107 (X'6B') Function Codes

MACRO	OPTION	FUNCTION CODE DEC/HEX		SERVICE CLASS	AUTHORIZATION
TREADY	LQ	00	00	A	LOG-TASK
TREADY	NO	01	01	A	IPL+LOG-TASK
TREADY	IO	02	02	A	KEY 0 PROGRAMS
TREADY	VTAM	03	03	A	VTAM
TREADY	CANCEL	04	04	A	VTAM+POW+ICCF
TREADY	VCANCEL	05	05	A	VTAM
GETFLD	SAVAR	06	06	B	CURR. TASK
GETFLD	PPSAVAR	07	07	B	CURR. TASK
GETFLD	LTAPTR	08	08	A	
GETFLD	CNCLCODE	09	09	A	
GETFLD	PIK	10	0A	A	
GETFLD	MAINTASK	11	0B	A	
GETFLD	VTAMOPEN	12	0C	A	VTAM
GETFLD	VTAMDISP	13	0D	A	VTAM
GETFLD	AOTPTR	14	0E	A	VTAM
MODFLD	SYSRESW	15	0F	A	KEY 0 PROGRAMS
MODFLD	CNCLCODE	16	10	A	VTAM+POWER+EOJ
MODFLD	VTAMOPEN	17	11	A	VTAM
MODFLD	VTAMDISP	18	12	A	VTAM
TREADY	START	19	13	A	JCL+POWER
TREADY	OC	20	14	A	JCL+POWER
TREADY	CANCEL	21	15	A	POWER
TSTOP	SYSBND,NO	22	16	C	SYSTEM-TASKS
TSTOP	SYSBND,YES	23	17	A	SYSTEM-TASKS
TSTOP	STOP	24	18	C	JCL
TSTOP	UNBATCH	25	19	C	JCL
GETFLD	CNCLALL	26	1A	B	TERMINATOR
GETFLD	ICCFPP	27	1B	A	ICCF
MODFLD	SAVAR	28	1C	B	IPL+EOJ
MODFLD	CNCLALL	29	1D	B	TERMINATOR
GETFLD	SYSRESW	30	1E	B	
GETFLD	ICCFRO	31	1F	A	ICCF
GETFLD	ACLOSE	32	20	B	CURR. TASK
GETFLD	STATUS	33	21	A	ICCF
MODFLD	ICCFPP	34	22	A	ICCF
MODFLD	ICCFRO	35	23	A	ICCF
MODFLD	ACLOSE	36	24	B	EOJ
GETFLD	NSUB	37	25	A	
GETFLD	CPUTIME	38	26	A	
MODFLD	VSAMOPEN	39	27	B	OPEN/CLOSE
GETFLD	ABINPR	40	28	B	ICCF
TREADY	ICCF	41	29	A	ICCF
GETFLD	LTAACT	42	2A	A	
GETFLD	OPENSVA	43	2B	B	CURR. TASK
MODFLD	OPENSVA	44	2C	B	CURR. TASK
MODFLD	ICCFSVC	45	2D	B	ICCF
GETFLD	PAGEIN	46	2E	A	
GETFLD	PAGEOUT	47	2F	A	
GETFLD	TERMACT	48	30	A	ICCF
GETFLD	EOTACT	49	31	A	ICCF
GETFLD	PCEXIT	50	32	B	CURR. TASK
GETFLD	ITEXIT	51	33	B	CURR. TASK
GETFLD	CNCLCOD2	52	34	A	
GETFLD	OCEXIT	53	35	B	CURR. TASK
TREADY	OCCF	54	36	A	OCCF
RLOCK	CRTSAV	55	37	C	OCCF
RLOCK	HCFCB	56	38	C	OCCF
TREADY	CRTSAV	57	39	A	OCCF
TREADY	HCFCB	58	3A	A	OCCF
TREADY	ATTINT	59	3B	A	OCCF
TREADY	OCCFIO	60	3C	A	OCCF

SVC 107 (X'6B') Function Codes (cont...)

MACRO	OPTION	FUNCTION CODE DEC/HEX		SERVICE CLASS	AUTHORIZATION
GETFLD	OCCFACT	61	3D	A	OCCF
GETFLD	BALANCE	62	3E	A	BAM
GETFLD	SSFLAGS	63	3F	A	SYSTEM-TASKS
GETFLD	COMRGPTR	64	40	A	
GETFLD	OWNER	65	41	A	
SRCHFLD	CHNUNIT	66	42	A	
SRCHFLD	DEVTYPE	67	43	A	
DEVUSE	PU	68	44	A	
DEVREL	PU	69	45	A	
SENTER	LIBR	70	46	A	
SLEAVE	LIBR	71	47	B	SYSTEM-TASKS
VIO	POINT	72	48	B	
GETFLD	USECNT	73	49	A	
GETFLD	PUSECNT	74	4A	A	
GETFLD	MOUNTFLG	75	4B	A	
MODFLD	MOUNTFLG	76	4C	A	JCL
TREADY	POWER	77	4D	A	POWER
GETFLD	PUBXPTR	78	4E	A	
GETFLD	PCBPTR	79	4F	A	
GETFLD	TCBPTR	80	50	A	
GETFLD	ABEXIT	81	51	B	CURR. TASK
GETFLD	MSECS	82	52	A	SYST.-TASKS+JCL
MODFLD	MSECS	83	53	A	SYST.-TASKS+JCL
VALID	READ	84	54	A	
VALID	WRITE	85	55	A	
GETFLD	VSAMOPEN	86	56	B	
MODFLD	PERBIT	87	57	B	SDAID
GETFLD	PU	88	58	A	
GETJA	PART	89	59	C	
MODFLD	RUNMODE	90	5A	A	SYSTEM
MODFLD	SASCOPE	91	5B	A	SYSTEM
MODFLD	PASCOPE	92	5C	A	SYSTEM+POWER+ OCCF
RLOCK	ALLOCR	93	5D	A	SYSTEM
RLOCK	RSGT	94	5E	A	SYSTEM
TREADY	ALLOCR	95	5F	A	SYSTEM
TREADY	RSGT	96	60	A	SYSTEM
MODFLD	LIBRSERV	97	61	A	KEY 0

How to locate Task status and Save Areas

COMREG

+5A	PIBTAB	+7C	PIB2TAB	

PIB2TAB & PIBTAB

+8 SYS	——> SYSPCB		+8 BG	——> BGPCB	
+8 Fn	——> FnPCB		————		
————			+8 F1	——> F1PCB	
AR			+4 BG	——> BGSAVE	
+4 Fn	——> FnSAVE		————		
————			+4 F1	——> F1SAVE	

PID & PIK (see Note 1)

		PIK Value in COMREG										
PID	Part. name	Number of Partitions										
		12	11	10	9	8	7	6	5	4	3	2
00	SYS	00	00	00	00	00	00	00	00	00	00	00
01	BG	10	10	10	10	10	10	10	10	10	10	10
0C	F1	C0	B0	A0	90	80	70	60	50	40	30	20
0B	F2	B0	A0	90	80	70	60	50	40	30	20	
0A	F3	A0	90	80	70	60	50	40	30	20		
09	F4	90	80	70	60	50	40	30	20			
08	F5	80	70	60	50	40	30	20				
07	F6	70	60	50	40	30	20					
06	F7	60	50	40	30	20						
05	F8	50	40	30	20							
04	F9	40	30	20								
03	FA	30	20									
02	FB	20										

SYSCOM (see Note 2)

+58	LIK/TIK		• • •	+11C	ICCF VT	

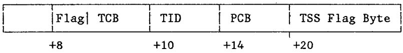
Low Core Pointers used for Task Selection:

Address	
X'248'	Address of PIBAREA (PIB2TAB,PIBTAB)
X'24C'	Address of System PCB
X'250'	Address of SCB address vector
X'254'	Address of current SCB
X'258'	Partition selection string (PSS)
X'260'	Address of currently active TCB
X'264'	Address of currently active TIB
X'268'	Address of currently active PIB
X'26C'	Address of currently active PCB
X'2C0'	Address of TIB table (TIBATAB)
X'2C4'	Address of Part. Priority Table (PPRTYOWN)

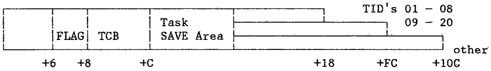
TIBATAB (see Note 3)

SNSTIB	DSKTIB	RASTIB	• • • •
• • • • • • • • • • •			ARTIB
BGTIB	F1TIB	F2TIB	• • • •
• • • • • • • • • • •			SUBTIB
SUBTIB	SUBTIB	• • • • • • •	
• • • • • • • • • • •			SUBTIB

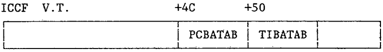
TIB (see Note 4)



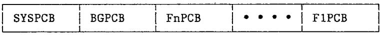
TCB



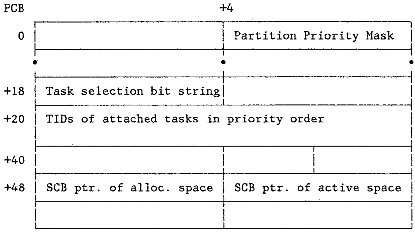
ICCF V.T.



PCBATAB



PCB



```
PSS bitstring     (see Note 5)
  0   1   2                          n-1  n
┌───┬───┬───┬─────────────────────────┬───┬───┐
│ s │ s │ s │ •  •  •  •  •  •  •  •  │ s │ s │
└───┴───┴───┴─────────────────────────┴───┴───┘
SYS  P1  P2                          Pn-1 Pn
├────────> priority order, high to low
```

TID

Tasks are identified by hexadecimal numbers 1 to n where n depends on the number of subtasks generated in the supervisor. The following table shows the task identifier (TID) values and their assignments to particular tasks:

System Tasks:

01	SNS	07	SUP	0D	reserved
02	DSK	08	DIR	0E	reserved
03	RAS	09	CRT	0F	SVT
04	PMR	0A	ASY	10 - 1F	reserved
05	reserved	0B	ERP	20	AR
06	PGN	0C	LCK		

Main Tasks:

20	AR	26	F5 *	2B	FA *
21	BG	27	F6 *	2C	FB *
22	F1 *	28	F7 *	2D	reserved
23	F2 *	29	F8 *	2E	reserved
24	F3 *	2A	F9 *	2F	reserved
25	F4 *				

* Depending on the number of partitions all or some of these keys may be unused (in descending order of values).

Subtasks: 30 - nn

Notes:

1. Pid like PIK assumes different values depending on the number of partitions generated (PID = PIK / 16).

2. LIK = Key of task owning LTA
   TIK = Key of task running

3. To correctly point to a TIBATAB entry, shift the task ID (TID) left by 2, and use the result as displacement in TIBATAB.

4. TIB + X'10' contains:
   User Task: TID of task
   System task: TID of serviced task

5. The value of 1 in PSS means: At least one task in selected partition is ready to run.

Event Control Block (ECB)

Reserved	Reserved	•	Reserved
0	1	2	3

X'80'	Event completed or terminated normally
X'C0'	Abnormal termination

Resource Control Block (RCB)

X'FF' or X'00'	RESERVED	X'80' or X'00'	ECB ADDRESS
0	1 3	4	5 7

Bytes	Description
0	X'FF' resource is in use X'00' resource is not in use
1 - 3	Reserved
4	X'80' Another task waiting for this resource
5 - 7	ECB address of current resource owner

Task Timer Table (TTTAB)

FLAG BYTE	EXIT ROUTINE ADDRESS	PSW KEY	SAVE AREA ADDRESS	TASK TIMER INTERVAL
0	1 3	4	5 7	8 15

Bytes	Description
0	Flag byte X'80' User EXIT routine already active 40 - 01 Reserved
1 - 3	User's STXIT TT routine address Zero if STXIT not yet issued
4	Caller's PSW key
5 - 7	Address of users save area Zero if STXIT not yet issued
8 -15	SETT issued: Interval time still left Bits 0-51 contain the time in microseconds Bits 52-63 are ignored No SETT issued: Zero or negative

Format of XECB Table Entry

Bytes	Description
0 – 7	XECB name
8	ACCESS Control byte
	X'80' Table entry in use
	40 Task that issued XPOST was canceled
	20 Reserved
	10 Reserved
	XWAIT access indicator
	08 ACCESS=XPOST specified
	04 ACCESS=XWAIT specified
	XPOST access indicator
	02 ACCESS=XWAIT specified
	01 ACCESS=XPOST specified
9 – 11	XECB address
12 – 13	TID of owner
14 – 15	TID of first task that posted XWAIT or XPOST for XECB
16 – 19	Forward chain pointer
20 – 23	Backward chain pointer

If bits 4 and 5 of byte 8 are set to 10, bytes 14 and 15 contain the TID of the first task that issued an XWAIT for this XECB.

If bits 6 and 7 of byte 8 are set to 10, bytes 14 and 15 contain the TID of the first task that issued an XPOST for this XECB.

Label XECBTAB identifies the first byte of the table.

SYSTEM COMMUNICATION REGION (SYSCOM)

Bytes		Label	Description
Dec	Hex		
0–3	0–3	IJBERBLC	Address of error block
4–7	4–7	IJBHWC	Hard Wait Code (Note 2)
8–11	8–B	IJBERR19	Address of CANCEL exit for ERP message writer
12–15	C–F	IJBPUBRS	Pointer to SYSRES PUB (set by IPL)
16–19	10–13		Reserved
20–23	14–17	IJBSPAVT	Address of supervisor address vector
24–27	18–1B	IJBPTCOM	Address of VSE/PT communication area
28–31	1C–1F	IJBLTA	Address of Logical Transient Area
32–35	20–23	IJBPPBEG	Begin of Problem Program Area (set by IPL)
36	24	IJBFLPTR	Free list pointer
37–39	25–27	IJBCHANQ	Address of Channel Queue
40–41	28–29	IJBQSIZE	Number of Channel Queue entries
42–43	2A–2B	IJBQLNG	Length of ERP Error Queue entry
44–45	2C–2D	IJBNPART	Number of partitions
46	2E	IJBFLG05	Flag byte
		IJBAF	X'80' AF System (always on)
			40 DOS/VSE System (always on)
			20 TP Balancing not active
			10 Reserved
			08 Console Buffering (CBF) active
			04 Reserved
		IJBCSCDS	02 CS and CDS supported
		IJBSIPOF	01 SIPO format flag
47	2F	IJBFLG06	Flag byte
		IJBEMODE	X'80' ECPS:VSE mode
			40 Reserved
			20 Reserved
			10 Reserved
			08 Reserved
		IJBCKD	04 CKD support generated (always on)
		IJBFBA	02 FBA support generated (always on)
		IJB3800	01 3800 support generated (always on)
48–51	30–33	IJBVSIZE	Total virtual storage size
52	34	IJBCONSP	
		IJBOCFLG	DOC configuration byte
			X'80' CRT support initialized
			40 Reserved
			20 Reserved
			10 Reserved
			08 Reserved
			04 Reserved
			02 Support for 3277 screen (always on)
			01 CRT support generated (always on)
53–55	35–37		Address of Console Communication Area (Address of CRT Table)
56–59	38–3B	IJBOCFCM	Address of OCCF Communication Area
60–63	3C–3F	IJBVIOCM	Address of VIO Communication Area

Note:
The address of SYSCOM can be found at fixed location X'80' - X'83'.

System Communication Region (SYSCOM) (cont...)

Bytes			
Dec	Hex	Label	Description
64	40	IJBFLG01	RMS flag byte
		IJBRMSR	X'80' RMSR support generated (always on)
		IJBRMS	40 Full RMS support (always on)
			20 Always off
			10 Reserved
			08 Reserved
			04 Reserved
			02 Reserved
		IJBITDWN	01 IT support down (Clock damage)
65	41	IJBFLG02	Flag byte
		IJBCBF	X'80' Console buffering active
		IJBJA	40 Job Accounting support active (SYS JA=YES)
		IJBDSDFP	20 DASD File Prot. support active (SYS DASDFP=YES)
		IJBSEC	10 Access Control support active (SYS SEC=YES)
			08 Reserved
			04 Channel Scheduler entered after interrupt
		IJBMPXGT	02 Byte MPX channel gating (switched on/off by AR MPXGTN ON/OFF)
		IJBIPLAC	01 IPL in progress
66	42	IJBFLG03	Flag byte
			X'80' Reserved
			40 RAS in special WAIT state
			20 RAS IPL in progress
		IJBIPLV	10 Virtual storage has been initialized by IPL
			08 VSE/POWER supported (always on)
			04 VSE/POWER initialized
			02 GETREAL in progress
			01 Reserved
67	43	IJBFLG04	Flag byte
			X'80' System GETVIS Area initialized
			40 EXCP REAL supported (always on)
			20 CDLOAD supported (always on)
		IJBVMBTM	10 BTAM AUTOPOLL enabled
			08 XECB supported (always on)
			04 Reserved
			02 Batch deactivated by TPIN
		IJBVMLE	01 VM Linkage Enhancements (MODE=VM)
68-69	44-45	IJBHSTID	Highest system task TID
70-71	46-47	IJBHMTID	Highest maintask TID
72-75	48-4B	IJBVPBEG	Begin of V-Pool for VIO
76	4C	IJBTHPTR	Track Hold free list header
77-79	4D-4F	IJBTKHLD	Address of Track Hold Table
80-87	50-57		Reserved
88-89	58-59	IJBLIK	Task ID of LTA owner
90-91	5A-5B	IJBTIK	Task ID of current task
92-95	5C-5F	IJBPWR	Address of POWER Table
96-99	60-63	IJBTCAVT	Address of VTAM Address Vector Table
100-103	64-67	IJBRFTAB	Address of RF Table
104-107	68-6B	IJBEUECB	Reserved
108-111	6C-6F	IJBOLTEP	Flag byte and address of OLTEP Bucket
108	6C	IJBOLTSW	Flag byte
		IJBOLTAC	X'80' OLTEP is active
109-111	6D-6F	IJBOLTPT	Address of OLTEP Bucket
112-115	70-73	IJBRASLN	Address of RAS Linkage Area
116-119	74-77	IJBTRTAB	Address of ASCII Table
120-123	78-7B	IJBPBOWN	Address of PUB Ownership Table

System Communication Region (SYSCOM) (cont...)

Bytes			
Dec	Hex	Label	Description
124–127	7C–7F	IJBJATAB	Addr. of Job Accounting Common Table
128–131	80–83	IJBPROCT	Address of Procedure Common Table
132–135	84–87	IJBIJBSD	Used by SDAID
136–139	88–8B	IJBSAVSD	Address of SDAID Area
140–143	8C–8F	IJBLNSTB	Address of Line Mode Table
144–147	90–93	IJBARBUF	Address of AR input buffer
148–151	94–97	IJBAPTA	Address of Physical Transient Area
152–153	98–99	IJBNDEV	Number of ADD–ed devices
154–155	9A–9B	IJBNSDEV	Number of ADD–ed partition sharable devices
156–157	9C–9D	IJBVTPIK	VTAM PIK (set by SUBSID)
158–159	9E–9F	IJBPWPIK	POWER PIK (set by SUBSID)
160–161	A0–A1	IJBICPIK	ICCF PIK (set by SUBSID)
162–163	A2–A3		Reserved
164–165	A4–A5	IJBLPBDV	PUB pointer of printer buffer load
166–167	A6–A7	IJBPHLSL	Length of phase load list
168–169	A8–A9	IJBDMPDV	cuu of SYSDMP device (from IPL DEF command)
170–171	AA–AB	IJBRECDV	cuu of SYSREC device (from IPL DEF command)
172–173	AC–AD	IJBCATDV	cuu of SYSCAT device (from IPL DEF command)
174–175	AE–AF	IJBRESDV	cuu of SYSRES device
176–179	B0–B3	IJBTTAB	Address of Task Timer Table
180–183	B4–B7	IJBSMCOM	Addr. of storage management comm.area
184–187	B8–BB	IJBPMCOM	Addr. of page management comm. area
188–189	BC–BD	IJBTPBAL	TP Balancing parameter
190–191	BE–BF	IJBTTPID	PIK of partition owning the Task Timer
192–202	C0–CA	IJBMFCER	Repositioning information for 2560/5424/5425 ERP
203	CB	IJBNERQ	Number of ERP Error Queue entries (always=1)
204–205	CC–CD	IJBPUBLN	Length of PUB Table
206–207	CE–CF	IJBAPNO	Number of active virtual partitions
208–211	D0–D3	IJBSEGT	Address of Segment Table (only for MODE=370)
212–215	D4–D7	IJBAPT	Address of page table
216–217	D8–D9	IJBNPGR	Total number of programmer LUBs (NPGR parameter)
218–219	DA–DB	IJBGHLUB	Highest used BG programmer Logical Unit
220–223	DC–DF	IJBASMCB	Address of SMCB Address Table
224–227	E0–E3	IJBDPDTB	Address of DPD Table
228–229	E4–E5	IJBOCDEV	cuu of SYSLOG device
230–231	E6–E7	IJBNTASK	Number of subtasks supported
232–235	E8–EB	IJBSSBEG	Addr. of first byte after supervisor
236–239	EC–EF	IJBEOR	End of real storage (only for MODE=370)
240–243	F0–F3	IJBFTTAB	Address of system library offsets for FETCH
244–247	F4–F7	IJBSVA	Flag and address of SVA
244	F4	IJBSVAFL	Flag byte for Shared Virtual Area
			X'80' Reserved
			40 SDL active
			20 Reserved
			10 SDL build in progress
			08 SDL overflow
			04 High level SDL search
			02 Reserved
			01 Reserved
245–247	F5–F7	IJBSVAAD	Address of Shared Virtual Area
248–251	F8–FB	IJBSVIS	Address of System Getvis Area
252–255	FD–FF	IJBARPSL	Address of RPS Local Directory List

System Communication Region (SYSCOM) (cont...)

Bytes		Label	Description
Dec	Hex		
256-259	100-103	IJBARPSR	Address of RPS Sector Calculation Routine
256	100	IJBRPSIS	RPS flag byte
260-263	104-107	IJBDLAB	Address of System Code
264-267	108-10B	IJBASY	Flag and addr. of Asynch. Operator Communication Table
264	108	IJBASYFL	Flag byte for Asynchronous Operator Communication
			X'80' Reserved
			40 ASYNOC task is active
			20 Read is requested
			10 Reply or command is already in input buffer
			08 Reserved
			04 Print message 0D13D
			02 Message 0D13D has been printed
			01 Reserved
265-267	109-10B		Address of Asynchronous Operator Communication Table
268-271	10C-10F	IJBSLACB	Address of SLA work areas
272-275	110-113	IJBSVIPL	Address of Supervisor-IPL Communication Area
276-279	114-117	IJBAMSVA	Address of SVA module area
280-283	118-11B	IJBNPDA	Address of NPDA appendage
284-287	11C-11F	IJBETSS	Address of ICCF Vector Table
288-291	120-123	IJBSCTAB	Address of Security Vector Table
292-295	124-127	IJBPCSAV	Address of Special Save Area for error in system code
296-299	128-12B	IJBINSTR	Pointer to instrumentation data
300-303	12C-12F	IJBPLCT	Address of Librarian Control Table
304	130	IJBFINSC	End of system communication area

PARTITION COMMUNICATION REGION (COMREG)

Bytes			
Dec	Hex	Label	Description
0-7	0-7	JOBDATE	MM/DD/YY or DD/MM/YY Updated by GETIME macro or set by DATE. Format is controlled by bit 0 of byte 53, see below.
8-11	8-B		Reserved
12-22	C-16	COMUSCR	User area
23	17	UPSI	User program switch indicator (UPSI byte)
24-31	18-1F	COMNAME	Job name from JOB statement
32-35	20-23	PPEND	End address of program space within partition
36-39	24-27	HIPHAS	End address of last phase loaded
40-43	28-2B	HIPROG	End address of largest phase for a multi-phase program (see SVC
44-45	2C-2D	LABLEN	51-X'33') Length of Problem Program label area (always 0)
46-47	2E-2F	PID	Partition identifier (PIK), same as PIB offset BG COMREG: PIK of active partition
48-51	30-33	EOCADR	End address of virtual storage
52	34	CONFIG	Machine configuration byte
			X'80' Standard storage protection (always on)
			40 Decimal feature (always on)
			20 Floating point feature (always on)
			10 Physical transient overlap option (always on)
			08 Standard timer feature (always on)
			04 Channel switching supported (always on)
			02 Support for burst mode on byte MPX (always on)
		RMSBIT	01 RMS support available (always on)
53	35	LTACT	System configuration byte
		DDMMYY	X'80' DDMMYY date format convention
			40 Two or more partitions (always on)
		DASDFPSW	20 DASD file protect active (SYS command)
			10 SYSFIL support (always on)
			08 Teleprocessing support (always on)
			04 Two or more partitions (always on)
			02 Multitasking support (always on)
			01 Track Hold support (TRKHLD parameter)
54	36	SOB1	Standard language translator options (generated value 1100110, changed by STDOPT statement)

Note:
The address of the communication region of the active partition can be found at fixed location X'14' - X'17'.

Partition Communication Region (COMREG) (cont...)

Bytes Dec	 Hex	Label	Description
54	36	SOB1 (cont.)	X'80' DECK option, object modules on SYSPCH
			40 LIST option, source listings and diagnostic on SYSLST
			20 LISTX option, hexadecimal object modules listings on SYSLST
			10 SYM option, symbol tables on SYSLST/SYSPCH
			08 XREF option, cross reference list on SYSLST
			04 ERRS option, diagnostics on SYSLST
			02 CHARSET option, 60 character set (else 48)
			01 Reserved
55	37	SOB2	Flag byte
			X'80' Always on
			40 STDOPT DUMP=YES or PART
			20 Partition waiting for volume mount (Job Control)
			10 STDOPT LOG=YES
		DUMDVC	08 Dummy device search in progress
			04 Reserved
			02 Relocating loader supported (always on)
			01 ASCII supported (always on)
56	38	JCSW1	Flag byte
		JASWITCH	X'80' Job Accounting not active (SYS command)
		JCOPEN	40 Return to caller on LIOCS disk open failure)
		JCINRDR	20 Job Control input from SYSRDR (else SYSLOG)
			10 Job Control output on SYSLOG
		JOBEND	08 Skip to end of job
			04 Pause at end of job step (JC PAUSE statement)
			02 Always 0
			01 SYSLOG assigned to same device as SYSLST
57	39	JCSW2	Linkage Editor control byte
			X'80' SYSLNK open for output
		IJBACTCL	40 Action clear indicator
		ALLOWEX	20 Allow EXEC
			10 Catalog Linkage Editor output
		IGNTESTM	08 Ignore test mode
			04 – 01 Reserved
58	3A	JCSW3	Non-standard language translator options (set by OPTION statement)
			X'80' DECK option, object modules on SYSPCH
			40 LIST option, source listings and diagnostic on SYSLST
			20 LISTX option, hexadecimal object modules listings on SYSPCH
			10 SYM option, symbol tables on SYSLST/SYSPCH
			08 XREF option, cross reference list on SYSLST
			04 ERRS option, diagnostics on SYSLST
			02 CHARSET option, 60 character set (else 48)
			01 Rewind/unload option

Partition Communication Region (COMREG) (cont...)

Bytes			
Dec	Hex	Label	Description
59	3B	JCSW4	Job Control flag byte
			X'80' Job in progress
		OPTDUMP	40 OPTION DUMP
			20 Pause at end of job step (AR PAUSE command)
			10 OPTION LOG
			08 Temporary assignment for SYSRDR
		TESTMODE	04 SDL scanned as specified by LIBDEF
		DATEBIT	02 DATE statement processed for current job
		BATINIT	01 START/BATCH command just issued
60	3C	JCSW7	Job control flag byte
		OPCNCL	X'80' Indicator for operator cancel
		JCLTSTRN	40 OPTION TSTRUN
		LIBPCHNG	20 LIBDEF PROC change
		PRCFRSTL	10 Used to control check for PROC statement
		PRCFRSTH	08 Used to control check for PROC statement
		IJBOVLOG	04 Procedure overwrite statements to be read from SYSLOG
		IJBJCCNL	02 Job Control CANCEL issued
		IJBUSRMD	01 User mode
61	3D	NSTLEVEL	Procedure nesting level
62	3E	JCSW8	Job control flags
		IJBCNCPD	X'80' Operator cancel pending
		IJBRCCNC	40 RC operator cancel
		IJBARCNA	20 Delay AR cancel
		IJBEOPDL	10 EOP delayed
		IJBABTRM	08 Abnormal termination
			04 Reserved
			02 Reserved
			01 Reserved
63	3F		Reserved
64–65	40–41	PUBPT	Address of PUB Table
66–71	42–47	IJBJOBST	Job start time
72–73	48–49	FICLPT	Address of FICL
74–75	4A–4B	NICLPT	Address of NICL
76–77	4C–4D	LUBPT	Address of partition LUB Table
78	4E	SYSLINE	SYSLST line count as specified by STDOPT LINES=nn
79–87	4F–57	SYSDATE	System date, MMDDYYDDD or DDMMYYDDD
79–82	4F–52	MMDD	MMDD or DDMM
83–87	53–57	YYDDD	YYDDD portion of date
88	58	LIOCSCOM	LIOCS communication byte 1
			X'80' Reserved
		LIOCSRDS	40 Return to $$BODSMO
			20 Reserved
			10 Reserved
			08 Reserved
		LIOCSOIP	04 Open/close in progress
		LIOCSCPO	02 CP/DI open indicator
			01 Reserved
89	59		LIOCS communication byte 2
		LIOCSRSV	X'80' Return from SVA
			40 Reserved
			20 Reserved
			10 Reserved
		LIOCSRTM	08 Reserved
		LIOCSRLB	04 Return from $$BOPLBL
		LIOCSQMT	04 QTAM DTF
		LIOCSRLK	02 Return from 'LOCK'
			01 Reserved
90–91	5A–5B	PIBPT	Address of PIB Table
92–93	5C–5D	CHKPTID	ID of last checkpoint

Partition Communication Region (COMREG) (cont...)

Bytes			
Dec	Hex	Label	Description
94-95	5E-5F	JOBZON	Job zone in minutes.
			Value is positive for ZONE=EAST
			and negative for ZONE=WEST.
96-97	60-61	DIBPT	Address of partition DIB Table
98	62	DEVFLG1	Flag byte
		OPN3800	X'80' One or more 3800 extended buffering DTF's open
			40 Reserved
			20 Reserved
			10 Reserved
			08 Reserved
			04 Reserved
			02 Reserved
			01 Reserved
99	63	OPNBYT2	Flag byte
		BTAMFLG	X'80' BTAM active in partition
			40 Reserved
			20 Reserved
			10 Reserved
			08 Reserved
			04 Reserved
			02 Reserved
			01 Reserved
100-105	64-69		Reserved
106-107	6A-6B	PWTIMS	PIK of partition
108-109	6C-6D	IJBSPID	Space id (370 only)
110-111	6E-6F	LTK	PIK of part. owning the LTA
			(set only in BG COMREG)
112-115	70-73	SYSPAR	Address of SYSPARM field
116-119	74-77	JAPART	Address of Job Accounting Table
120-123	78-7B	TODCOM	Address of TOD common area
124-125	7C-7D	PIB2PTR	Address of PIB2 Table
126-127	7E-7F	PDTABB	Address of MICR DTF Table
128-131	80-83	LABELPTR	Reserved for LIOCS
132-133	84-85	BGCOMPT	Address of BG COMREG
134	86	OPTNBYTE	Flag byte
			X'80' Reserved
			40 Reserved
			20 Reserved
			10 Reserved
			08 Reserved
			04 Reserved
		JAPGCIND	02 Count pages for Job Accounting
		ANCHTBIT	01 GETVIS area initialized
135	87	RMSROPEN	Flag byte
			X'80' PCIL support (always on)
		TODBIT	40 TOD support (always on)
			20 PFIX support (always on)
			10 Fetch $$BOPEN by $JOBCTLJ
			08 Fetch $$BOPEN by $JOBCTLD
			04 Fetch $$BOPEN by $JOBCTLJ
			02 Reserved
			01 RPS support
136-139	88-8B	IJBJCWA	Addr. of job control work area
140	8C	STDOPT	Job Control standard option
			(STDOPT statement)
			Generated value is 010000-0
			X'80' EDECK
			40 ALIGN
		OPTPDUMP	20 PARTDUMP
			10 RLD
			08 SXREF
			04 TERM
			02 Reserved
			01 ACANCEL

Partition Communication Region (COMREG) (cont...)

Bytes		Label	Description
Dec	Hex		
141	8D	TEMOPT	Job Control temporary option (OPTION statement)
			X'80' EDECK
			40 ALIGN
			20 PARTDUMP
			10 RLD
			08 SXREF
			04 TERM
			02 SUBLIB=DF
			01 Reserved
142	8E	DISKCONF	Disk configuration byte
			X'80 – 10' Reserved
			08 3350 support (always on)
			04 3340 support (always on)
			02 3330 support (always on)
			01 2311 and 2314/2319 support (always on)
143–150	8F–96	PROCNAM	Procedure name
151	97	PSWTCH	Interface byte for Cataloged Procedures
		IJBPCALL	X'80' Procedure being executed
		IJBPOVMD	40 Overwrite processing
		IJBPDATA	20 SYSIPT data present
		IJBPOVRQ	10 Overwrite request for Job Control
		IJBPINST	08 Insert request for Job Control
		IJBPNDMK	04 End of procedure
		IJBPSLOG	02 Called from SYSLOG
		IJBPOVBT	01 Overwrite request for supervisor
152–158	98–9E	POVNAM	JCL statement name for Cataloged Procedure
159	9F	INSIZE	Flag byte
			X'80' Permanent 81 bytes on SYSRDR
			40 Permanent 81 bytes on SYSIPT
		RDR81T	20 Temporary 81 bytes on SYSRDR
		IPT81T	10 Temporary 81 bytes on SYSIPT
			08 – 02 Reserved
		CATALSA	01 Allow /& within procedure to be cataloged
160–163	A0–A3	POWPCB	Pointer to VSE/POWER Partition Control Block
164	A4	POWFLG1	VSE/POWER flag byte
		POWACCT	X'80' POWER accounting supported
		POWUPART	40 POWER controlled partition
		POWPART	20 POWER partition
		POWPDORM	10 POWER partition dormant
		POWWPART	08 POWER controlled partition waiting for work
			04 – 01 Reserved
165	A5	POWFLG2	Reserved for VSE/POWER
166–167	A6–A7	IJBVSSNP	VSAM snap dump function bytes
166	A6	IJBSNP01	X'80' SNAP dump indicator 1
			X'40' SNAP dump indicator 2
			X'20' SNAP dump indicator 3
			X'10' SNAP dump indicator 4
			X'08' SNAP dump indicator 5
			X'04' SNAP dump indicator 6
			X'02' SNAP dump indicator 7
			X'01' SNAP dump indicator 8

Partition Communication Region (COMREG) (cont...)

Bytes Dec	 Hex	Label	Description
167	A7	IJBSNP09	X'80' SNAP dump indicator 9
			X'40' SNAP dump indicator 10
			X'20' SNAP dump indicator 11
			X'10 - 01' Reserved
168-171	A8-AB	LUBEXT	Address of LUB Extension Table
172	AC	JCSW5	Flag byte
			X'80' EXEC LNKEDT statement to be generated
			40 EXEC statement to be generated
			20 Skip link and execution, except for OPTION LINK
			10 NEWVOL ignored
		LSTLOG	08 LISTLOG called for cancel
		ASIPL	04 ASI IPL
		ASICONT	02 Job Control first time activation passed
		JCLACTIV	01 Job Control active
173	AD	JCSW6	Flag byte
			X'80' Reserved
		ONLNSYSG	40 On-line system generation
			20 Reserved
		JOBLOGSW	10 Write job statement to HC file
		JCLOUTA	08 Alternate assignments exist for SYSOUT
		SLAACTIV	04 SLA active
		SYSPROC	02 System procedure library in use
		IJBFNSLB	01 Allow to add system labels from this partition (Fn)
174	AE	STDOPT2	Reserved for Job Control standard options (STDOPT)
		OPTNFSTR	X'80' NOFASTTR
		OPTSDUMP	40 SYSDMP
		OPTPROC	20 PROC
		OPTPARM	10 PARM
		OPTJCNCL	08 JCANCEL
		OPTNHCTR	04 NOHCTRAN
			02 Reserved
			01 Reserved
175	AF	TEMOPT2	Job Control temporary options (OPTION statement)
			X'80' NOFASTTR
			40 SYSDMP
			20 PROC
			10 PARM
			08 JCANCEL
			04 NOHCTRAN
			02 Reserved
			01 Reserved
176-179	B0-B3	IJBJPL	Address of JPL of partition
180-183	B4-B7	IJBAFCB	Reserved for CICS
184-187	B8-BB	IJBPHLST	Address of Fetch/Load Trace Table
188-195	BC-C3	IJBJOBLG	Address of last job statement on
188	BC		Cycle byte of job statement
189-193	BD-C1	IJBDSKAD	Disk address of job statement
194-195	C2-C3	IJBDSKLR	Logical record of job statement
196-199	C4-C7	IJBASPF	Address of SPF control information
200-203	C8-CB	IJBGVCTL	Address of GETVIS control information
204-207	CC-CF	IJBIJJT	Address of Tape Open control block
208-215	D0-D7	IJBSPNAM	System GETVIS partition pool
216-223	D8-DF	IJBPHNAM	Exec phase name
224-227	E0-E3	IJBDECPY	Mirror DE entry chain
228	E4	COMREND	End of partition communication region

Storage Management Control Block (SMCB)

SMCB Address Table Format:

Address of SVA Entry	Address of BG Entry	Address of Fn-1 Entry	••••	Address of F1 Entry
0	4	8	12	4xn

n = Number of partitions specified at supervisor generation (NPART)

Note:
The pointer to the SMCB Address Table can be found in the SYSCOM at offset 220 (X'DC').

SMCB Entry Format (SMCB)			
DEC	HEX	Label	Description
0	0	SMAXPFIX	Partition: PFIX limit in pages System : SVA PFIX limit in pages
2	2	SMPFIX	Partition: PFIX count in pages System : SVA PFIX count in pages
4	4	SMPSAVE	Partition: Save area address System : Reserved
8	8	SMVFLAG	Partition: GETVIS area flags X'80' : SETLIMIT given indicator
8	8	SMVGVIS SMSGVIS	Partition GETVIS area address System GETVIS area address
12	C	SMVPBEG SMSVABEG	Virtual Partition Begin Address SVA Begin Address
16	10	SMVPEND SMSVAEND	Virtual Partition End Address + 1 SVA End Address + 1
20	14	SMRPBEG	Real Begin Address
24	18	SMRPEND	Real End Address + 1
28	1C	<—— Length of SMCB	

Format of Storage Management Communication Area (SMCOM)

Storage Management Communication Area (SMCOM)			
DEC	HEX	Label	Description
0	0	SMALCVSZ	Allocated virtual storage in K.
4	4	SMFSVP	Size of fixed supervisor in pages
6	6	SMPPMIN	Minimum page pool in pages
8	8	SMINSVPX	Minimum system real partition in pages
10	A	SMINPART	Minimum partition size in K
12	C	SMINSIZE	Minimum permanent virtual 'SIZE' in K
14	E		Minimum temporary virtual 'SIZE' in K
16	10		Reserved
18	12		Minimum temporary real 'SIZE' in K
20	14	SMINGTVS	Minimum permanent virtual GETVIS in K
22	16		Minimum temporary virtual GETVIS in K
24	18		Reserved
26	1A		Minimum temporary real GETVIS in K
28	1C	<—— Length of SMCOM	

Format of the GETVIS Control Information Area (Anchor Table)

GETVIS Area Control Information Layout (MAPGVCTL)

DEC	HEX	Label	Description
0	0	ANCHDIR	Start of 51 CDLOAD Entries
1024	400	BVIRTMEM	Pointer to begin of GETVIS area
1028	404	EVIRTMEM	Pointer to end of GETVIS area
1032	408	BVISTAB	Begin of VISTAB
1036	40C	EVISTAB	End of VISTAB
1040	410	BSUBPIND	Begin of Subpool Index Table
1044	414	ESUBPIND	End of Subpool Index Table
1048	418	BSUBPCHN	Begin of Subpool Page Chain Table
1052	41C	ESUBPCHN	End of Subpool Page Chain Table
1056	420	EGVCTLB	Last byte of control information
1060	424	ENDGVCTL	End of control area
1064	428	GTVSHIGH	Page Chain high water mark
1068	42C	FIRSTPNT	First page within empty pool
1072	430	CURPOINT	Start of chain of last used pages
1076	434	SSEARCH	New start search address (work field)
1080	438	SVWORK1	Save area for register 1
1084	43C	NBRGVPG	Number of pages in GETVIS area
1086	43E	GTVSPGCT	Number of current used pages
1088	440	GTVSMXCT	Maximum number of pages to be used
1090	442	GTVSEXCT	Max. number of pages for excessive requestors
1092	444	MXSUBPLH	Maximum number of subpools available
1094	446	VISTAB	Begin of bit pattern
xxxx	XXX		Begin subpool of Index Table
yyyy	YYY		Begin subpool of Chain Table
zzzz	ZZZ	<— Length of Anchor Table, depends on length of VISTAB, Subpool Index Table and Subpool Chain Table	

Note:
The Anchor Table is located at the end of the virtual partition. It is pointed to by the partition COMREG (offset X'C8').

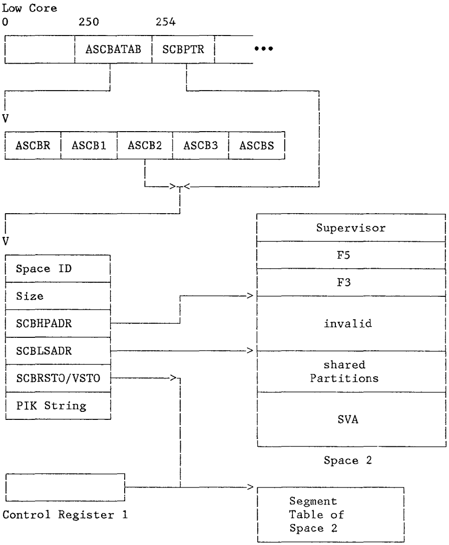
Due to compatibility reasons, the VSAM control information remains at the same location within the GETVIS area, i.e. it has the same offsets relative to PPEND as in former releases. The mapping macro for the VSAM control information is still MAPANCH and contains only this information.

Format of Anchor Table Entry

Anchor Table Entry Layout (ATENTRY)

DEC	HEX	Label	Description
0	0	ATPHSNME	Phase Name Field
8	8	ATLOADP	Load Point in GETVIS Area
12	C	ATENTP	Entry Point in GETVIS Area
16	10	ATPHSLEN	Length of loaded Phase
20	14	<—— Length of Anchor Table Entry (ATENTRY)	

Space Control Block (SCB) Data Relationship for a /370 VAE System

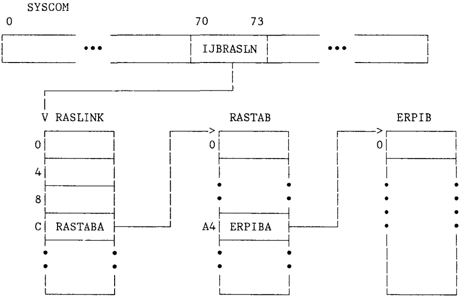


Space Control Block

Bytes Dec	Hex	Label	Description
0	0	SCBID	Symbolic Space Identifier predefined values: 'R' for real space '1' for primary virtual spaces 'N' for additional virtual spaces (370) 'S' for shared virtual spaces (370)
2	2	SCBSPN	Space number
3-5	3-5		reserved
6-7	6-7	SCBSIZE	Size of allocated partition space in K
............End of SCB of E and VM mode...........................			
8-11	8-B	SCBHPADR	Upper limit of private area
12-15	C-F	SCBLSADR	Lower limit of shared area
16-19	10-13	SCBVSTO	Virtual address of segment table
20	14	SCBRSTO	Segment table origin for DAT
		SCBSTL	(length of segment table)/64-1
21-23	15-17		reserved
24-55	18-37	SCBPSTR	PIK list of allocated partitions

MACHINE AND CHANNEL CHECK CONTROL BLOCKS

Machine/Channel Check Control Block Relationship



RAS Linkage Area (RASLINK)

Bytes Dec	Bytes Hex	Label	Description
0-3	0-3	CPUIDW1	First part of CPUID field
4-7	4-7	CPUIDW2	Second part of CPUID field
5	5	CPUID	Model number in CPUID field
6	6	RASMCELL	Length of machine check extended logout area
8	8	RASDMC	Damaged channel ID
9	9	RASFLAGS	RAS flag byte
		RASACT	X'80' RAS task activated
		RASMCACT	40 Machine check handling
		RASCCACT	20 Channel check handling
		RASEMGEX	10 Emergency handling
		RASSTERM	08 System termination
			04 Reserved
		RASNORTY	02 Retry not possible
		RTAIOA	01 RAS task I/O active
10	A	MCFLAGS	Machine check flags
		MCHARD	X'04' Hard machine check
11	B	RASRSFLG	RAS recording status flag
		RASNOFCH	X'80' Fetch of R-transient fails
			40 Reserved
		RASNOMSG	20 Unrecoverable channel check on SYSLOG
			10 Reserved
			08 Reserved
		RASBTDEQ	04 BTAM dequeue request
		RASMSGRT	02 Return from RAS message writer
		RASMSGIO	01 RAS message I/O
12-15	C-F	RASTABA	Address of RAS monitor table (RASTAB)
16-19	10-13	RASBASE	RAS base address
20-21	14-15	RASIMOD	Internal model number
22-23	16-17	RASIOELL	Length of I/O extended logout area
24-27	1C-1F	RASMCELA	Address of machine check extended logout area X'80' Indicates field contents not valid

RAS Monitor Table (RASTAB)

Bytes Dec	 Hex	Label	Description
0-3	0-3	LD00SLOT	$$RAST00 communication bytes
4-7	4-7	LD01SLOT	$$RAST01 communication bytes
8-11	8-B	LD02SLOT	$$RAST02 communication bytes
12-15	C-F	LD03SLOT	$$RAST03 communication bytes
16-19	10-13	LD04SLOT	$$RAST04 communication bytes
20-23	14-17	LD05SLOT	$$RAST05 communication bytes
24-27	18-1B	LD06SLOT	$$RAST06 communication bytes
28-31	1C-1F	LD07SLOT	$$RAST07 communication bytes
32-35	20-23	LD08SLOT	$$RAST08 communication bytes
36-39	24-27	LD09SLOT	$$RAST09 communication bytes
40-43	28-2B	LD10SLOT	$$RAST10 communication bytes
44-47	2C-2F	LD11SLOT	$$RAST11 communication bytes
48-51	30-33	LD12SLOT	$$RAST12 communication bytes
52-55	34-37	LD13SLOT	$$RAST13 communication bytes
56-59	38-3B	LD14SLOT	$$RAST14 communication bytes
60-63	3C-3F	LD15SLOT	$$RAST15 communication bytes
64-67	40-43	LD16SLOT	$$RAST16 communication bytes
68-71	44-47	LD17SLOT	$$RAST17 communication bytes
72-75	48-4B	LD18SLOT	$$RAST18 communication bytes
76-99	4C-63		reserved
100-103	64-67	LD25SLOT	$$RAST25 communication bytes
104-115	68-77	RASCCB	RAS CCB
116-147	78-97	RASCCWS	RAS CCW chain
148-154	98-9E	RASEEK	Seek address of RAS seek
155	9F	RTAOWN	R-transient identifier
156-157	A0-A1	MCPIK	Index to PIB active at machine check time
158-159	A2-A3	MCTIK	Index to TIK active at machine check time
160-163	A4-A7	ERPIBA	Address of work ERPIB
164-167	A8-AB	CCENTADR	Address of channel check routine
168	AC	RTAID	Requestor ID for RTA I/O
		RASRECID	X'08' RAS recording request
		RASRTYID	X'04' Channel retry request
169	AD	ERPID	Return load index for WTOR
170-171	AE-AF	RASRES	Device address of SYSRES
172-173	B0-B1	RASREC	Device address of SYSREC
174-175	B2-B3	RASLOG	Device address of SYSLOG
176-243	B4-F3	TRANSAV	RTA register save area, Register 0 to Register 15
244-307	F4-133	SYSREGS	RAS monitor register save area, Register 0 to Register 15
308-311	134-137	SUPLINK	Service routine address for RTA in RAS monitor
308	137	LINKFLAG	Flag byte indicat. requested service
		RASLIO	X'80' Perform normal I/O
		RASLEMIO	40 Perform emergency I/O
		RASLFTCH	20 Fetch another transient
		RASLWAIT	10 Perform wait
		RASLPDEQ	08 Dequeue page frame
		RASLDEQ	04 Dequeue CCB/IORB
		RASLFREE	02 Free I/O extended logout area
		RASLTIME	01 Get timer value for RTA
		RASLEXIT	00 Exit from RAS transient
312-323	138-13F	HIR	Hardware instr. retry accumulator
312-313	138-139	HIRACNT	Accumulated HIR count
314-315	13A-13B	HIRLCNT	Threshold value for count
316-319	13C-13F	HIR1TME	Time of day for first error of group
320-323	140-143	HIRLTME	Time threshold value in timer units
324-335	144-14F	ECCMAIN	Main storage error accumulators
324-325	144-145	ECCACNT	Accumulated ECC count for main stor.
326-327	146-147	ECCLCNT	Threshold value for count
328-331	148-14B	ECC1TME	Time of day for first error of group
332-335	14C-14F	ECCLTME	Time threshold value in timer units

RAS Monitor Table (RASTAB) (cont...)

Bytes			
Dec	Hex	Label	Description
336	150	MCMODE	Hardware operation mode
337	151	BUFDEL	Count of buffers deleted
338	152	RASMSG1	RAS Message byte 1
		MTICLDMG	X'10' Clock and or timer damage
		MTIMDMG	08 Timer damage
		MECQUIET	04 Contr. stor. ECC in quiet mode
		MPERFDEG	02 System performance degradation
		MEFLOVFL	01 EFL overflow
339	153	RASMSG2	RAS Message byte 2
		MCLOKDMG	X'80' Clock damage, all modes quiet
		MLASTTR	40 Threshold on rec-file reached
		MPAGEDEL	20 Buffer pages deleted
		MHIR	10 Soft MCI disabled
		MECC	08 ECC MCI disabled
		MFILEFL	04 Recorder file full
		MUNRCIO	02 Error on recorder file
		MCRECOV	01 Successful recovery from machine check
340-341	154-155	RASIND	RAS indicators
		RASNODEQ	X'80' Page frame not dequeued
342-343	156-157		Reserved
344-347	158-15B	RASPFT	Page frame table pointer
348-356	15C-164	INTERSEG	Interface segment build area
348-350	15C-15E	ILOGADR	Address of logout
351	15F	INOFN	Sequence number: record one of n
352	160	ILOGL	Logout length in record one
353	161	IRECL	Total length of record one
354	162	NNOFN	Sequence number record n of n
355	163	NLOGL	Logout length in record n
356	164	NRECL	Total length of record n

Error Recovery Procedure Information Block (ERPIB)

Bytes			
Dec	Hex	Label	Description
0–7	0–7	ERPIBCSW	Saved CSW
0	0	ERPIBSTC	ERPIB status codes
		ERPIBFRE	X'FE' Indicate free ERPIB
		ERPIBCNC	X'FD' Indicate task is to be canceled
		ERPIBCCR	X'FC' Indicate retry unsuccessful
		ERPIBCCS	X'FB' Indicate retry successful
0–3	0–3	ERPIBCCW	Address of failing CCW + 8
4	4	ERPIBST1	First status byte
5	5	ERPIBST2	Second status byte
6–7	6–7	ERPIBCNT	Residual count in CSW
8–11	8–B	ERPIBIOE	Pointer to corresponding I/O extended logout area
12	C		Reserved
13	D	ERPIBDMC	Damaged channel ID
14–15	E–F	ERPIBPUB	PUB address of failing device
16	10	ERPIBCQP	Channel queue pointer from the PUB
17	11	ERPIBRTC	RAS retry counter
18	12	ERPIBMSG	Message indicator
		ACTMSG	X'80' Wait for operator response
		CCDONE	40 Channel check handling complete
		CCNODEQ	20 PUB not queued in error
			10 – 04 Reserved
		RECCC	03 Recovered channel check
		ERRCC	02 Channel check
		HRDCC	01 Unrecoverable channel check
19	13	ERPIBREQ	Requestor ID
20	14	ERPIBFLG	Flag byte
		CCSIO	X'80' Channel check on SIO
		CCDAM	40 Channel damage
			20 – 10 Reserved
		CCREC	08 Record build or written
			04 Reserved
		CCDSK	02 Channel check on disk device
		CCSKM	01 Skip message writer
21–23	15–17	ERPIBESW	Extended CSW
24	18	ERPIBEND	X'FF' End of ERPIB

Track Hold Table (THTAB)

Bytes			
Dec	Hex	Label	Description
0	0	THPTR	Index of next entry in the forward chain pointer (X'FF' = last entry)
1–3	1–3	THCCB	Address of CCB/IORB
4–11	4–B	THTRK	CKD dev.: Addr. of track (BBCCHHOO) FBA devices: Physical block numbers of first and
12	C	THBWPTR	Index of previous entry in the chain (backward pointer)
13	D	THFLG	Flag and count byte
			X'80' Another task is waiting for this track/block
			40 First entry within a PUB chain
			20 –10 Reserved
		THCTR	Bits 4–7: Number of concurrent holds – 1
14–15	E–F	THTID	Task ID of track/block owner

Bytes 76-79 (X'4C' - X'4F') of the SYSCOM contain both, the free list pointer and the address of the Track Hold Table.

LOCK MANAGEMENT AREAS (DTLADR, LOCKADR, LOKOADR, DLFADR)

Define the Lock (DTLADR)
LOCKTAB Entry (LOCKADR)
Owner Element (LOKOADR)
DASD Sharing Dsect (DLFADR)

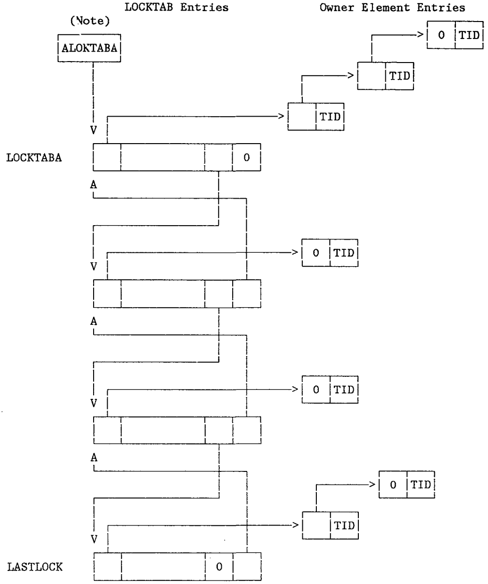
Define the Lock (DTLADR)

On entry to SVC 110 (X'6E') register 1 contains the address of DTL.

Bytes			
Dec	Hex	Label	Description
0 - 1	0 - 1	DTLLENF	Length of DTL
2	2	DTLFLG1	Flag Byte 1:
			CONTROL option
			X'80' Reserved
			X'40' Reserved
			X'20' Reserved
		DTLEXC	X'10' CONTROL=E(xclusive)
			LOCKOPT option
			X'08' Reserved
		DTLOPT4	X'04' LOCKOPT=4
		DTLOPT2	X'02' LOCKOPT=2
		DTLOPT1	X'01' LOCKOPT=1
3	3	DTLFLG2	Flag Byte 2:
		DTLKEEP	X'80' KEEP=YES
		DTLPART	X'40' OWNER=PARTITION
		DTLREDC	X'20' CHANGE=ON
		DTLEXTR	X'10' SCOPE=EXT
		DTLVOL	X'08' VOLID specified
			X'04' Reserved
			X'02' Reserved
			X'01' Reserved
4 - 15	4 - F	DTLNAME	Resource Name
16 - 21	10 - 15	DTLVOLID	Volume Identification
22	16	DTLLEN	Length of DTL

LOCKTAB Er try (LOCKADR) and Owner Element (LOKOADR)

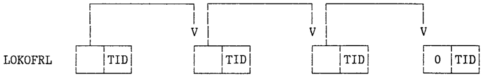
Relationship Between LOCKTAB and Owner Elements



Note: Identified via eye catcher 'LOCKSP' + 8 in pageable part of the supervisor.

Free-list of Owner Elements

If an owner element is freed, it will be put in front of the free-list.



LOCKTAB Entry (LOCKADR)

Bytes			
Dec	Hex	Label	Description
0 - 3	0 - 3	LOCKCHN	Chain pointer to Owner elements
4 - 15	4 - F	LOCKRESN	Resource Name
16	10	LOCKFLG1	Flag Byte 1:
			CONTROL option
			X'80' Reserved
			X'40' Reserved
			X'20' Reserved
		LOCKEXC	X'10' CONTROL=E(xclusive)
			LOCKOPT option
			X'08' Reserved
			X'04' LOCKOPT=4
			X'02' LOCKOPT=2
			X'01' LOCKOPT=1
17	11	LOCKFLG2	Flag Byte 2:
		LOCKUSED	X'80' LOCKTAB entry in use
		LOCKPART	X'40' LOCK owned by partition
		LOCKWAIT	X'20' Task waits for resource
		LOCKEXT	X'10' Cross system lock
			X'08' Reserved
			X'04' Reserved
			X'02' Reserved
			X'01' Reserved
18 - 19	12 - 13	LOCKCNTE	Number of exclusive users
20 - 23	14 - 17		Reserved
24 - 27	18 - 1B	LOCKPTR	Forward chain pointer
28 - 31	1C - 1F	LOCKBPTR	Backward chain pointer
32	20	LOCKLEN	Length of LOCKTAB Entry

Owner Element (LOKOADR)

Bytes			
Dec	Hex	Label	Description
0 - 3	0 - 3	LOKOCHN	Chain pointer to next Owner Element
4 - 5	4 - 5	LOKOTID	Task Identification of owning task
6 - 7	6 - 7	LOKOCNTS	Number of shared users
8 - 9	8 - 9	LOKOCNTE	Number of exclusive users
10	A	LOKOFLG	Flag Byte:
		LOKOKEEP	X'80' Keep until end of job
			X'40' Reserved
			X'20' Reserved
		LOKOEXC	X'10' Exclusive usage
11	B	LOKOFLG2	Flag Byte 2
12 - 15	C - F		Reserved
16	10	LOKOLEN	Length of Owner Element

DASD Sharing Dsect (DLFADR)

Bytes Dec	 Hex	Label	Description
.... First 20 bytes of lock file header record			
0 - 1	0 - 1	DLFCHAR	Block identification
2 - 3	2 - 3	DLFNCPUS	Number of sharing CPUs
4 - 5	4 - 5	DLFLBLK	Physical block length
6 - 7	6 - 7	DLFNBLK	No. of physical blocks in data area
8 - 9	8 - 9	DLFNENT	No. of entries per block
10 - 11	A - B	DLFLENT	Length of one lock entry (12+NCPU)
12 - 13	C - D	DLFBLKLL	Lower limit on FBA
		DLFCYL	Cyl. address of external file (CKD)
14 - 15	E - F	DLFREC#	Number of blocks per track (CKD)
16 - 17	10 - 11	DLFTRCK#	Number of tracks per cylinder (CKD)
18	12	DLFDEVT	Flag - device type
		DLFRPS	X'03' External file on RPS CKD
		DLFCKD	02 External file on CKD
		DLFFBA	01 External file on FBA
19	13	DLFDEVC	Device code
.... Start of 8 byte CPU field			
20	14	DLFCPUS	Start of 8 byte CPU field
		DLFCPUF1	Flag byte 1 in CPU entry
		DLFCPUUS	X'80' CPU field in use
20 - 21	14 - 15	DLFUNT	Channel and unit of external file
22 - 23	16 - 17	DLFPUB	PUB index (for physical addressing)
24	18	DLFFLG1	Flag - byte 1
		DLFINT	X'80' DSHRINIT processed successful
		DSHRDOWN	40 DASD sharing support down (I/O error)
		DLFACT	20 DASD sharing support is active
		DSDWNMSG	10 DASD-SHR-DOWN message to be displayed
		DLFCHAIN	08 write chained to device release
25	19	DLFFLG2	Flag - byte 2 (reserved)
26 - 27	1A - 1B	DLFINDEX	Number of this CPU (0 until NCPU-1)
.... End of IPL DLF table			
28	1C	DLFLENI	Length of DLF table (for IPL)
....			
28 - 31	1C - 1F	DLFAREA	I/O area for external file
32 - 33	20 - 21	DLFHBLK	Actual block in lock file (hash no.)
34	22	DLFLEN	Length of DLF table (full length)

```
* This DSECT is used to address the DASD sharing interface table
* (DLFTABLE).
* It is used by the lock manager and the IPL routines.

* The first part (fields DLFCHAR till DLFEOVC) describes the
* external file itself. It contains information which is
* valid for any sharing CPU.
* The second part (DLFUNT till DLFHBLK) contains information
* specific to the various sharing CPUs .

* Note: The header record of the external file contains the
        first part of the DLFTABLE followed by the 8-byte
        CPU-fields (each consisting of the flag bytes and
        a 6-byte CPU-identification).
```

DISK INFORMATION BLOCK (DIB) TABLES

Disk Information Block (DIB) Table for CKD and DISKETTE Devices

Current	Length of		End of Extent	Head No.		Max	Notify	Flag	
Record Addr.	Key	Data	Address	High	Low	Rec.	Rec.No.		
0 6	7	8 9	10 16	17	18	19	20 21	22	23

Byte(s)	Description
0 CURRENT RECORD ADDRESS	Specifies the disk address of the next sequential record. The format differs slightly depending on the file and the device. For CKD-devices: SYSIN : BBCCHHR, SYSPCH : BBCCHHR, SYSLST : BBCCHHR For DISKETTE-devices: SYSIN : 0000CHR, SYSPCH : 0000CHR, SYSLST : 0000CHR
7 KEY LENGTH	Always zero
8 DATA LENGTH	Data length of record to be processed For CKD-devices: SYSIN : X'0050' or X'0051', SYSPCH : X'0051', SYSLST : X'0078' For DISKETTE-devices: SYSIN : X'0000', SYSPCH : X'0000', SYSLST : X'0000'
10 END of EXTENT ADDRESS	Specifies the disk address of the last record within the given extent. For CKD-devices: SYSIN : BBCCHHR, SYSPCH : BBCCHHR, SYSLST : BBCCHHR For DISKETTE-devices: SYSIN : 0000CHR, SYSPCH : 0000CHR, SYSLST : 0000CHR
17 HIGHEST HEAD NO.	Highest head number accessible on this device.
18 LOWEST HEAD NO.	Lowest head number accessible on this device.
19 MAXIMUM NO. of RECORDS	Maximum number of records that fit on one track.
20 NOTIFY RECORD NUMBER	This field specifies the number of records that the user wants to be checked at EOJ time of whether they still fit into the specified extent (applicable for output only). This field is set by the JCL SET statement (RCLST or PCPCH). A warning message will be issued when this minimum number has been reached or exceeded during the previous JOB.
22 FLAG BYTE	Flag byte: X'40' Device with RPS feature
23 RESERVED	Not used
Note: The DIB is initialized by Job Control with Extent Info. and updated by PIOCS on every I/O oper. to the appropriate device.	

Bytes 96-97 (X'60' - X'61') of the Partition Communication Region contain the address of the DIB Table.

Disk Information Block (DIB) Table for FBA Device

Bytes Dec	Hex	Label	Description
0-3	0-3	ULPBN	End address of extent. Upper limit of physical block number
4-7	4-7	CRPBN	Current address. Current physical block number
8-9	8-9	CIOFF	Offset of current record within control interval
10-11	A-B	LNGCI	Length of control intervals in bytes
12	C	PBPERCI	Number of physical blocks per control interval
13-15	D-F	PBUFFER	Pointer to data buffer
16	10	DIBFLAGS	X'80' DIB gate flag X'40' Task waiting for DIB X'20' Reserved X'10' Source begin readjustment required X'08' Reserved X'04' Force write out X'02' End of extent reached X'01' Buffer-in-use flag
17-19	11-13	PDIBX	Pointer to DIB extension (DIBX)
20-21	14-15	DIBRSCNT	Residual count for JCL message
22-23	16-17		Reserved

DIB Extension Table (DIBX) for FBA Devices

The FBA device also requires a DIB Extension (DIBX) Table.

Bytes Dec	Hex	Description
0-23	0-17	Input Output Request Block (IORB)
24-31	18-1F	Fixlist first area
32-39	20-27	Fixlist second area
40-47	28-2F	DEFINE EXTENT CCW
48-55	30-37	LOCATE CCW
56-63	38-3F	READ/WRITE CCW
64-79	40-4F	DEFINE EXTENT Parameter list
80-87	50-57	LOCATE Parameter list

ERBLOC Area

Bytes Dec	Hex	Label	Description
0–7	0–7	SVC5NM	Name of first/next ERP Transient to be fetched
8–11	8–B	YRETRY	Continuation address for retry I/O request (INITRG)
12–15	C–F	YIGNORE	Continuation address to ignore I/O error (IGNORE)
16–19	10–13	ACANCEL	Continuation address to cancel I/O request (ERR1A)
20–23	13–17	YERPEXIT	Common DSK/ERP return address (ERPEXIT)
24–75	18–4B	ERQ1	Area to pass recovery and recording information to the ERP. Its layout is the same as for a single error block, except for the 8–byte header (see Note)
76–111	4C–6F	SNSSDAID	Sense data saved by SDAID
112–119	70–77	ERCHNOFT	Chain header offset table, used to address the following error chains
120–123	78–7B	RASERCHN	Address of first RAS error entry
124–127	7C–7F		Pointer to RAS TIB
128–131	80–83	ERPERCHN	Address of first ERP error entry
132–135	84–87		Pointer to ERP TIB
136–139	88–8B	DSKERCHN	Address of first DSK error entry
140–143	8C–8F		Pointer to DSK TIB
144–147	90–93	SNSERCHN	Address of first SNS error entry
148–151	94–97		Pointer to SNS TIB

Bytes 0-3 (X'00 - X'03') of the System Communication Region contain a pointer to the ERBLOC area.

There is one I/O error block for each device. Field PBXERBLK in the PUBX contains a pointer to this block.

An additional error block exists for some system tasks. The address of this block is contained in field TCBERBLK of the system task TCB.

XPCC CONTROL BLOCKS (IDCB, CRCB)

Identification Control Block	(IDCB)
Connect Request Control Block	(CRCB)

Identification Control Block (IDCB)

Bytes			
Dec	Hex	Label	Description
0- 3	0- 3	XPIDPT	Pointer to next ID-CB
4- 7	4- 7	XPICRPT	Pointer to first CR-CB
8- 9	8- 9	XPIPART	Offset to that part of CR-CB which belongs to current application
10-11	A- B	XPITID	TID of ID-CB owner
12-13	C- D	XPIMTID	TID of corresponding maintask
14-21	E-15	XPIMTID	Identification key (token)
22-29	16-1D	XPIAPPL	Application name
30-31	1E-1F	XPICRQS	Number of requested connections
32-33	20-21	XPICNTR	Number of open connections
34	22	XPIFLG1	Flag byte
		XPISUBS	X'80' IBM-subsystem
		XPITMQ	40 Application issued TERMQSCE
			20 Reserved
			10 Reserved
			08 Reserved
			04 Reserved
			02 Reserved
			01 Reserved
35	23	XPIFLG2	Flag byte (reserved)
36	24	XPIDEND	Length of IDCB

Bytes			
Dec	Hex	Label	Description
0- 3	0- 3	XPZTCBC	TCB chain pointer
4-11	4- B	XPZCRTK	Path-id (connection request token)
12-15	C- F	XPZBUFAD	SEND buffer address
12	C		X'80' Last buffer of a list
16-19	10-13	XPZBUFLN	Buffer length
20-67	14-43		max. 7 entries in list
68-71	44-47	XPZTOTAL	Total buffer length
72-75	48-4B	XPZREPLY	Address of reply area
76	4C	XPZFLAG	Flag in user area
77-79	4D-4F	XPZRPYLN	Reply area length
80-87	50-57	XPZUSER	User data
88-93	58-5B	XPZSPACE	SCB pointer of partner
92	5C	XPZFCT	Function code
93	5D	XPZFLG1	Flag byte
		XPCONCL	X'80' Connection is completed
		XPCONBSY	40 Connection is busy
		XPCINTCB	20 In TCB chain
			10 - 04 Reserved
		XPZCONPE	02 Connection exit pending
		XPZRPOST	01 Post at receive after SENDR
94	5E	XPZFLG2	Flag byte (reserved)
		XPTERMAB	X'80' Other side terminated abnorm.
		XPTERMNO	40 Other side terminated normally
		XPDSCED	20 Other side disconnected
			10 - 04 Reserved
		XPINOVM	02 Partner in other VM machine
		XPCURRSP	01 Both part. in current space
95	5F	XPZREAS	Reason code
96	60	XPZCEND	Length of Common Part
.... Block of First Communication Partner			
96- 99	60-63	XPZNXTCR	Pointer to next CR-CB
100-101	64-65	XPZPART	Offset to that part of CR-CB which belongs to current application
102-103	66-67	XPZTID	TID of connect owner
104-107	68-6B	XPZPCCB	XPCCB address
108-111	6C-6F	XPZIDADR	Address of corresponding IDCB
112-119	70-77	XPZTOAP	To-Application name
120	78	XPZFLG3	Flag byte
		XPSEND	X'80' SEND pending
		XPSENDR	40 SENDR pending
		XPCLEAR	20 Sender cleared request
		XPRECVE	10 Receive after SENDR executed
			08 - 01 Reserved
121-123	79-7B		Reserved
.... End of First Part			
124	7C	XPZFEND	Len. of First Partner + Common Sect.
.... Block of Second Communication Partner			
124-127	7C-7F		Pointer to next CR-CB
128-129	80-81		Offset to that part of CR-CB which belongs to current application
130-131	82-83		TID of connect owner
132-135	84-87		XPCCB address
136-139	88-8B		Address of corresponding IDCB
140-147	8C-93		To-Application name
148	94	XPZFLG3#	Flag byte
			X'80' SEND pending
			40 SENDR pending
			20 Sender cleared request
			10 Receive after SENDR executed
			08 - 01 Reserved
149-159	95-9F		Reserved
160	A0	XPZREND	Total Length of CRCB

PDTABB AND PDTABA TABLES

Bytes 126 and 127 (X'7E'-X'7F') of the partition communication region contain the address of the Paper Document processing Table.

Label PDTABB identifies the first byte of the table. The tables are used for handling external interrupts on magnetic ink or optical character recognition devices.

PDTABBB contains six 8-byte entries; one for each line of the direct control feature on the system.

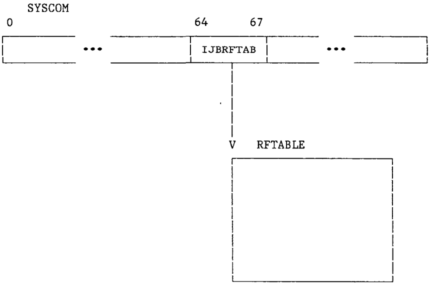
Table for MICR DTF Addresses Entries (PDTABB)

BYTE		AND INSTRUCTION		OWNER	DTF ADDRESS for MICR
DEC	HEX				
0	0	NI	PDSTAT+1,X'FE'	TID	Device on LINE 7
8	8	NI	PDSTAT+1,X'FD'	TID	Device on LINE 7
16	10	NI	PDSTAT+1,X'FB'	TID	Device on LINE 7
24	18	NI	PDSTAT+1,X'F7'	TID	Device on LINE 7
32	20	NI	PDSTAT+1,X'EF'	TID	Device on LINE 7
40	28	NI	PDSTAT+1,X'DF'	TID	Device on LINE 7

Bytes	Description	
0 – 3	The NI instruction is executed in the external Signal Interrupt handler to turn off the external line status as soon as this line interrupt is being processed (any other external line signal remains affective). PDSTAT+1 is the fixed main STORAGE location 135 (X'87') and contains the external signal codes that have not yet been processed in Bits 2–7.	
	Bits	Description
	7	External signal from line 7
	6	External signal from line 6
	5	External signal from line 5
	4	External signal from line 4
	3	External signal from line 3
	2	External signal from line 2
4	Contains the PIK of the partition containing the DTF	
5 – 7	Contain the address of the DTF table	

Note: The contents of PDSTAT+1 (bits 2-7) is used to index a one byte entry in table PDTABA which, in turn indexes the DTF address entry, whithin table PDTABB of the external signal line with the currently highest priority. (Line 2 has highest, line 7 lowest priority).

Recorder File Table Relationship



Recorder File Table (RFTABLE)

Bytes Dec	 Hex	Label	Description
		RFTABLE	Label of Starting Address
0	0	RFFLAGS1	Flag byte 1
		RFFULL	X'80' File full
		RFRDE	40 RDE option included
		RFIPL	20 Initial IPL
		RFNO	10 RF=No option
		RFCREATE	08 File is to be created
		RFBUILT	04 File has been created
		RFONFBA	02 File on FBA device
		RFREADY	01 File ready
1	1	RFFLAGS2	Flag byte 2
		FFMSG	X'80' File full message request
		LTMSG	40 Last track message request
		IEMSG	20 I/O error message request
		DLMSG	10 Data lost message request
		RFEVA	08 EVA message request
		RFRTAOWN	04 File owned by RTA recorder
		RFPTAOWN	02 File owned by PTA recorder
		RFEREP	01 File being accessed by EREP
2	2	RFFLAGS3	Flag byte 3
		LTMISSUD	X'80' Last track msg issued once
		RECDERR	40 Error is to be recorded
		RECDSF	20 Short form record request
		RFIRULT	10 Individual records for unlabeled tapes
			08 Reserved
		RFHIOERR	04 Error in writing RFHEADER
		RFBOMT05	02 Exit to $$BOMT05 indicator for $$BOPEN
		RFBOMT01	01 Exit to $$BOMT01 indicator for $$BOPEN
3	3	RFFLAGS4	Flag byte 4 X'80' - X'02' Reserved
		RFRNW	01 No record written
4	4	RFFLAGS5	Flag byte 5 X'80' - X'02' Reserved
		RFFLG5BD	01 BOPEND called by OPEN
5	5	RFNOFN	N of N for records (low order 4 bits contain the number of records to be recorded and high order 4 bits cont. the nbr of the record being recorded)
6	6	RFRECTYP	Record type code
7	7	RFREL	Release level code of VSE/Adv.Funct.
8	8	RFRDSW1	Record dependent bit 1
		RFTEMP	X'40' Temporary error
9	9	RFRDSW2	Record dependent bit 2
10-11	A- B	RFBUFLG	Length of data buffer (FBA)
.... CKD Device Related Information			
12-13	C- D	RFMCONST	Multiplier for track balance
14-15	E- F	RFDCONST	Divisor for track balance
16-17	10-11	RFOCONST	Overhead for track balance
18-19	12-13	RFRECLEN	Length of record
20	14	RFRDSW3	Record dependent switch 3
21-23	15-17		Reserved
24-27	18-1B	RFRECADR	Address of record
28-34	1C-22	RFSEEK	Work area for seek addr.BBCCHHR
28-29	1C-1D	RFSEEKBB	BB portion of seek
30-31	1E-1F	RFSEEKCC	CC portion of seek
32-33	20-21	RFSEEKHH	HH portion of seek
34	22	RFSEEKR	R portion of seek
35	23	RFEREPK	Key of EREP partition
36-39	24-27	RFHDRCH	SYSREC cylinder/head
36-37	24-25	RFHDRCYL	Cyl. address of file start
38-39	26-27	RFHDRTRK	Head address of file start
.... End of CKD Device Related Information			

Recorder File Table (RFTABLE) (cont...)

Bytes Dec	Hex	Label	Description
.... FBA	Device Related	Information	....
12-15	C- F	RFBUFAD	Address of data buffer
16-17	10-11	RFNAVR	Displacement of next available RDF in buffer (FBA)
18-19	12-13	RFRECLEN	Length of record
20	14	RFRDSW3	Record dependent switch 3
21-23	15-17		Reserved
24-27	18-1B	RFRECADR	Address of record
28-31	1C-1F	RFCUBL	Work area for block number
32-34	20-22		Reserved
35	23	RFEREPK	Key of EREP partition
36-39	24-27	RFHDRBL	SYSREC block number
.... End	of FBA Device	Related	Information ....
40-41	28-29	RFCHMAP	Map of supported channels
42-49	2A-31	RFCHIDC	Channel ID codes
50	32	RFRDSW0	Record dependent switch 0
51	33		Reserved
52-55	34-37	RFEXIT	Exit phase name or exit address
56	38	RFEVARTH	EVA read threshold
57	39	RFEVAWTH	EVA write threshold
58-59	3A-3B	RFP2ENTL	Length of PUB2 table
60-63	3C-3F	RFP2ENT	Address of PUB2 table
64-..	40-..	RFP2ITAB	PUB2 index table (see Note)

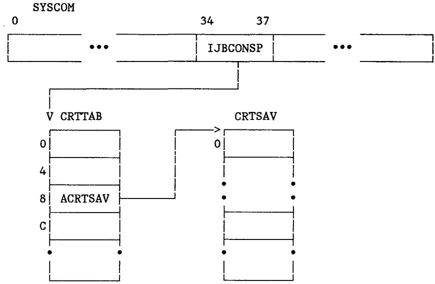
Note: Two bytes are generated for each PUB2 index entry.

Console Buffer Table (CBTAB)

Label CBTAB identifies the first byte of the Console Buffer Table. Label CBNEXT points to the next free entry within this table.

Bytes Dec	Hex	Label	Description
0-7	0-7	CBCCW	CCW: Command code, chain byte and count have been copied from the user's CCW. The data address is always the addr. of CBDATA (byte 24-103 see below).
8-23	8-17	CBCCB	CCB: An area in CCB format whose CCW address field always points to CBCCW (see bytes 0-7 above).
24-103	18-67	CBDATA	Console Buffer: An output area in which the user's data is kept
• • •		• • •	• • •

CRT AREAS (CRTTAB, CRTSAV)



CRT Constant Table

Bytes Dec	Bytes Hex	Label	Description
0-7	0-7	CRTNAME	Name of CRT routine
8-11	8-B	ACRTSAV	Address of CRT save area (CRTSAV)
8	8	CRTNAM1	Phase Identifier
			Last character of phase that is
			to regain control after Attention
			Interrupt or I/O error are processed.
8	8	SENSEBT	Dummy sense byte
			X'80' Command reject
			40 Operator intervention required
			20 Reserved
			10 Equipment check
			08-02 Reserved
			01 Operation check
12-15	C-F	ACRTTRNS	Address of C-Transient area (CRTTRNS)
16-19	10-13	AATTNINT	Address of Attention interface rout.
20-23	14-17	ACRTUNPS	Address of CRT deactivation routine
24-27	18-1B	ACRTNWSO	Service owner of CRT
28	1C	CRTFLG1	Flag byte 1
		CRTERPBT	X'80' ERP message
		CRTUNITC	40 Unit check for CRT SYSLOG requ.
		CRTFETCH	20 Fetch of $$BOCRTK is in progr.
		CRTATTH	10 Device end simulated
		CRTERADR	08 Validation error
		CRTREDSP	04 Redisplay in progress
		CRTERR	02 CRT I/O error
		CRTBUSY	01 CRT busy
29	1D	CRTFLG2	Flag byte 2
			X'80' Reserved
		CRTSENS	40 Sense Byte (see byte 8) was
			set up by CRT
		CRTEOJ	20 End of CRT routines
		CRTDATRD	10 Data already read
		CRTATTPD	08 Attention Interrupt pending
		CRTRQPD	04 Request pending
		CRTATTRQ	02 Attention request being handled
		CRTEOJO	01 EOJ on CRT
30-35	1E-23	CRTEINF	CRT error information
36-39	24-27	AHCFIOMD	Entry address of I/O module
			for hardcopy file access

CRT Save Area (CRTSAV)

Bytes			
Dec	Hex	Label	Description
0	0	CRTSAV	CRT save area – Set on doubleword boundary
0–7	0–7	SAVOLDP	Save area for old SVC PSW
8–11	8–B	ACRTSAVA	Address of problem program save area
12–55	C–34	CSAVEAR	Channel scheduler save area
12–15	C–F	CRTSV1	save area for register 1
		CCBSAVAR	CCB address
16–19	10–13	CRTSV2	save area for register 2
20–23	14–17	CRTSV3	save area for register 3
24–27	18–1B	CRTSV4	save area for register 4
28–31	1C–1F	CRTSV5	save area for register 5
32–35	20–23	CRTSV6	save area for register 6
36–39	24–27	CRTSV7	save area for register 7
		CRTPSWM	Save area for CRT system mask
40–43	28–2B	CRTIOSB	save area for IOS base register
44–47	2C–2F	CRTINTER	Address of I/O interrupt routine
48	30	CRTCCBB1	save CCB communication byte 1
49	31	CRTCCBB2	save CCB communication byte 2
50	32	CRTCCBB3	save CCB communication byte 3
51	33	CRTATTRB	Message attribute byte
Constants needed for CCW processing			
56–63	38–3F	CRTCCW0	CCW for write screen control char.s
64–71	40–47	CRTCCW	CCW built by CRT routines
72–79	48–4F	CRTCCB	CCB modified by CRT routines
80–83	50–53		CCW address
84	54		Flag byte
85–87	55–57		CSW CCW address
88–95	58–5F		CCW
96–99	60–63	CRTSNSI	CRT sense information
100–103	64–67	ASUPSAVA	Address of SUP system task save area
104–107	68–6B	CRTNEXT	Next CCW to process
108–111	6C–6F	ACTLCCW	Address of actual CCW
112–115	70–73	CONTCCW	Address of cont. CCW
116–117	74–75	CONTRDSV	Save byte count of cont. CCW
118	76	ATTLENG	Length of attention input
119	77	CRTNAM2	Save area for CRT char. in error case
120–121	78–79	CRTUTID	TID of task requesting CRT
122–123	7A–7B	CRTUPIK	PIK of task requesting CRT
Constants needed for hard copy processing			
124	7C	CRTHCPIK	Translation PIK for cont.-lines
125	7D	CRTFLGHC	Flags for Hard copy file (HC)
		CRTHCOPN	X'80' HC opened
		CRTHCOVR	X'40' HC in overlay mode
		CRTHCWRN	X'20' Warning (2 tracks left) sent
		CRTIPL	X'10' HC IPL switch
		HFTOOPEN	X'08' HC must be created
		HFEQUNO	X'04' HC not in use
		HCERR	X'02' HC has unrecoverable error
		HCINCL	X'01' Incorrect length during HC disk I/O
126	7E	PRTLOCK	Lock for PRINTLOG function X'00' – open, X'FF' – closed
127	7F	HCFLG	Flags for Hard copy file support
		OVERLAY	X'80' HCF in overlay mode
		PRINTLOG	X'40' PRINTLOG no select active
		HCINCERR	X'20' Inconsistent state in HC-supp.
		NOTCMPLT	X'10' HC file not yet full
			X'08' Reserved
			X'04' Reserved
			X'02' Reserved
			X'01' Reserved

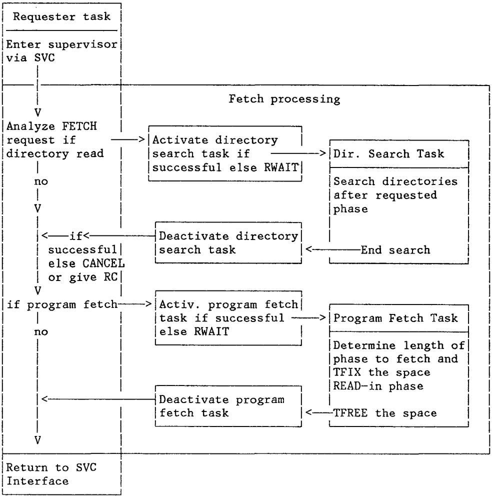
CRT Save Area (CRTSAV) (cont...)

Bytes			
Dec	Hex	Label	Description
HCFCB extension			
128–131	80–83	HCFCBWRT	Address of write HCFCB
132–135	84–87	HCFCBHDR	Pointer to HCFCB for write header
136–137	88–89	HCBOWNER	Owner of HC file
138	8A	HCFDEVTP	Device type (GETVCE output)
139	8B		Reserved
140–141	8C–8D	HCFBLKLN	Physical block length
142–149	8E–95	CWRPDADR	Addr. of last 'print logged' HCF rec.
150–155	96–9B	CWARNSKA	Disk address of warning message in HCF overlay mode
156–163	9C–A3	IPLDADR	Address of IPL–record on HCF
164–165	A4–A5	HCMSGLNG	Length of message 3277
166–167	A6–A7	HCFNRTR	Tracks/Cylinder
168–169	A8–A9	HCFNRBLK	Number of physical records/track
		HCFBLFBA	Block length of FBA device
170–177	AA–B1	HCCSW	CSW without 1st byte
178–201	B2–C9	HCSNS	HC file sense bytes
Constants used by CRT–redisplay feature			
204–205	CC–CD	PARTRED	Current partition redisplaying
206–207	CE–CF	PARTRED1	Partition id unchecked
208	D0	OCCFLG	Current OCCF options redisplay
209	D1	OCCFLG1	OCCF options specification unchecked
210–211	D2–D3	MSGACOO	Residual lines on screen
212–215	D4–D7	SCREENAD	Address of screen buffer save area
216–219	D8–DB	AHCFCBRD	Address of HCFCB for redisplay
220–221	DC–DD	LINEAL1	Line count 1. screen line – all msg.
222–223	DE–DF	LINECOA	Actual line counter
224–225	E0–E1	LINEPA1	Line count 1. screen line – partition
226–227	E2–E3	LINECOP	Actual line counter for selection
228–229	E4–E5	LINECNT	Line count indicated by command
230–231	E6–E7	LINECNT1	Line count unchecked command
232	E8	DISPF	Display flag
		BW	X'80' Actual reading is backward
		OCCFPAR	X'40' OCCF options specified
			X'20' Reserved
			X'10' Reserved
			X'08' Reserved
			X'04' Reserved
			X'02' Reserved
			X'01' Reserved
233	E9	REDISFLG	Communication redisplay routines
		SCRSAVE	X'80' Save current display
		SCRREST	X'40' Restore current display
		SCRRET	X'20' Return to start point
		SCRFW	X'10' Forward redisplaying
		PARCHG	X'08' Partition changed
		DISPCNT	X'04' Display content of part. line
		BYPSCOM	X'02' Bypass command checking
			X'01' Reserved
234	EA	FLG1	Communication byte command checking
		NOFRST	X'80' No first parameter indicated
		PARTPAR	X'40' partition parameter indicated
		DIRPAR	X'20' Direction parameter indicated
		RETURPAR	X'10' Return parameter indicated
		COUNTPAR	X'08' Count parameter indicated
		NOSEC	X'04' No second parameter indicated
		SCRFW1	X'02' Forward redisplay indicated
		ERRRET	X'01' Error return indicator
Temp. save area for SCT–pointer and screen buffer addr. used by $$BOCRTC and $$BOCRTD			
236–239	EC–EF	R3SAV	Save area for register 3
240–243	F0–F3	RDSAV	Save area for register D

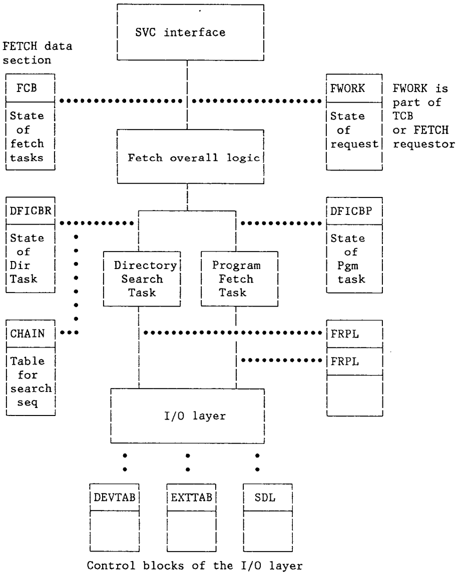
CRT Save Area (CRTSAV) (cont...)

Bytes Dec	Bytes Hex	Label	Description
Constants needed for screen management			
244-247	F4-F7	SEGVAL0	Auto. del. default for 3277
248-251	F8-FB	SEGVAL1	K-command default S/125
252	FC	ACTCCW	Actual CCW indicator for screen cmd.
253	FD	MSGIND	Message indicator in HEX
254-257	FE-101	SEGVAL2	Work segment value
258-259	102-103		Reserved
260-263	104-107	CRTPOS1	Position avail.for data in curr.line
264	108	CRTPOS2	
265	109	CRTPOS3	
266-282	10A-11A	DELTAB	Deletion table for ASY OC
283	11B		End of deletion table (X'0F)
284-286	11C-11E	POWERCUU	For PGO commands CUU is stored
287	11F		Reserved
288-259	120-127	CRTCCWS	CCW save area
End of Constants needed for screen management			
296-319	128-13F		Interphase communication flags
322-418	142-1A2	AUXTAB	Auxiliary screen description table
419-503	1A3-1F7	IOAREA	Hard copy file I/O area
504-623	1F8-26F	PRINTSNS	Support for 3284/86/87 printer
Layout for 3277 screen			
624-748	270-2EC	SCRNCTL	Screen control table
749-2619	2ED-A3B	SCRIMG	Buffer for screen image
2620-2667	A3C-A6B	CRTBUAD	Device buffer line addresses
2668-2749	A6C-ABD	TABASE	Device buffer line offsets
2750-3078	ABE-C06	BLKLNE	Line frames
3079	C07		Reserved
3080-3143	C08-C47	CRTMVCSA	Move routine save area

Fetch Control Flow (Part 1)


Requester task
Enter supervisor via SVC
Fetch processing
Analyze FETCH request if directory read
Activate directory search task if successful else RWAIT
Dir. Search Task
Search directories after requested phase
End search
no
Deactivate directory search task
if successful else CANCEL or give RC
if program fetch
no
Activ. program fetch task if successful else RWAIT
Program Fetch Task
Determine length of phase to fetch and TFIX the space READ-in phase
TFREE the space
Deactivate program fetch task
Return to SVC Interface

Fetch Control Flow (Part 2)



Fetch Concept in New Librarian

The new librarian supports a uniform and condense-free library concept. A New Library (NLIB) consists of a non-empty set of sublibraries each may contain members of various types like PHASE, MODULE, PROCEDURE etc. A sublibrary consists of a directory, alphamerically ordered after 'TYPE.MEMBERNAME', and a member space. It may have more than one extent on more than one volume of the same disk device type. For faster search algorithm, the directory can be accessed via an index set (B-tree).

The physical organization of the library is done into so called Library Blocks (LBs) of the size of 1K. The LBs are comparable to the CIs (Control Intervals) in VSAM. A LB contains the data record and VSAM like control information. This is called LBCF and consists of CIDF (Control Interval Definition Field), RDF (Record Definition Field), phase ID and LB chaining field. The next logical LB entity is addressed by the LB chaining field. In such a way the requirement of condense-freeness is satisfied.

As a consequence however, the contiguity of the directory and the space of an individual member cannot be guaranteed. In a frequently updated library respectively sublibrary the degree of fragmentation (directory-, index- and member-space) is increased during its lifetime. The resulting FETCH performance will be essentially decreased. A reorganization of the library is recommended for a proper FETCH performance.

Library Format

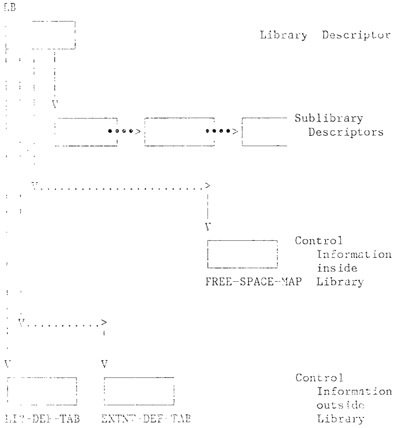
```
LB (Library Block)
|<------------------------ LB Size ---------------------------->|
┌───────────────────────────────────────────────────────────────┐
|/////////////////////////////|                      |//////////|
└───────────────────────────────────────────────────────────────┘
|<------- Data ------------->|<-- Free Space -->|<- LBCF ->|

LBCF (Library Block Control Field)
|<------------ Size = F(#records) ---------------------------------------->|
┌─────┬───┬─────┬────┬────┬────┬─────┬─────┬─────┬────┬─────┬─────┐
|L'RCn|...|L'RC1| ID |FLAG|#REC |L'REC|A'FSP|L'FSP|#CLB|BPRBA|FPRBA|
└─────┴───┴─────┴────┴────┴────┴─────┴─────┴─────┴────┴─────┴─────┘
|<------┬-------->|<------------ fixed length ------------------------>|
        |
        └── only for compressed or variable length records
```

applied only for directory and index LBs (not in TEXT and RLD LBs)

```
            xPRBA
            |<-2>|<--4->|
            ┌───────────┐
            |OFFS|RBLKNR|
            └───────────┘
            |<-->|<---->|
               |    |
               |    └──── block number relative to LIB-start
               └───────── offset in library block
```

F(#records):	function of number of records contained in the LB
L'RCn:	length of record number n (at least one record differs in length from the others)
ID:	phase ID
#REC:	number of records
L'REC:	length of records (if all records of same length)
A'FSP:	begin address of free space
L'FSP:	length of free space
#CLB:	number of contiguous LBs following this LB
BPRBA:	backward pointer RBA (relative byte and block address)
FPRBA:	forward pointer RBA of next logical LB
xPRBA:	FPRBA or BPRBA

Library Structure



'•' represents the logical LB chain pointers
'.' represents the logical chaining of data entities

Notes:

1. The control information tables are not necessarily located as physical fields in the library. They may be built during "Library Allocation" time by means of label information etc.

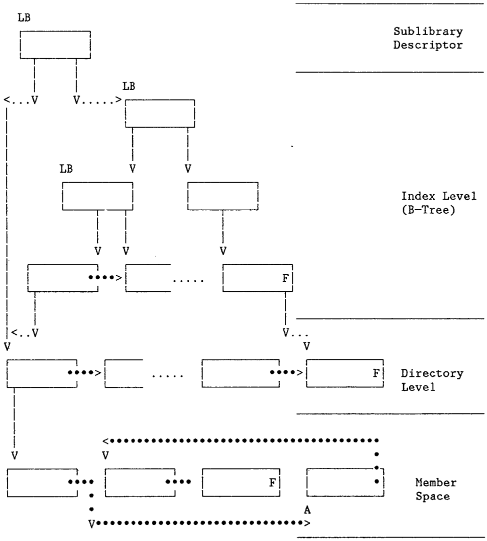
2. All directory LBs are on the same (lowest) index level and are alphamerically sorted after "TYPE.MEMBERNAME". The highest index level of a sublibrary consists of one or more LBs (performance considerations).

3. The data length of TXT, or RLD LBs is L'LB - L'LBCF.

4. The EOB indication for DIR or INDEX LBs is given by:
LBCF.L'REC = X'0'

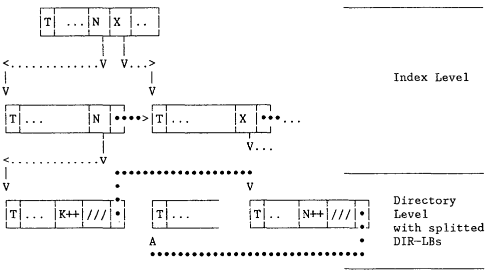
5. End of a logical chain (e.g. member, directory) is given by:
FBRBA = X'FFFFFFFFFFFF'.

Sublibrary Structure



'•' represents the logical LB chain pointers.
'.' represents the logical chaining of data entities.

Directory and Index

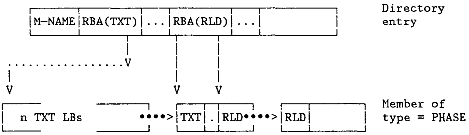


'•' represents the logical LB chain pointers.
'.' represents the logical chaining of data entities.

Note:

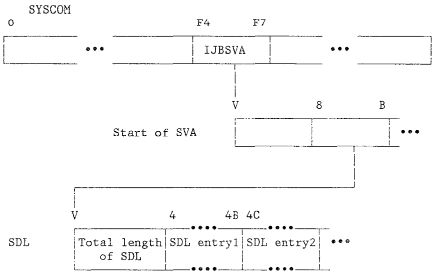
T:	TYPE entry
N:	Index entry
X:	Index entry
K++:	Directory entry
N++:	Directory entry

Library Member



'•' represents the logical LB chain pointers.
'.' represents the logical chaining of data entities.

How to Locate SDL Entries



SDL format of a directory entry

DEC	HEX	Label	Description
0	0	SDLESEG1	Directory Entry (DE) - common segment
0	0	SDLENAM	Member name
8	8		Reserved
9	9	SDLEDEF1	Attributes for DE (flag byte)
		SDLEETYP	X'80' Type of entry = type
		SDLEEHLX	40 Type of entry = high level index
		SDLEEDIR	20 Type of entry = directory
			10 - 01 Reserved
10	A	SDLEPRBA	PRBA of member
16	10	SDLECONT	Number of contiguous LBs
18	12		Reserved
20	14	SDLEPFL	User areal (type = PHASE)
20	14	SDLEFLG	Flags
		SDLEBSR	X'80' Self relocating phase
		SDLEBRL	40 Relocating phase
		SDLEBSE	20 SVA eligible
		SDLEBSV	10 Phase in SVA
		SDLEBPC	08 PCIL flag for incore directory
		SDLEBNF	04 Not found flag (incore directory)
		SDLEBAC	02 Entry active (incore directory)
			01 Reserved
21	15	SDLESWT	Switches
		SDLECLM	X'80' Set SDL: move mode phase
		SDLECLS	40 Set SDL: SVA eligible
			20 - 01 Reserved
22	16		Reserved
24	18	SDLEPLN	Length of phase(TXT) in bytes
28	1C	SDLELPT	Load point at link-edit time
32	20	SDLEENP	Entry point at link-edit time
36	24	SDLESTR	Partition start at link-edit time
40	28	SDLERLD	Number of RLD items
42	2A	SDLERLDA	PRBA of first RLD item if any, otherwise x'FF'
48	30		Reserved
56	38	SDLESVAP	Entry point in SVA if any, otherwise X'00'
60	3C	SDLEIDEN	Library block id
64	40	SDLEALIB	Address of LIB-DEF-TAB
68	44	SDLEASLB	Address of SUBLIB-DEF-TAB
72	48		Total length

Layout of the Old LIBRARIAN User DE-Format

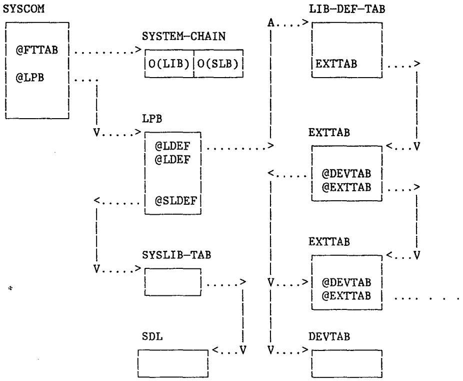
DEC	HEX	Label	Description
0	0	DIRNAME	Member name
8	8		*** internally used ***
11	B	DIRN	Number of halfword containing user data
12	C	DIRTT	Number of TXT blocks (1024 bytes)
14	E	DIRNN	TXT bytes in last TXT block
16	10	DIRC	Flags
		SELFREL	X'80' selfrelocatable
		RELPHASE	X'40' relocatable
		SVAELIG	X'20' SVA eligible
		SVAPHASE	X'10' phase is SVA-loaded
		PCIL	X'08' not-SYSLIB flag for in-core-DE
		NOTFND	X'04' not found flag
		ACTIVE	X'02' active DE (but possibly not found)
			X'01' reserved
17	11		Reserved
18	12	DIRPPP	Load point at LINKEDT time
21	15	DIREEE	Entry point at LINKEDT time
24	18		*** not supported ***
27	1B	DIRAAA	Partition begin at LINKEDT time
30	1E	DIRVEE	SVA entry point (if SVA-loaded)
34	22	A	*** not supported ***
38	26		Total Length

Layout of the New LIBRARIAN User DE-Format

DEC	HEX	Label	Description
0	0	DIRNAME	Member name
8	8		X'FFFFFF'
11	B	DIRN	Number of halfword containing User data (X'0E')
12	C	DIRLMBR	Length of phase in bytes
16	10	DIRC	Flags
		SELFREL	X'80' selfrelocatable
		RELPHASE	X'40' relocatable
		SVAELIG	X'20' SVA eligible
		SVAPHASE	X'10' phase is SVA-loaded
		PCIL	X'08' not-SYSLIB flag for in-core DE
		NOTFND	X'04' not found flag
		ACTIVE	X'02' active DE (but possibly not found)
			X'01' reserved
17	11		reserved
20	14	DIRCOPY	PTR TO DE-COPY
24	18	DIRPPP	Load point at LINKEDT time
28	1C	DIREEE	Entry point at LINKEDT time
32	20	DIRAAA	Partition begin at LINKEDT time
36	24	DIRVEE	SVA entry point (if SVA-loaded)
40	28		Total Length

Relationship Between Library Control Blocks

The relationships between the control blocks in the Supervisor are the same as for the NEW LIBRARIAN. A so called system searching chain is established during the FETCH initialization and will be maintained by the LIBRARIAN services.

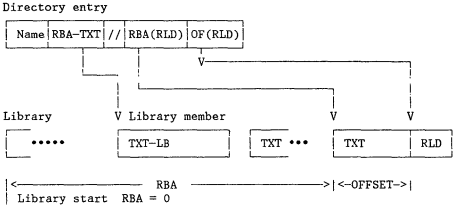


Notes:

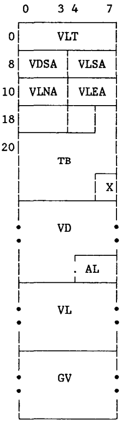
1. The SLD of the SYSLIB is built by the librarian at end of IPL time.

2. The meanings of the various control blocks are given below.

Relationship Between Directory and Phase-Member.



SHARED VIRTUAL AREA (SVA)



Field	Length	Contents
VLT	8	Date plus time SVA has been updated
VDSA	4	Start Address of the System Directory List
VLSA	4	Start Address of the virtual library (Note 1)
VLNA	4	Address of next available location in SVA
VLEA	4	Address of end of SVA
	4	Pointer to next free SDL entry
	2	Number of SDL entries
TB	16	STOWTABLE (Communication area between Job Control and $MAINDIR/$MAINDIF)
X	2	Number of bytes in SDL
VD	any	System Directory List (Note 2)
AL	<8	Alignment bytes for doubleword boundary
VL	any	Virtual library containing reenterable and relocatable phases
GV	any	GETVIS area for the system (starts on page boundary)

Notes:

1. Address of first doubleword aligned byte after SDL.

2. The layout of this area is compatible with a directory entry in the library. The SDL has fixed legth entries of 72 bytes. The last contains 8X'FF' as phasename.

JOB ACCOUNTING TABLES (ACCTCOMN, ACCTABLE)

Job Accounting Common Table (ACCTCOMN)

Bytes Dec	 Hex	Label	Description
0-3	0-3	ACCTPCNT	Count of active partitions
4	4	ACCTSWCH	Job control switches
		ACCTCTSW	X'20' Catal switch
5-7	5-7		Reserved
8-11	8-B	ACCTABLN	Length of JA partition table
12-15	C-F	ACCTUSEP	Address of JA user save area
16-19	10-13	ACCTUSEL	Length of JA user save area

Bytes 124-127 (X'7C - 7F') of the System Communication Region (SYSCOM) contain the address of the Job Accounting interface common table.
Label ACCTCOMN identifies the first byte of the table.

Job Accounting Interface Partition Table (ACCTABLE)

Bytes		Label	Description
Dec	Hex		
0	0	ACCTSWTC ACCTACTV	Accounting partition switches X'80' Indicate JCL-N/$JOBACCT active
1-7	1-7		Reserved
8-11	8-B	ACCTSVPT	Address of job card field following job name
12-13	C-D	ACCTNSIO	Current number of SIO count fields
14-15	E-F	ACCTLEN	Length of SIO area = 6n+1, where n is the number of devices accessed by the job step
16-23	10-17	ACCTCLCK	Time field in seconds
24-27	18-1B	ACCTLADD	Address of label area
28-31	1C-1F	ACCTCPUT	Partition CPU time counter for current step
32-35	20-23	ACCTOVHD	Overhead time counter for current job step (distributed in proportion to CPU time)
36-39	24-27	ACCTBNDT	System wait time for current job step (distributed in equal parts to activate partition) (Note)
40-47	28-2F	ACCTSVJN	Save area for job name during simulated EOJ
.... Following information passed to the user			
48-55	30-37	ACCTJBNM	Job name from job card
56-71	38-47	ACCTUSRS	User information from job card
72-73	48-49	ACCTPTID	Partition ID
74	4A	ACCTCNCL	Cancel code for job step
75	4B	ACCTYPER	Type of record: 'L'=last job step, else 'S'
76-83	4C-53	ACCTDATE	Date of end of job step in the format MM/DD/YY or DD/MM/YY, depending on the DATE standard option
84-87	54-57	ACCTSTRT	Stop time of previous job step, in packed decimal
88-91	57-5B	ACCTSTOP	Stop time of job step, in packed decimal
92-95	5C-5F	ACCTDUR	Step duration time in seconds, in binary
96-103	60-67	ACCTEXEC	Phase name taken for EXEC statement
104-107	68-6B	ACCTHICR	length of page * number of partition pages referenced (or PFIXed for real execution) in the current job step. For MODE=VM, this field contains the highest virtual storage address allocated to this partition
108-111	6C-6F	ACCTIMES	Same as ACCTCPUT at the end of the job step
112-115	70-73		Same as ACCTOVHD at the end of the job step
116-119	74-77		Same as ACCTBNDT at the end of the job step
120	78	ACCTSIOS	Six bytes for each device accessed by the job step, as follows: 2 bytes for device addr.(0cuu) 4 byte for SIO count in current job step
last byte			Overflow byte: always X'20', indicating no overflow

Note:

Bytes 116-119 (X'74 - 77') of the Partition Communication Region (COMREG) contain the address of the ACCTABLE.

PAGE MANAGEMENT

Segment Table Entry (370 Only)

0	4	29	30	31
PTL	page table addr	0	C	I

```
PTL :    (16/max * len) - 1
         len = actual length of page table
         max = maximum size of page table
page table address:
         address of page table segment allocated to entry
C   :    common segment bit
I   :    invalid segment bit ( = 0 - the segment is valid)
                             ( = 1 - the segment is invalid)
```

Page and Page Table Entries (370 Only)

Bit	Label	Description
0-15	PTE	Page addressable
0-11	PFRA	Page frame number
12	IBIT	Invalid bit = 0
13-15		Architected = 0

Page Table Entry (PTE) for Addressable Page

Bit	Label	Description
0-15	PTE	Page not addressable
0- 4	STKEY	Storage key of page
5- 9		reserved
10	HABIT	Invalid state: HABIT = IBIT = 1
11	COBIT	Connected state : COBIT = IBIT = 1
12	IBIT	Invalid bit = 1
13-14		Architected = 0
15	PDSBIT	Valid copy on PDS = 1 no copy on PDS = 0

Page Table Entry (PTE) for not Addressable Page

Page Frame Table Entry (PFTE)

0	1	3	4	5	6	8	12 15
PFTE Flag	Page Number (PNR)	370 Mode Flag	Waiting Task ID (WID)	PFIX Counter	TFIX Counter	Forward Pointer	Backward Pointer

Byte(s)	Bit	Description
0 - 2		Page frame number (0, 1, 2, 3, ...)
3	4	=0 Page frame belongs to supervisor or IPL partition.
	4	=1 Page frame belongs to initial page pool
4 - 7		zero

PFT Entry Byte Description

Byte	Label	Description
0	PFTEFLG	PFTE flag
	HBIT	X'80' Each task causing a page fault can use the the page before it is disconnected again.
	POEBIT	40 The PFTE is enqueued for page-out.
	POBIT	20 An active entry from the PMR task is enqueued for page-out.
	POABIT	10 I/O for a page-out has been started for this PFTE.
	PCBIT	08 The page which belongs to the page frame has connected state. Either a page-in or an unconditional page-out request is in progress.
	POSYSBIT	04 A page-out request is in a system queue.
		02 - 01 Reserved
1- 2	PNR	If a page belongs to the page frame, these bytes contain the page number (='virtual-page-address'/pagesize). If a block of VIO storage belongs to the frame, these bytes contain the block number.
3	S370FLG	370 mode flag
	NFRP	X'80' Frame is used by a PFIXed page. Since the frame is in the PSQ or IPFQ this page must not be TFIXED if the TFIX counter is zero.
	NFVP	40 Page belonging to this frame is requested by PFIX. The frame is not in the PSQ. The PFIX request cannot be satisfied immediately.
	DRAP	20 The address space belonging to the PFTE is failing storage.
	PFTEBLK	10 Only block of VIO-storage connected to frame
	PNRINV	08 Page frame is unused. The PNR-, FIX- and WID fields are invalid. Also the PFTE- and 370 mode flags (except for NFRP and DRAP bits) are invalid.
	PFTEREAL	04 Frame is used by real partition.
		02 - 01 Reserved
4	PFTEWID	Waiting task id (370 mode only): Contains the PIK of the partition requesting PFIX. The page frame of the page to be PFIXed does not belong to the corresponding real partition.
5	PFIXC	Indicates how often the page is PFIXed.
6- 7	TFIXC	Indicates how often the page is TFIXed.
8-11	PFTEFPTR	Pointer to the next PFTE.
12-15	PFTEBPTR	Pointer to the preceding PFTE.

Note:

The pointers in bytes 8 through 15 are only valid if the PFTE is in the PSQ, or, for 370 mode, in the IPFQ.

Page Table Assignment String (PTAS)

Bytes	Label	Description
0 - 1	PTASE	Entry length 2 bytes
0	PTASESPN	Space number where the PTAB belongs to
1	PTASESGN	Segment number where the PTAB is assigned to

Page Data Set Table (DPDTAB)

Bytes 224-227 (X'E0'-X'E3') of the System Communication Region (SYSCOM) contain the address of the DPDTAB.

The DPDTAB consists of a header and 15 extent definitions.
Label DPDTAB identifies the first byte of the table.

Dec	Hex	Label	Description
0-15	0- F	DPDADR	Header
0- 1	0- 1	DPDEXT#	Number of possible extents
2- 3	2- 3	DPDAEXT#	Number of actual extents
4- 7	4- 7	DPDPAG#	Number of supported pages
8-11	8- B	DPDLLCON	Address of load leveling constants
12-13	C- D		Reserved
14-15	E- F	DPDLEN	Length of header

Page Data Set Table Header

Dec	Hex	Label	Description
0-31	0-19	DPDENTR	Extent definition
0- 1	0- 1	DPDUNT	CUU of PDS device
2	2	DPDDEVT	Device type:FBA, CKD, RPS
3	3	DPDDEVC	Device code (DTF)
4- 5	4- 5	DPDREC#	CKD: # records/track
4- 5	4- 5	DPDBLKLG	FBA: block length
6- 7	6- 7	DPDTRCK#	CKD: # tracks/cylinder
6- 7	6- 7	DPDBLKPG	FBA: # blocks/page
8-11	8- B	DPDRTLL	CKD: track# of lower extent limit
8-11	8- B	DPDBLKLL	FBA: block# of lower extent limit
12-15	C- F	DPDTRCKU	CKD: # of used tracks
12-15	C- F	DPDBLKU	FBA: # of used blocks
16-17	10-11	DPDPUB	PUB index
18-23	12-17	DPDVOLID	Volume id of PDS
24-27	18-1B	DPDPGUL	Page # of upper limit
28-31	1C-1F	DPDDEVCB	Addr. of DEVCB for extent

Page Data Set Extent Definition

Device Control Block (DEVCB)

Every PDS device is described by its associated Device Control Block (DEVCB).

Bytes		Label	Description
Dec	Hex		
0	0	DEVCB	Device control block
0- 3	0- 3	DEVCBNXT	Addr. of next DEVCB if any, addr. of first DEVCB in chain for last DEVCB
4	4	DEVSTAT	Status byte
		DEVSTRT	X'80' I/O request started
		DEVEMPTY	X'40' no I/O request enqueued
		DEVPGWO	X'20' request waits for unconditional page out
5	5	DEVCBTYP	Device type: FBA,CKD,RPS
6- 7	6- 7	DEVEXT#	Number of extents on device
8- 11	8- B	DEVACT	Address of PGQE
12- 15	C- F	DEVDPD	Addr. of 1st DPD entry for device
16- 19	10- 13	DEVRELO	Relocation for 1st DPD entry on device
20- 23	14- 17	DEVAPTAS	Addr. of 1st PTASE for device
24- 25	18- 19	DEVPTASA	Highest offset of PTASE already occup.
26- 27	1A- 1B	DEVPTASB	Number of contiguously located PTASEs and still available on device
28- 31	1C- 1F	DEVPCB	Address of related PCB
32- 35	20- 23	APFPSS	Address of PFPSS for dev.
36- 36	24- 27		Reserved
40- 55	28- 37	DEVCCB	CCB for device
56-103	38- 67	DEVCCW	CCW program area
104-107	68- 6B	PFRQBEG	Begin addr. of system page fault queue
108-111	6C- 6F	PFRQEND	End addr. of system page fault queue
112-	70-		Partition queue headers in the sequence BG, Fn, ... , F1 length = NPART*2*4
			NPART = 12 :
208-211	D0- D3	PORQBEG	Begin address of page-out queue
212-215	D4- D7	PORQEND	End address of page-out queue

Page I/O Request Element (PGQE)

The PGQE is part of Task Information Block (TIB). The following fields are relevant for page management.

Dec	Hex	Label	Description
0	0	TIBADR	Task information block
0- 3	0- 3	TIBCHAIN	
4- 7	4- 7	TIBSTATE	Bound state information page-in: page fault addr page-out: pageframe addr
8-11	8- B	TIBPFAPP	Addr. of PHO appendage
8-11	8- B	TIBVIOTB	Addr. of VIOTAB entry
12-15	C- F	PGQE	Page I/O request element
12	C	PGQTYP	Request type
		PGSEL X'80'	Page selection required
		PGNCNT X'40'	Page-in, counting done
		PGO X'10'	Page-out request
		PGOWAIT X'18'	Page-out req. with waiting task
		PGOPGIN X'14'	Page-out req. with waiting page-in
		PGOVIO X'12'	Page-out req. from VIO
13-15	D- F	PGINF	Information for page I/O handling
.	.		.... further TIB
.	.		

Page-in Table (PAGETAB)

TID	PAGE Address	FLAG	ECB Address
0	1 3	4	5 7
• • •			• • •

Bytes	Description
0	Identifier of task that issued the PAGEIN macro
1 – 3	Pointer to the areas to be paged–in.
4	Flag Byte: X'80' PAGEIN request completed, second scan needed. 40 Reserved. 20 At least one page is outside partition boundary. 10 At least one entry with a negative length was found 08 Reserved. 04 Paging activity too high, termination was requested by LOAD LEVELER. 02 Task is terminating, entry has to be deleted. 01 Second scan in progress.
5 – 7	Pointer to ECB (if used) or zero.

If the address of an ECB was specified in the PAGEIN macro, information is returned in byte 2 of that ECB as shown below:

Bits	Meaning	Set by: SVC Routine	Set by: PGN Task
0	PAGEIN request completed (see Note)	Y	Y
1	Page–in table (PAGETAB) is full.	Y	N
2	One or more of the requested pages are outside the address range of the requesting program's partition.	N	Y
3	At least one negative length has been detected in the processed area specifications.	N	Y
4	List of areas that are to be paged–in is not completely contained in the requesting program's partition.	Y	N
5	Paging activity too high. PAGEIN request terminated by LOAD LEVELER.	N	Y
6 + 7	Reserved.		

Note:
Bit 0 is set by the PGN system task if that task receives control to process the pertinent PAGEIN request, otherwise the bit is set by the SVC routine.

Routine Identifiers (RID) as Used by the Page Fault Handler (PFFLIH)

NAME	ID	MEANING	ACTION
SYSTEMID	00	System error condition, for example, page fault in the I/O interrupt handler	Hard wait.
REENTRID	04	Page fault or GETREAL request in a reenterable routine.	Save PSW and registers in user task's system save
USERID	08	Page fault from a user task or from a system task.	Hard wait X'FFB' if this is a system task and the TCB shows that the task does not expect page faults; else registers and interrupt status are saved in the users save area. If the task operated in disabled mode, the task is canceled with cancel code X'15'; otherwise the page request is enqueued.
APPENDID	0C	Page fault in I/O appendage routine	Task is canceled with cancel code X'36'.
RESVCID	10	Page fault in SVC 7 (X'07') in SVC 13 (X'0D')	Set RETRYSVC bit in TIB save interrupt status and regs. in user save area; enqueue page request.
DISPID	14	Page fault in a routine which does not require any inform. to be saved, e.g. page fault in the disp.	Enqueue page request.
PFARID	18	Page fault in a page fault appendage routine.	Save interrupt status and registers in an internal save area and cancel user task w. cancel code X'0E'
ETSSID SUBSYSID	1C	Page fault in subsystem	Save interrupt status and registers in an internal save area.
MICRID	20	Page fault in MICR or subsystem appendage.	Save interrupt status and registers in an internal save area and cancel user task w. cancel code X'0E'
	40 . . . . FF	Page fault in a gated Supervisor service.	Close gate to routine (routine cannot be used until gate is opened). Save PSW and registers in the user task's system save area; set TIBFLAG to return to SVRETURN. Enqueue page request. (Any task accessing a gated resource is put in a wait state and is marked resource bound. It is released from the wait state when the resource is ungated after the page request has been completed

VTAM Address Vector Table (ISTAVT)

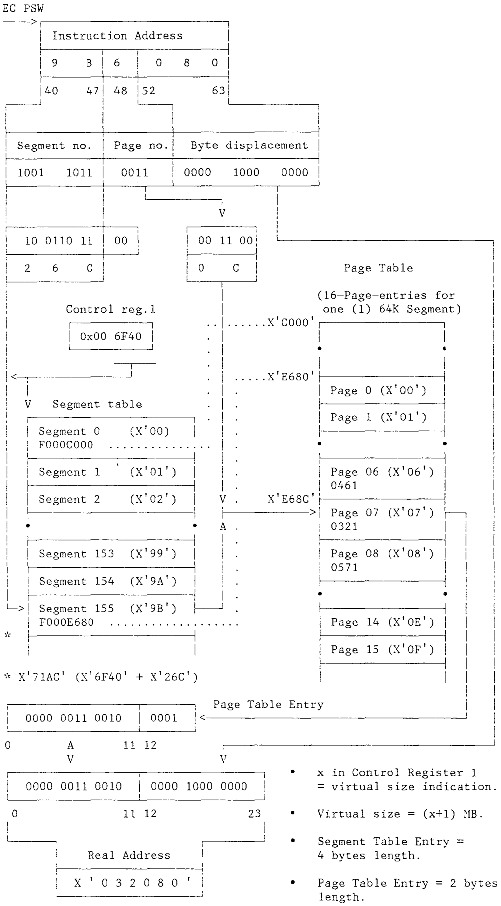
Label	Length	Description
ISTACVT	4	Address of VTAM CVT
ISTAS49	4	Address of SVC 49 Code
ISTAS53	4	Address of SVC 53 Code
ISTCFCSA	4	Address of Command Handler
ISTAPSEX	4	Address of APS Exit
ISTAPSTA	4	Address of APS Table
ISTARID	4	Address of RID
ISTVTTP	4	Address of Code to check for
ISTRETR6	4	Base Register for VSE Dispatcher
ISTRETR7	4	Return Register for Dispatcher
ISTTTXSZ	4	
ISTVTTIK	4	VTAMRP Task ID
ISTPHNM	5	Phase Name of Transient
ISTX1	1	TOLTEP saves SIO Condition Code

Note:
Bytes 96-99 (X'60-63') of the System Communication Region (SYSCOM) contain the address of the VTAM Address Vector Table.

Entry in the Asynchronous Process Scheduler (APS) Table

Label	Length	Description
APSFLAG	1	X'20' User Exit Delay for LTA
	3	Pointer to VTAM APT
APSCNT	1	Count of VTAM ACBs open
VTAMFLAG	1	X'80' TPBAL issued (AR only)
		X'40' DUMP already taken
		X'20' VTAM delayed Cancel
		X'10' AP delayed for temination
		X'08' VTAM User Exit in control
		X'04' VTAM SVC active
		X'02' VTAM Appendage active
		X'01' Key 0 / Substate required
	3	Reserved

Converting Virtual to Real Address (/370 u-program)



- x in Control Register 1 = virtual size indication.
- Virtual size = (x+1) MB.
- Segment Table Entry = 4 bytes length.
- Page Table Entry = 2 bytes length.

Note: Values used in the figure are hypothetical

CHAPTER 5. SERVICE AIDS

OLTEP

Functions:

- Diagnosing I/O errors
- Verifying I/O device repairs and engeneering changes
- Checking I/O devices
- Verify the integrity of customer data

System Generation requirements: none

Requirements for Execution of OLTEP:

- OLTs and CDSs available in Sublibrary IJSYSRS.SYSLIB
- Devices to be tested are in 'Ready' status
- Any real partition having at least 18K of storage
- If OLTEP runs in a VSE/POWER controlled partition or has to test devices being spooled by VSE/POWER, please consult the DOS/VSE-OLTEP SRL (GC33-6156) for special recommendations

JCL to invoke OLTEP

Statement	Comments
// JOB XXXXX	Required
// ASSGN SYS000,cuu	This ASSGN statement is necessary if NST loading is to be performed.
// DLBL BGBPDTF, 'SAK.SYSM'	This DLBL statement is necessary if NST loading is to be performed.
// EXTENT BGBPDTF	This EXTENT statement is necessary if NST loading is to be performed.
// ASSGN SYSnnn,cuu	One ASSGN statement is necessary for each device tested or accessed by a test. None is required if the device was permanently assigned.
// UPSI 01	This statement is necessary if a console device is available but the test-run definition is to be entered via the input job stream (SYSIPT).
// UPSI 10	This statement is necessary if READD data input is contained on diskette.
// UPSI 11	This statement is used when both of the above options are being used.
// UPSI 001	This statement is needed if OLTs reside on NST disk.
// UPSI 0001	This statement is needed if OLTs reside on NST tape.
EXEC IJZADOLT, REAL[,SIZE=nK]	Mandatory, OLTEP will run only in real. SIZE= parameter must be a minimum of 18K; this will allow a 4K OLT to execute. If OLTs larger than 4K are to be run, the SIZE=parameter must specify a size equal to 14K plus the size of the OLT. The size specified must be a multiple of 2K.If the SIZE=parameter specifies a size of 22K or larger, add 4K since the 4K history tape input buffer will be automatically allocated during OLTEP initialization. If NST loading is to be performed, the minimum core needed is 26K. If 30K is available, a history tape read buffer will automatically be allocated. The SIZE-parameter must always be used when NST loading, because a 4K buffer is allocated in the GETVIS area.
dev/test/opt	This statement is included if the test-run definitions are entered via the input stream.
/*	Required
/&	Required

Table of Options

Option	Entry	Description
Testing Loop	TL(n) NTL	Recycle the test. If you specify a value (n), OLTEP runs the test the number of times indicated If you do not specify a value, the test cycles 10 times. The maximum value allowed is 32,767 decimal. (see Note 1)

Error Loop	EL(n)
	EL(I)
	NEL
		Authorizes any error loop coded in the OLT to be executed the specified number of times. If you specify a value (n), the test loops the number of times indicated. If you do no specify a value, the test loops the number of times indicated in the preface of the OLT. If you specify the character I, a flag is set which indicates to the OLT, that it must loop indefinitely on the error. You can terminate the loop by specifying NEL following a request for communication (Note 1)

Print	PR NPR	Print messages from the OLT. If you enter NPR, all messages originated by the OLT and normally designated for SYSLST are suppressed. (Note 2)
Error Print	EP NEP	Print diagnostic error messages from the OLT. The FE option overrides NEP when a first error is encountered (once per section). (Note 3)
Control Print	CP NCP	Print OLT start and termination messages nn SYSLST and SYSLOG.
Parallel Print	PP NPP	Use the consol device, in addition to SYSLST for OLT messages. Four levels of print are available on the parallel printer by entering one of these numbers at (n). 0: HEADER only 1: HEADER, DESCRIPTION and COMMENTS 2: HEADER and RESULTS 3: HEADER, DESCRIPTION, COMMENTS and RESULTS
First error Communications	FE NFE	Forces a communications interval when the first error is encountered. (Note 3) A message is printed indicating the test being run and the device being tested. This is followed by the 01E105D message, that allows you to: • Change the device and/or test fields. • Continue the test by entering /// or //(Option change)/. • Enter any OLTEP verb. • Cancel OLTEP by entering CANCEL. There cannot be a first error communication if a console device is not available.
Manual Intervention	MI NMI	Informs the OLT section to run all manual intervention routines within the test request. (MI and RE are mutually exclusive options.)

Table of Options (cont...)

Option	Entry	Description
Trace	TR NTR	Trace all functions called by OLT. (Note 4)
EXT=	EXT=	Information following this option is passed to the OLT section by way of a 56-byte buffer. This information must be the last entry in the option field and can contain any character but a slash. EXAMPLE: 181/2400C/TL,EXT=BLOCK 4FFPRINT/ BLOCK 4FFPRINT goes into a buffer area within OLTEP and then passes on to the OLT section.

The default options are underlined.

Notes:

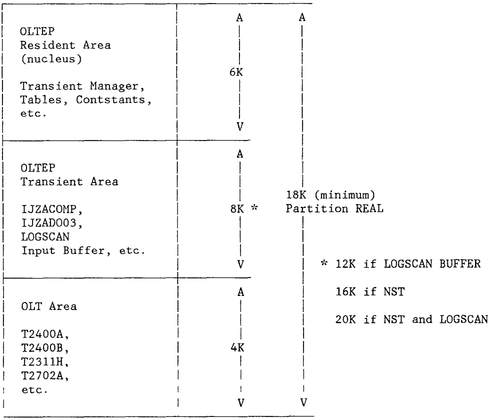
1. The FE option overrides the TL and EL options, unless NPR is also in effect. However if you enter /// or //option change/ at a first error communications interval, the TL and EL options, if specified, are in effect.

2. NPR without EL and/or TL is ignored.

3. EP and FE are ignored if no print and either EL or TL are specified.

4. Routine to routine linkage is not traced. Do not attempt to use Trace function when SYSLST is assigned to the test device.

Storage Layout



Example of OLTEP Operation

Note: Example is shown for OLTEP operation in the background partition of a multiprogram environment. A console device is available, and console interrupts are permitted. The Operator responses are underlined.

```
BG assgn sys010,181                   ┌─────────────────────────────────┐
                                      │ OLTEP is loaded into the back-  │
BG assgn sys011,182                   │ ground partition. Example loads │
                                      │ OLTEP into the background parti-│
BG // exec ijzadolt,real[,size=nK]    │ tion, OLTEP can also be loaded  │
                                      │ into a foreground partition.    │
BG E102I OLTS RUNNING                 └─────────────────────────────────┘

BG E134I WARNING - DASD VOLUME LABELED CEPACK NOT PROTECTED FROM WRITE

BG E107I OPTIONS ARE NTL,NEL,NPP,     ┌─────────────────────────────────┐
                 NFE,NMI,CP,PR,NTR    │ Initial communications interval │
BG 01E105D ENTER - DEV/TEST/OPT/      └─────────────────────────────────┘

BG 181/T3420a/fe/                       ┌───────────────────────────────┐
                                        │ OLTEP finds a nonstandard     │
                                        │ labeled tape mounted on the   │
BG E119I NON-STANDARD TAPE LABEL 0181   │ test device.                  │
                                        └───────────────────────────────┘
BG 04E139D REPLY B TO BYPASS,R TO RETRY,P TO PROCEED (MAY DESTROY DATA)

BG p
                                      ┌─────────────────────────────────┐
BG E158I S T3420A $ UNIT 0181         │ Error encountered               │
                                      └─────────────────────────────────┘
BG E129I FIRST ERROR COMMUNICATION T3420A 001 UNIT 0181

BG E107I OPTIONS ARE NTL,NEL,NPP,FE,NMI,EP,CP,PR,NTR

BG 01E105D ENTER - DEV/TEST/OPT/      ┌─────────────────────────────────┐
                                      │ First error communications      │
BG 182//nfe,tl(2)/                    │ interval.                       │
                                      └─────────────────────────────────┘
BG E158I *T T3420A $ UNIT 0181

BG E158I S T3420A $ UNIT 0182

BG E158I T T3420A $ UNIT 0182
                                      ┌─────────────────────────────────┐
BG E158I S T3420A $ UNIT 0182         │ Console interrupt key pressed.  │
                                      └─────────────────────────────────┘
BG E107I OPTIONS ARE TL,NEL,NPP,NFE,NMI,EP,CP,PR,NTR

BG 01E105D ENTER - DEV/TEST/OPT/      ┌─────────────────────────────────┐
                                      │ Interrupt communications        │
BG /t3420c,e/ntl/                     │ interval                        │
                                      └─────────────────────────────────┘
BG E158I *T T3420A $ UNIT 0182

BG E158I S T3420C $ UNIT 0182

BG E158I T T3420C $ UNIT 0182

BG E158I S T3420E $ UNIT 0182

BG E158I T T3420E $ UNIT 0182

BG E107I OPTIONS ARE NTL,NEL,NPP,NFE,NMI,EP,CP,PR,NTR

BG 01E105D ENTER - DEV/TEST/OPT/      ┌─────────────────────────────────┐
                                      │ Test completion communications  │
BG cancel                             │ interval                        │
                                      └─────────────────────────────────┘
BG 1I00A READY FOR COMMUNICATIONS
```

EREP

You can use the IFCEREP1 service aid program to retrieve all or selected records from the input data set(s), edit the records, and write them to any specified output device supported ba the Sequential Access Method (SAM). A direct access device may be required for allocation of a temporary work data set. SAM is the access method utilized for sequentially writing this temporary data. IFCEREP1 processes SYSREC both sequentially and randomly using the EXCP access method.

VSE Storage Requirements

EREP requires at least 100K of virtual storage. This provides for a sort table of 4K bytes, the VSE TABSIZE default. The 4K-byte sort table permits the processing of approximately 400 records for a report.

Executing IFCEREP1

Program IFCEREP1 is the main execution routine for running EREP. EREP parameters may be specified only via input statements (SYSIPT).

The operator should execute the ROD command prior to running EREP from SYSREC.

Executing IFCOFFLD

Program IFCOFFLD is a special purpose load of EREP modules that allows you to clear SYSREC, under emergency conditions, without losing the data recorded thereon. You should create a member of PROCLIB so the operator can start the job from the console.

No parameters are allowed when executing IFCOFFLD.

- SYSREC is input.
- SYS009 is the history output data set logical unit number.
- System Summary Report is printed.
- If message IFC119I is received, alter the SIZE parameter on the // EXEC card and, if necessary, alter the partition size.

EREP Basic Functions
1. Create an Accumulation data set from the SYSREC data set.
2. Clear SYSREC.
3. Copy an input Accumulation data set to an Output Accumulation data set.
4. Merge data from an Accumulation data set and SYSREC.
5. Print detail description of hardware and software error records.
6. Summarize and print statistics for device failures.

EREP Reporting Functions
1. System Summary Reporting
2. Trends Reporting
3. Event History Reporting
4. Threshold Reporting
5. Record detail and/or Summary Reporting
6. Offload

Statement	Usage
// JOB EXAMPLE	This statement initiates the job.

// TLBL HISTINT
or
// DLBL HISTIND
// EXTENT SYS008,XXXX,1,,
XXXX,XX (Note 1)
// ASSGN SYS008,cuu
	These job control statements are used to process history input. Either history input or SYSREC input or both must be processed each IFCEREP1 execution. The ASSGN statement must always be used for history input. The TLBL is used for tape resident history input; the DLBL and EXTENT for disk resident history input.

// TLBL HISTOT
or
// DLBL HISTOD
// EXTENT SYS009,XXXX,1,,
XXXX,XX (Note 1)
// ASSGN SYS009,cuu
	These job control statements are used to create a history output data set. The TLBL statement is used for tape resident history output; the DLBL and EXTENT statements are used for disk resident history output.

// ASSGN SYS001,cuu
// DLBL IJSYS01
// EXTENT SYS001,XXXX,1,,
XXXX,XX (Note 1,2)
	These job control statements are used to define the temporary work data set on a direct access device. These statements are necessary when a history input data set is read (Note 3). Records are handled via SAM in undefined format. The EXTENT and DLBL statements should not be necessary as SYS001 should already be defined for the linkage editor. The standard SYS001 EXTENT should provide enough space for most IFCEREP1 executions; at any rate, enough space must be allocated to store all records selected from the input data set(s).

// EXEC PGM=IFCEREP1,
SIZE=100K
	This statement specifies the program name to be executed. The minimum virtual region size for DOS/VSE is 100K. GETVIS requires 1K for each 100 records to be processed.

The following system logical units are used by IFCEREP1 but should already be assigned.

SYSREC	The assignment for the System Recording data set must already be made.
SYSLST	Both message output and report output are sent to this logical unit.
SYSLOG	If SYSLST is unavailable, a termination message is written to the console.
SYSIPT	IFCEREP1 input parameters and control cards are input to this required system logical unit.

Notes:

1. Ask your system programmer how to code EXTENT statements.
2. SYS001 must be a ONE EXTENT data set.
3. PRINT=NO does not require the work data set for history input.

Logical Units Required by Function

Logical Units	EREP Keywords				
	ACC=Y	HIST=Y	MERGE=Y	ALL OTHER	HIST=N
SYS009	X				
SYS008		X	X		
SYS001		X	X		
SYSREC			X	X	X
SYSLST	X	X	X	X	X
SYSLOG	X	X	X	X	X
SYSIPT	X	X	X	X	X

EREP Keywords

EREP uses parameters to determine the functions requested and any restrictions placed on the records to be processed. The parameters are in alphabetical order. The default value, if applicable, is underlined. All keyword parameters in EREP are optional; none have to be specified. However, you should check the default options to ensure they are the ones you want.
For details please consult EREP User's Guide and Reference (GC28-1378).

<table>
<tr><td>ACC [Y]|N
Default exceptions:
THRESHOLD</td><td>Accumulate selected records</td></tr>
<tr><td>CPU=(serial.model
[,serial.model]...)
maximum of 6 entries</td><td>Select CPU by serial number (nnnnn) and model (nnnn)</td></tr>
<tr><td>CPUCUA=(serial.addr
[,serial.addr]...)
maximum of 4 entries</td><td>Select unique device addresses on a specific CPU</td></tr>
<tr><td>CUA=(entry[,entry]...)
maximum of 8 entries</td><td>Select device/control units by unique addresses (nXX, nnX or nnn)
n=hex digit, X=character 'X'</td></tr>
<tr><td>DATE=(yyddd[}−|,]yyddd])
single date or date range</td><td>Date span for selected records</td></tr>
<tr><td>DEV=(type[,type]...)
maximum of 8 entries</td><td>Select device type (nnnn or nnXX)
X=character 'X'</td></tr>
<tr><td>DEVSER=(serial[,serial]...)
maximum of 8 entries</td><td>Select device serial numbers (nnnnn)
(34XX devices only)</td></tr>
<tr><td>ERRORID=(seqno[,cpuid,asid,
hh,mm,ss,t])</td><td>Valid only for MVS software records</td></tr>
<tr><td>EVENT[=Y]|=N</td><td>Provide an event history report</td></tr>
<tr><td>HIST[=Y|=N</td><td>Indication for input data set</td></tr>
<tr><td>LIA|LIBADR=address</td><td>Select records by Line interface base address (XXXX − hexadecimal)</td></tr>
<tr><td>LINECT=nnn
Number of lines
Default=50 lines</td><td>Number of lines to be printed on a page (nnn − decimal)</td></tr>
<tr><td>MERGE[=Y]|=N</td><td>Allows merging of EREP input (Accumulation data set + SYSREC)</td></tr>
<tr><td>MOD=(model[,model]...)
maximum of 4 entries</td><td>Select specific CPU−models (nnn or nnnn)</td></tr>
<tr><td>MODE=}370|370XA|ALL]</td><td></td></tr>
<tr><td>PRINT=}AL|DR|PS|PT|SD|SU|NO]</td><td>Select format of printout (Note 2)</td></tr>
<tr><td>SHORT[=Y]|=N</td><td>Allows suppression of detail printing of OBR records</td></tr>
<tr><td>SYMCDE=[nnnn|nnnX|nnXX|nXXX]</td><td>Select records by fault symptom code
(33XX − disks only)
n = hexadecimal digit
X = character 'X'</td></tr>
</table>

EREP Keywords (cont...)

SYSEXN[=Y]|=N	Specifies EREP to produce a System Exception Report
SYSUM[=Y]|=N	Allows printing of a system summary report

TABSIZE=nnnK
Default=4K for VSE Systems
	Specify size of internal sort table

TERMN=name	Select records by terminal name (VTAM only)
THRESHOLD=(xxx,yyy)	Specify threshold value for temporary read/write errors (3410, 3420, 8809 only) – decimal digits
TIME=(hhmmm}–|,]hhmmm) Time range	Time span for selected records
TRENDS[=Y]|=N	Generate a Trends Report
TYPE=[C][D][E][H][I][M] [O][S][T]	Select records by their type (Note 1)
VOLID=(volser[,volser]...) maximum of 4 entries	Select records by volume serial number (3410,4320,8809,disk) 1 to 6 alphameric characters
ZERO[=Y]=N	Clear SYSREC after processing

Notes:

1. Record Types

Code	Meaning	Selection Keywords
C	CCH	CPUCUA, CUA
D	DDR	CPUCUA, CUA, DEV
E	EOD	
H	MIH	CPUCUA, CUA, DEV
I	IPL	
M	MCH	ERRORID (MVS only)
O	OBR	CPUCUA, CUA, DEV, DEVSER, SYMCDE, TERMN, VOLID
S	SFT	ERRORID (MVS only)
T	MDR	CPUCUA, CUA, DEV, LIA/LIBADR, VOLID

* Other selection keywords apply to all record types.

2. Print Keywords

AL	print all possible detail reports
DR	request only Data Reduction reports
NO	no reports will be printed
PS	print full record and summary
PT	suppress summary printing (print full record only)
SD	request detail summaries and Data Reduction reports
SU	suppress full printing (print summary only)

Incorrect EREP Parameter Combinations

Keyword Parameter Specifications not accepted by EREP.
(X = Conflicting Parameters)

	ACC	CPU	CPUCUA	CUA	DATE	DEV	DEVSER	ERRORID	HIST	LIA/LIBADR	LINECT	MERGE	MOD	MODE	SHORT	SYMCDE	TABSIZE	TERMN	TIME	TYPE	VOLID	ZERO
ACC	X						X															
CPU		X	X										X									X
CPUCUA		X	X				X						X									X
CUA			X	X																		X
DATE					X																	X
DEV						X	1			2										3	4	X
DEVSER	X		X			1	X	X		X			X		X	X		X				X
ERRORID							X	X														X
HIST									X			X										X
LIA/LIBADR						2	X			X											X	X
LINECT											X											
MERGE									X			X										
MOD		X	X				X						X									X
MODE														X								5
SHORT							X								X							
SYMCDE							X			X						X		X			X	X
TABSIZE																	X					
TERMN							X			X						X		X			X	X
TIME																			X			X
TYPE						3														X		X
VOLID						4				X						X		X			X	X
ZERO		X	X	X	X	X	X	X	X	X			X	5		X		X	X	X	X	X

Notes:

1. DEVSER is used for the Threshold Summary only, so the only devices are 3410, 3420, 8809, and 34XX.

2. LIA/LIBADR applies only to TP communication controllers, so the only valid devices are 3705 and 3725.

3. DEV is valid with only four record types: DDR(D), MIH(H), OBR(O), and MDR(T).

4. VOLID applies only to 33XX disks and 34XX tape dev.

5. ZERO is valid if you code default MODE=ALL.

Invalid Selection Parameters for EREP Reports

REPORT PARA-METERS	Selection Parameters														
	CPU	CPUCUA	CUA	DATE	DEV	DEVSER	ERRORID	LIA/LIBADR	MOD	MODE	SYMCDE	TERMN	TIME	TYPE	VOLID
EVENT=Y						X									
PRINT=AL						X									
=DR						X									
=PS						X									
=PT						X									
=SD						X									
=SU						X									
=NO						X									
SYSEXN=Y	X	X	X		X	X	X	X	X	X	X	X		X	X
SYSUM=Y						X									
TRENDS=Y						X									
THRESHOLD	X	X			1		X	X	X		X	X		X	

X means, selections are invalid.
All other combinations are valid.

Note:
The following devices are allowed: 3410,3420,8809,34XX.

Examples for an EREP-run

(For further examples please refer to 'EREP User's Guide and Reference' GC28-1378)

```
1. Copy SYSREC to a tape data set . 2. Print a summary report from tape
                                  .
// JOB EREP                       . // JOB SUMMARY
// TLBL HISTOT                    . // TLBL HISTINT
// PAUSE ASSIGN SYS009 TO TAPE    . // ASSGN SYS008,SYS009
// PAUSE ISSUE ROD COMMAND        . // ASSGN SYS009,UA
// ASSGN SYS001,cuu               . // EXEC IFCEREP1,SIZE=AUTO
// DLBL IJSYS01                   . SYSUM
// EXTENT SYS001,xxx ...          . ACC=N
// EXEC IFCEREP1,SIZE=AUTO        . TABSIZE=50K
PRINT=NO                          . ENDPARM
ACC=Y                             . /*
ZERO=N                            . // MTC REW,SYS008
ENDPARM                           . /&
// MTC REW,SYS009                 .
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
3. Update the history tape (Merge SYSREC with existing history tape)

// JOB MERGE
// TLBL HISTOT,'EREP.HIST.TAPE',nnn
// TLBL HISTINT,'EREP.HIST.TAPE',nnn
// PAUSE MOUNT SCRATCH TAPE AND ASSGN SYS009
// PAUSE MOUNT EREP.HISTORY.TAPE AND ASSGN SYS008
// EXEC IFCEREP1,SIZE=AUTO
PRINT=NO
MERGE
ACC=Y
ZERO=Y
ENDPARM
/*
// MTC REW,SYS009
// MTC REW,SYS008
/&
```

Abbreviations Used in EREP Output

BPI	Bits per inch
BUFE	Buffer error
BYTES RD/SRCHD	Megabytes read/searched
CCH	Channel check handler
CCH-CRH	CCH-Channel reconfiguration hardware
CCH-INC	CCH incomplete record
CDDA	Command data
CHNL	Channel
CHP/CHPID	Channel path ID
CK or CHK	Check
CLNACT	Cleaner action
CMD or CMND	Command
CNT or CT	Count
CNTRL	Control
CNTRLR or CTLR or CT	Controller
COMP/MOD	Component/Module
CONS+UR	Console plus unit record
CORR	Correctable
COR	Corrected
CPU or CP	Central processing unit (Processor)
CRW	Channel report word
CSECTID	Control section (CSECT) identification
CSID	Channel set ID
CSW	Channel status word
CTLID	Controller ID
C.U.	Control Unit
CUA	Channel-control unit-device address
DATAXFER	Data transfer
DATA CKS CORR/RTRY	Data checks correctable/retry
DDR	Dynamic device reconfiguration
DDR-OPR	DDR-Operator requested
DDR-SYS	DDR-System requested
DEV or DEVNO or DEVNUM or DNO	Device number
DEVT	Device type
DRCT	Storage director
DTE	Date
EOD	End of day
EQUCHK	Equipment check
EQUIP	Equipment
ERDS	Error recording data set (SYSREC for VSE)
ERROPS	Error operations
ERSGAP	Erase gap
ESW	Extended status word
EXTD	External damage
FCG	Floating channel group
FLG	Flag
FMT	Format
HDR SER	Header (tape)/serial number of drive that created tape
HIRS	Hardware instruction retry (successful)
ID	Identification
INV	Invalid
INVK	Invoked
IPL	Initial program load
IRB	Interrupt response block
LEN	Length
MB/MBYTE	Megabyte
MCH	Machine check handler
MCH-TRM	MCH-System terminated
MCK	Machine check
MDR	Miscellaneous data record
MDR-DAS	DASD (disk)-MDR record
MIH	Missing interrupt handler
MIH-CE	MIH-channel end pending
MIH-DE	MIH-device end pending
OBR	Outboard record

Abbreviations Used in EREP Output (cont...)

OBR-DMT	OBR-Demount record
OBR-DPA	OBR-Dynamic pathing availability
OBR-EOD	OBR-End-of-day
OBR-PRM	OBR-Permanent error record
OBR-SHT	OBR-Short record
OBR-TMP	OBR-Temporary error
OVERRN or OVRN	Overrun
OVERRUN CDDA or CDDA	Overrun command data/command data
PCUA	Primary channel-control unti-device address
PERM or PRM	Permanent
PFU	Probable failing unit
PRGM INT	Program-initiated
PRI	Primary
PROG-EC	Program-extended control mode
PSW	Program status word
RCVRYXIT	Recovery exit module
RD(S)	Read error(s)
REC-TYP	Record type
RTN	Routine
RTRY	Retry
SCSW	Subchannel status word
SCP	System control program
SCU	Storage control unit
SCUA	Secondary channel-control unit-device address
SCUID	Storage control unit ID
SD	Storage director
SEC	Secondary
SEEKS CNTR/HH	Seek errors cylinder track/head
SFT	Software (record)
SFT-ABN	SFT-ABEND record
SFT-MCH	SFT-machine error, recoverable
SFT-PI	SFT-program interrupt
SFT-RST	SFT-restart
SLH	Subchannel logout handler
SIO	Start I/O
SKS	Seeks; data access errors
SNID	Sense path group ID (DPA)
SPID	Set path group ID (DPA)
SRCHD	Searched
SSYS ID	Sub-system identifier
STOR	Storage error
TEMP or TMP	Temporary
TERM	Terminal
UCB	Unit control block
VOLID	Volume serial number
WRT(S)	Write error(s)

General Information

You submit the SDAID statements with one of the following methods:

- Direct input mode in the attention routine or partition.

- Job control procedures in a partition.

- Prompts in the attention routine (AR).

Initialization in Direct Input Mode

In direct input mode the SDAID information is entered in form of commands to the attention routine or as SYSIN statements in a partition.

The SDAID program identifies the mode of initialization via the format of the TRACE and OUTDEV statement. In direct input mode these statements must contain at least one operand.

The following examples show two initialization jobs, one entered in a partition the other one entered via SYSIN.

Example of a trace initialization in direct input mode in the attention routine:

```
==> sdaid #
AR 4C05I PROCESSING OF 'SDAID'  COMMAND SUCCESSFUL.
AR 1I40I READY
==> outdev tape=280 #
AR 4C05I PROCESSING OF 'OUTDEV'  COMMAND SUCCESSFUL.
AR 1I40I READY
==> trace sio unit=009 #
AR 4C05I PROCESSING OF 'TRACE'  COMMAND SUCCESSFUL.
==> trace io unit=009 output=ccw #
AR 4C05I PROCESSING OF 'TRACE'  COMMAND SUCCESSFUL.
==> ready #
AR 4C05I PROCESSING OF 'READY'  COMMAND SUCCESSFUL.
AR 1I40I READY

  #  indicates the Enter key pressed
```

Example of a trace initialization via direct input mode statements read in from SYSIN:

```
  // EXEC SDAID
  OUTDEV TAPE=280
  TRACE SIO UNIT=009
  TRACE IO UNIT=009 OUTPUT=CCW
  /*
```

Initialization via Job Control Procedures

The easiest way to initialize a trace is to use cataloged procedures. An example of such a trace procedure statement is shown below.

For example:

```
  // EXEC PROC=SDIO.UNIT=009.TAPE=280
```

Initialization via Prompts in the Attention Routine

You start the initialization process with the attention routine command 'SDAID' The necessary trace definitions are given in response to promptings after you entered the TRACE or OUTDEV statement without an operand.

You enter the prompt mode whenever you define these two commands without an operand.

Example of a trace initialization via prompts in the attention routine:

```
==> sdaid  #
AR 4C05I PROCESSING OF 'SDAID'  COMMAND SUCCESSFUL
==> outdev  #
AR 4C08D SPECIFY OUTPUT DEVICE.  +
==> tape  #
  ...

#  indicates the Enter key pressed
```

Note that you enter the prompt mode also if you specify direct input mode statements combined with prompt mode statements like a question mark (? requests the help function of SDAID).

The example below shows, how you can combine the two input modes. You would be prompted after the question mark has been processed.

```
TRACE SIO AREA=BG ?
- direct input—>|< prompt mode
```

Trace Type Summary

'Trace Type'	'Provides a Trace of:'
BRanch	instructions which caused a branch
BUffer	the trace buffer full condition
CAncel	program (main task) cancel or EOJ
EXTernal	external interrupts
INSTruction	selected or all instruction(s) execution
IO	I/O interrupts
MONitorcall	MC instructions
PAGing	page faults
PGMCheck	program checks
PGMLoad	phase load requests, or actual load
REGister	alteration of general purpose registers
SIO	SIO instructions
STorage	storage alterations
SVC	executed supervisor calls
VTAMBU	usage of ACF/VTAM buffers
VTAMIO	VTAM I/O operations

System Performance Degradation

The tracing of events with SDAID may affect overall system performance. This may especially affect time dependent programs (such as programs controlling MICR equipment or doing input/output via telecommunication lines).

When you invoke SDAID in the AR, console input is blocked during processing of each SDAID command until the final READY command has been processed successfully.

SDAID Space Requirements

Space Requirements during Initialization in the AR

The SDAID set-up phases require approximately 100K bytes of virtual storage within the system GETVIS area. When initialization is complete (the READY command is processed successfully), that GETVIS space is released.

Space Requirements during Initialization in a Partition

Beside the GETVIS space of 100K bytes the phase SDAID (called via EXEC SDAID) requires approximately 16K bytes of partition virtual storage. This is significantly less than the minimum VSE partition size. Therefore SDAID will run in any foreground or background partition.

Space Requirements for SDAID Execution

The amount of storage required for SDAID execution depends on the combination of trace operations that you request and on the size of the output buffer (specified in the OUTDEV command).

Basic requirement for SDAID execution:	12K
Additional requirements	
Per specified trace:	2K
If BUFFER=nn is specified:	2K + buffer size
If OUTPUT is used in the trace commands:	12K
If OUTDEV is a printer:	8K
If OUTDEV is a printer and OUTPUT is defined:	16K

The size of the SDAID area allocated by default is 48K bytes.

The size of the SDAID area may be increased during IPL using the SDSIZE parameter of the SYS command.

Statement Format Considerations

- The various operands may be separated by a comma or by at least one blank.
- Enter all operands in the order explained.
- Most keywords may be abbreviated. The possible abbreviation is shown through lower case letters.
- Mandatory operands are highlighted.
- Optional operands are enclosed in square brackets '[]'
- Operands separated by a '|' denote choices one of which must be selected.
- Command continuation is allowed. It is specified by a trailing minus sign ('-').
- Comments may be specified via SYSIN together with SDAID statements or as separate comment lines.

  A '/*' sign specifies the begin of a comment. All text from the '/*' sign up to the end of the line is treated as a comment. '/*' must not start in column 1.

Start SDAID Trace Initialization

[//] **EXEC SDAID | SDAID**

You start the initialization process with the statement 'SDAID'.
If you want to set-up the trace from the AR, type only 'sdaid'.
If you want to initialize the trace in a partition via SYSLOG, enter 'exec sdaid' in that particular partition.
Submit 'exec sdaid' or '// exec sdaid' if you use SYSRDR as input device.

End the SDAID Trace Initialization

READY	In the Attention Routine
/*	In a Partition via SYSIPT
EOB	In a Partition via Console

You end the initialization process with the statements 'READY' or '/*'. Now you can start the initialized trace with the AR command STRTSD.

Define the Output Device in Direct Input Mode

OUTDEV [BUffer=nn][Tape=cuu|Printer=cuu}

Device	Buffer	'Output when:'	'SDAID Statements'	Note
Printer	no	immediatelv	OUTDEV P=cuu	1
Tape	yes	3K buffer full	OUTDEV T=cuu	2
–	yes	–	OUTDEV BU=nn	3
Printer	yes	certain event occurs	OUTDEV BU=nn P=cuu TRACE type OUTP=BU	4
Tape	yes	certain event occurs	OUTDEV BU=nn T=cuu TRACE type OUTP=BU	5

Notes:

1. No buffer is allocated. The event records are printed on the printer with the device address cuu.

2. The event records are written into a 3K bytes buffer. This buffer is written to a tape mounted on the device cuu when it is full or when an ENDSD or STOPSD command is issued.

3. A trace defined with this OUTDEV statement writes the event records into a wrap-around buffer. You can retrieve the trace records only with the attention routine command: **DUMP BUFFER,cuu**

4. A trace defined with this OUTDEV and TRACE statement prints the contents of the buffer when the event (type), defined with the TRACE statement, occurs.

5. The defined wrap-around buffer is written to tape (cuu) when the trace event (type) occurs.

```
----------------------------------------------------------------------
TRACE BRanch ARea=  [-]
            [OUTPut=]  [-]
            [OPTion=]
----------------------------------------------------------------------
TRACE BUffer[OPTion=]
----------------------------------------------------------------------
TRACE CAncel ARea=      [-]
            [OUTPut=]   [-]
            [OPTion=]
----------------------------------------------------------------------
TRACE EXTernal [KEY][TIMER][SIGNAL]  [-]
               [OUTPut=]             [-]
               [OPTion=]
----------------------------------------------------------------------
TRACE INSTruction=opcode|(opcode1 opcode2 ..)|*|BRanch ARea=  [-]
                                                [OUTPut=]  [-]
                                                [OPTion=]
----------------------------------------------------------------------
TRACE IO [ARea=BG|Fn|SUPvr|ALL] [-]
         [UNit=|CHannel=|CU=]   [-]
         [OUTPut=]              [-]
         [OPTion=]
----------------------------------------------------------------------
TRACE MONitorcall=mc|(mc1 mc2 ..)|* ARea=      [-]
                                   [OUTPut=]   [-]
                                   [OPTion=]
----------------------------------------------------------------------
TRACE PAGing ARea=      [-]
         [OUTPut=]      [-]
         [OPTion=]
----------------------------------------------------------------------
TRACE PGMCheck=pgmc|(pgmc1 pgmc2 ..)|* ARea= [-]
                                   [OUTPut=]  [-]
                                   [OPTion=]
----------------------------------------------------------------------
TRACE PGMLoad [ALL|REQ|HDL] [PH=phase] ARea=   [-]
                                      [OUTPut=]  [-]
                                      [OPTion=]
----------------------------------------------------------------------
TRACE REGister=reg|(reg1 reg2 .. reg8) [PATTern=pp|(pp nn)]  [-]
         ARea=          [-]
         [OUTPut=]      [-]
         [OPTion=]
----------------------------------------------------------------------
TRACE SIO [ARea=BG|Fn|SUPvr|ALL]  [-]
          [UNit=|CHannel=|CU=]                [-]
          [OUTPut=]                           [-]
          [OPTion=]
----------------------------------------------------------------------
TRACE STorage [PATTern=xxxxxxxx] ARea= ADDress=  [-]
                                  [OUTPut=]  [-]
                                  [OPTion=]
----------------------------------------------------------------------
```

```
TRACE SVC=svc|(svc1 svc2 ..)|* ARea=  [-]
                                [OUTPut=]     [-]
                                [OPTion=]
```

```
TRACE VTAMBU [OUTPut=]  [-]
             [OPTion=]
```

```
TRACE VTAMIO ARea=BG|Fn    [-]
                  [UNit=|CHannel=|CU=]  [-]
                  [OUTPut=]             [-]
                  [OPTion=]
```

Additional Definitions

Operand	Function
ARea	Limit tracing to a certain system area.
ADDress	Limit tracing to a certain address range.
OFFset	Limit tracing in a partition or phase area.
PHase	Limit tracing to a certain phase.
LTA SVA	Define additional trace area.
UNit	Define the device address.
CHannel	Define the channel address.
CU	Define the control unit address.
OUTPut	Define additional trace output.
OPTion	Define additional trace options.

```
ARea=partition-id [ADDress=][LTA|SVA|(LTA SVA)]|
    =partition-id [OFFset=][LTA|SVA|(LTA SVA)]|
    =partition-id [PHase= [OFFset=]][LTA|SVA|(LTA SVA)]|
    =SUPvr [OFFset=|ADDress=|PTA]|
    =ALL [ADDress=]|
    =space-id [ADDress=]
```

```
ADDress=addr1:addr2|addr1:*|0:*
```

```
OFFset=reladdr1:reladdr2|reladdr1:*|0:*
```

```
PHase=phase-name [OFFset=]
```

```
LTA|SVA|(LTA SVA)
```

```
UNit=cuu|(cuu cuu)|(cuu-cuu)
```

```
CU=cu|(cu cu)|(cu-cu)
```

```
CHannel=c|(c c)|(c-c)
```

```
OUTPut=definition|(definition1 definition2)
```

```
OPTion=definition|(definition1 definition2)
       HALT
       NOJCL
       TERMinate
       OCCurrence=occ1:occ2
```

Output Definition Summary

Definition	What it records/prints in addition:
BUffer	Contents of SDAID output buffer
CCB	CCB or IORB (TRACE=IO, SIO, or VTAMIO only)
CCW	CCWs (TRACE=IO,SIO, or VTAMIO only)
CCWD=nnnn	CCWs plus nnnn bytes of data (TRACE=IO, SIO, or VTAMIO only)
COMReg	Partition communication region
CREG	Control registers
DUMP	Virtual or real storage
FREG	Floating point registers
GREG	General purpose registers
IOTab	PUB, LUB, ERBLOC, ERRQ, CHANQ
LTA	Logical transient area
LOwcore	Processor storage from zero to X'BC'
PTA	Physical transient area
PIB	Partition information block
PTAB	Partition related control blocks
SUPvr	Supervisor plus GREG and CREG
SYSCom	System communication region
TOD	Time-of-Day clock
TTAB	Task related control blocks

Summary of Trace Procedures

Procedure	'Provides Information on:'
SDBRANCH	successfully executed branch instructions
SDINST	selected or all instruction(s) execution
SDIO	I/O interrupts and SIO instructions
SDLOAD	phase load requests, or actual load
SDPGMC	program check interruptions
SDREG	contents or alterations of 1 to 8 registers
SDSTOR	storage alterations
SDSVC	executed supervisor calls

Trace Initialization Procedures

```
-------------------------------------------------------------------------
// EXEC PROC=SDBRANCH,AREA=partition-id[,OFFSET=|ADDRESS=|PHASE=OFFSET=]
                          =SUPvr[,OFFSET=|ADDRESS=]
                          =ALL|space-id[ADDRESS=]
                        [,OUTPUT= ]
                        [,OPTION=NOJCL|opt|'opt1 opt2 ...']
                        [,BUFFER=nn][,BUFFOUT= ][,TERM=]
                        [,Tape=cuu|Printer=cuu]
-------------------------------------------------------------------------
// EXEC PROC=SDINST,AREA=partition-id[,OFFSET=|ADDRESS=|PHASE=OFFSET=]
                               =SUPvr[,OFFSET=|ADDRESS=]
                               =ALL|space-id[ADDRESS=]
                       [,INST=*|inst|'inst1 inst2 ..']
                       [,OUTPUT=GREG|'outp1 outp2 ...']
                       [,OPTION=NOJCL|'opt1 opt2 ...']
                       [,BUFFER=nn][,BUFFOUT= ][,TERM=]
                       [,Tape=cuu|Printer=cuu]
-------------------------------------------------------------------------
// EXEC PROC=SDIO
             [,UNIT=cuu|'cuu1 cuu2 ..']
             [,AREA=BG|Fn]
             [,OUTPUT='CCWD=256'|'outp1 outp2 ...']
             [,OPTION= ]
             [,BUFFER=nn][,BUFFOUT= ][,TERM=]
             [,Tape=cuu|Printer=cuu]
-------------------------------------------------------------------------
// EXEC PROC=SDLOAD,AREA=partition-id|SUPvr|
                            =ALL|space-id
                      [,PHASE=phasename]
                      [,ADDRESS=0:*|addr1:addr2]
                      [,OUTPUT= ][,OPTION= ]
                      [,BUFFER=nn][,BUFFOUT= ][,TERM=]
                      [,Tape=cuu|Printer=cuu]
-------------------------------------------------------------------------
// EXEC PROC=SDPGMC,AREA=partition-id|SUPvr|
                        =ALL|space-id
                       [ PGMC=*|pgmc|'pgmc1 pgmc2 ..']
                       [,ADDRESS=0:*|addr1:addr2]

                       [,OPTION= ]
                       [,BUFFER=nn][,BUFFOUT= ][,TERM=]
                       [,Tape=cuu|Printer=cuu]
-------------------------------------------------------------------------
```

```
------------------------------------------------------------------------
// EXEC PROC=SDREG,REG=reg|'reg1 reg2 ..'
                  ,AREA=partition-id[,OFFSET=|ADDRESS=|PHASE=[OFFSET=}]
                                  =SUPvr[,OFFSET=|ADDRESS=]
                                  =ALL|space-id[ADDRESS=]
                        [,PATTERN=xxxxxxxx|'xxxxxxxx yyyyyyyy']
                        [,OUTPUT= ]
                        [,OPTION= ]
                        [,BUFFER=nn][,BUFFOUT= ][,TERM=]
                        [,Tape=cuu|Printer=cuu]
------------------------------------------------------------------------
// EXEC PROC=SDSTOR,AREA=partition-id[,OFFSET=|ADDRESS=|PHASE=OFFSET=]
                                  =SUPvr[,OFFSET=|ADDRESS=]
                                  =ALL|space-id[ADDRESS=]
                      [,PATTERN=xxxxxxxx]
                      [,OUTPUT= ]
                      [,OPTION= ]
                      [,BUFFER=nn][,BUFFOUT= ][,TERM=}
                      [,Tape=cuu|Printer=cuu]
------------------------------------------------------------------------
// EXEC PROC=SDSVC,AREA=partition-id[,OFFSET=|ADDRESS=|PHASE=OFFSET=]
                              =SUPvr[,OFFSET=|ADDRESS=]
                              =ALL|space-id[ADDRESS=]
                         [,SVC=*|svc|'svc1 svc2 ..']
                         [,OUTPUT= ]
                         [,OPTION= ]
                         [,BUFFER=nn][,BUFFOUT= ][,TERM=]
                         [,Tape=cuu|Printer=cuu]
------------------------------------------------------------------------
```

Additional Keyword Operands in Trace Procedure Statements

Operand	Function
ADDRESS	Limit tracing to a certain address range.
AREA	Limit tracing to a certain system area.
OFFSET	Limit tracing to a partition or phase area.
PHASE	Limit tracing to a certain phase.
OPTION	Define additional trace options.
OUTPUT	Define additional trace areas.
UNIT	Define the device address.
BUffer	Define the size of the output buffer.
BUFFOUT	Define the event to write the buffer.
TERM	Define the event which terminates the trace.
Printer	Define the printer device address.
Tape	Define the tape device address.

Define the Output Device in a Procedure Statement

```
------------------------------------------------------------------------
 [BUffer=nn] [Printer=cuu|Tape=cuu]       (Note 1)
------------------------------------------------------------------------
 BUFFOUT=CANCEL|PGMC|FULL|EXT             (Note 2)
------------------------------------------------------------------------
 TERM=CANCEL|PGMC|EXT                     (Note 3)
------------------------------------------------------------------------
```

Notes:

1. The definition of a large wrap-around buffer may cause a lack of SDAID storage.

2. If you specify BUFFOUT=CANCEL or BUFFOUT=PGMC the keyword operand 'AREA=BG|Fn' has to be specified, too.

3. If TERM=CANCEL or TERM=PGMC is specified, the AREA=BG|Fn has to be specified also.

Input Command Summary

Command	Description
SDAID	Attention routine command to invoke the SDAID program.
OUTDEV	Defines output device for the trace (printer, tape, or buffer).
TRACE	Defines the event(s) to be traced. At least one TRACE command is required; up to ten may be submitted.
READY	Ends input of initialization commands OUTDEV and TRACE.

The various SDAID Commands

SDAID prompts you for output device of the trace when you enter **OUTDEV**.

One OUTDEV definition can be active in the system at one time. Any newly entered OUTDEV command overwrites the existing one.

Enter **TRACE** to be prompted by SDAID for the type(s) of traces you want. Up to ten 'TRACE' commands may be entered in one session.

You end the trace initialization in the attention routine with the **READY** command. When the READY command has been processed, no further OUTDEV or TRACE command can be entered.

Sample SDAID Trace Initialization via the Attention Routine

SDAID Trace Initialization

```
==> sdaid #
    4C05I PROCESSING OF 'SDAID'  COMMAND SUCCESSFUL
==> outdev #
    4C08D SPECIFY OUTPUT DEVICE.  +
==> buffer #
    4C08D SPECIFY SIZE OF WRAP BUFFER.  +
==> 8 #
    4C08D SPECIFY OUTPUT DEVICE FOR BUFFER.  +
==> tape #
    4C08D SPECIFY PHYSICAL ADDRESS OF PRINTER/TAPE.  +
==> 281 #
    4C05I PROCESSING OF 'OUTDEV' COMMAND SUCCESSFUL.
==> trace #
    4C08D SPECIFY TRACE TYPE.  +
==> inst #
    4C08D SPECIFY OP CODES OR * OR BR.
==> * #
    4C08D SPECIFY TRACE AREA.  +
==> bg #
    4C08D SPECIFY TYPE OF LIMITS.  +
==> #
    4C08D SPECIFY ADDITIONAL TRACE AREA.  +
==> #
    4C08D SPECIFY OUTPUT.  +
==> #
    4C08D SPECIFY OPTIONS.  +
==> nojcl #
    4C08D SPECIFY OPTIONS.  +
==> #
    4C05I PROCESSING OF 'TRACE'  COMMAND SUCCESSFUL.
==> ready #
    4C05I PROCESSING OF 'READY' COMMAND SUCCESSFUL.
    AR 015 1I40I READY.
```

Notational Conventions

- SDAID messages (or help information) are shown in upper-case with a message number.
- Responses or commands for you to enter are shown in lower-case. In most cases a short form of the command is also allowed.
- It is possible to abbreviate the SDAID parameters. For example the BRanch trace type specification can be abbreviated in the following way:

  BR BRa BRan BRanc BRanch

  Note that the minimum definition is indicated through capital letters.
- The symbol + indicates you are to press the ENTER key (generally after entering any response or command).

How to Use Help and Cancel in Prompt Mode

- Messages for which you can request additional help information are indicated by a plus sign (+) at the end of the message.

- Request additional help by entering a question mark (?).

- You can cancel data entered for the current command by entering two question marks (??).

  Figure below shows how you can request help information and how the initialization process can be canceled.

```
==> trace  #
    4C08D SPECIFY TRACE TYPE.   +
==> ?  #
    ENTER ONE OF THE FOLLOWING KEYWORDS:
     SVC          PGMCHECK     MONITOR      CANCEL
     INSTR        REGISTER     STORAGE      BRANCH
     PAGING       PGMLOAD      EXTERNAL     BUFFER
     IO           SIO          VTAMIO       VTAMBU
==> ??  #
    4D03I COMMAND CANCELED DUE TO USER REQUEST
```

OUTDEV Command Input Path

```
             ┌───SPECIFY OUTPUT DEVICE
             │
             │          ┌───SPECIFY SIZE OF WRAP BUFFER
             │          │
             │          │        ┌── SPECIFY OUTPUT DEVICE FOR BUFFER
             │          │        │
             V          V        V
OUTDEV>──>O──BUffer──O── nn ──O......... # .............───>;
             │                   │    SPECIFY PHYSICAL ADDRESS
             │                   │         OF PRINTER/TAPE
             │                   │                 │
             │                   │                 V
             │                   ├──Printer──>┌───O── cuu ─────>;
             │                   └──Tape ───>│
             │                                │
             ├──Printer────────────────────>│
             │                                │
             └──Tape──────────────────────────┘
```

TRACE Command Input Path

```
TRACE
 |  ┌── SPECIFY TRACE TYPE
 |  V
 └─>O──BRanch─────────────────────────────────────────────────>┐
    |                                                          |   ┌───┐
    ├──CAncel─────────────────────────────────────────────────>└──>| 2 |
    |              ┌── SPECIFY TYPE OF EXTERNAL INTERRUPT           └───┘
    |              V                                               ┌───┐
    ├──EXTernal────O.......ALL.......┌────────────────────────────>| 6 |
    |              |   ┌──── , ────┐ |                             └───┘
    |              |   V           | |
    |              └─(───┬──KEY──┬─┘ )┘
    |                    ├─TIMER─┤
    |                    └─SIGNAL┘
    |
    |              ┌── SPECIFY OP CODES OR * OR BR
    |              V
    ├──INSTr───────O──────── * ─────────┬─────────────────────────>┐
    |              ├──op-codes──────────┤                          |
    |              └──BRanch────────────┘                          |
    |                                                              |
    |              ┌─── SPECIFY MONITOR CALL CLASS(ES) OR *        |
    |              V                                               |
    ├──MONitorcall─O────────── * ───────────────┬─────────────────>|
    |              └────monitorclass-list───────┘                  |
    |                                                              |
    ├──PAGing─────────────────────────────────────────────────────>|
    |                                                              |
    |              ┌─── SPECIFY PGM INTERRUPT CODE(S) OR *         |
    |              V                                               |
    ├──PGMCheck────O──────── * ─────────────────────────────────-->|
    |              └──pgmc-codes─────────┘                         |
    |                                                              |
    | SPECIFY PGMLOAD TRACE OPTIONS                                |
    |              |                SPECIFY NAME OF PHASE          |
    |              |                TO BE MONITORED, OR PRESS      |
    |              |                      |     END/ENTER          |
    |              |                      |                        |
    |              V                      V                        |
    ├──PGMLoad─────O...... ALL .......┌───O... # .(all)...┌──>└──>┌───┐
    |              |    ┌──── , ───┐  |   └───phasename───┘       | 2 |
    |              |    V          |  |                           └─V─┘
    |              └─(────┬─ HDL ─┬┴─)┘                            AREA
    |                     └─ REQ ─┘                             definition
    |
    |
    |            ┌─── SPECIFY REGISTER NUMBER(S)
    |            |
    |            |                     ┌─── SPECIFY REGISTER
    |            V                     V            PATTERN
    ├──REGister──O──reg-numbers────────O.... # ..(every)....┌──┐
    |                                  ├──pattern1─────────>|  |
    |                                  └──pattern1 pattern2─┘  |
    |                                                          |
    |            ┌─── SPECIFY STORAGE PATTERN                  |
    |            V                                             |
    ├─STorage────O........... # ..(every).┌──────────────────->|
    |            └───────────pattern──────┘                    |
    |                                                          |
    |            ┌─── SPECIFY SVC NUMBER(S) OR *               |
    |            V                                             |    ┌───┐
    ├──SVC───────O──────── * ──────┬───────────────────────────┴───>| 2 |
    |            └──svc-numbers────┘                                └─V─┘
    |                                                        AREA definition
  ┌─┴─┐
  | 1 |
  └─V─┘
```

```
┌───┐
│ 1 │ (TRACE TYPE, cont'd)
└─┬─┘
  │
  │                      ┌── SPECIFY PARTITION ID
  │                      V                                ┌───┐
  ├──IO──────────┬───────O....ALL...........>┌──────────>│ 5 │ I/O definition
  │              │       ├────partition-id──>│           └─V─┘
  ├──SIO─────────┘       └────SUPvr──────────┘
  │
  │
  │                      ┌── SPECIFY PARTITION ID
  │                      V                                ┌───┐
  ├──VTAMIO──────────────O────partition-id──>──────────>│ 5 │ I/O definition
  │                                                      └─V─┘
  │
  │                                                      ┌───┐
  ├──VTAMBU─────────────────────────────────────────────>│ 6 │ Trace output
  │                                                      └─V─┘ definition
  │
  │                                                      ┌───┐
  └──BUffer─────────────────────────────────────────────>│ 8 │ Occurrence
                                                         └─V─┘ definition

     ┌── SPECIFY            ┌── SPECIFY TYPE OF LIMITS
     │   TRACE AREA         │           ┌── SPECIFY OFFSET RANGE
     V                      V           V
┌───┐
│ 2 │>├──partition-id──>O..OFFset....>O.........0:*.............┐
└───┘ │                 │           ├──reladdr1:*───────────>│
      │                 │           └──reladdr1:reladdr2─────>│
      │                 │                                      │
      │                 │           ┌── SPECIFY ADDRESS RANGE  │
      │                 │           V                          │
      │                 ├──ADDress──>O......0:*...............>│
      │                 │           ├──addr1:*──────────────>│
      │                 │           └──addr1:addr2───────────>│
      │                 │                                      │   ┌───┐
      │                 │                                      └─>│ 4 │
      │                 │                                          └───┘
      │                 │           ┌── SPECIFY NAME OF PHASE
      │                 │           V                              ┌───┐
      │                 └──PHase───>O────phasename────────────>│ 3 │
      │                                                            └───┘
      │                 ┌── SPECIFY TYPE OF LIMITS
      │                 │           ┌── SPECIFY OFFSET RANGE
      │                 V           V                              ┌───┐
      ├─SUPvr───────────O...>OFFset...O..........0:*............┌──>│ 6 │
      │                 │           ├──reladdr1:*───────────>│   └─V─┘
      │                 │           └──reladdr1:reladdr2────>│
      │                 │                                     │
      │                 │           ┌──SPECIFY ADDRESS RANGE │
      │                 │           V                         │
      │                 ├──ADDress──>O......0:*..............>│
      │                 │           ├──addr1:*─────────────>│
      │                 │           └──addr1:addr2──────────>│
      │                 │                                     │
      │                 └────>PTA────────────────────────────┘
      │
      │                  ┌─── SPECIFY ADDRESS RANGE
      │                  V                                          ┌───┐
      ├──space-ID──┬─────O........0:*..........┬─────────────────>│ 6 │
      │            │     ├──addr1:*──────────>│                    └─V─┘
      │            │     └──addr1:addr2───────┘
      └─────ALL────┘
  SPECIFY OFFSET RANGE                ┌───┐     ┌── SPECIFY ADDITIONAL
            |                         │ 4 │     |   TRACE AREA
            V                         └─┬─┘     V
┌───┐       |                           V       V                        ┌───┐
│ 3 ├───>O.........0:*..........┬──────────>O....... # .(none)┬──>│ 6 │
└───┘     ├──reladdr1:*──────────>┤          |  ┌─── , ───┐     |    └─V─┘
          └──reladdr1:reladdr2────┘          |  V         |     |    Output de-
                                             └─(───┬─LTA─┬─┴──)─┘    finition
                                                   └─SVA─┘


                 ┌─── SPECIFY RANGE OF IO/SIO TRACE
                 V
┌───┐                                                                ┌───┐
│ 5 ├──────>O..... # ......(all units traced).........┬────────>│ 6 │
└───┘            |                                     |             └───┘
 I/O             |            ┌─SPECIFY CHANNEL ADDRESS |
definition       |            V                        |
                 ├────CHannel───O──channel address(es)─┬────>|
                 |              └──────────  ,  ───────┘      |
                 |                                            |
                 |              ┌── SPECIFY CU ADDRESS        |
                 |              V                             |
                 ├────CU────────O──control unit(s)─────┬────>|
                 |              └──────────  ,  ───────┘      |
                 |                                            |
                 |              ┌── SPECIFY UNIT ADDRESS      |
                 |              V                             |
                 └────UNit──────O──device address(es)──┬─────┘
                                └──────────  ,  ───────┘


             ┌── SPECIFY OUTPUT
             V
┌───┐                                                                   ┌───┐
│ 6 ├───>O......................... # ...(no additional output)..┬─>│ 8 │
└───┘        |                      ┌───────────  ,  ──────────┐  |    └─V─┘
trace        |                      V                           |  |   Option
output       └──────────────────(───┬─GReg──────────────────────┬─┴─)─┘   defi-
definition   OUTPut                 ├─CReg────────────────────>|         nition
                                    ├─FReg────────────────────>|
                                    ├─COMReg──────────────────>|
                                    ├─SYSCom──────────────────>|
                                    ├─IOTab───────────────────>|
                                    ├─LTA─────────────────────>|
                                    ├─PTA─────────────────────>|
                                    ├─PIB─────────────────────>|
                                    ├─CCB─────────────────────>|
                                    ├─CCW─────────────────────>|
                                    ├─TOD─────────────────────>|
                                    ├─LOwcore─────────────────>|
                                    ├─Supvr───────────────────>|
                                    ├─BUffer──────────────────>|
                                    ├─PTAB────────────────────>|
                                    ├─TTAB────────────────────>|
                                    |      ┌─SPECIFY LENGTH OF DATA |
                                    |      V                       |
                                    ├─CCWD-O─── data length ─────>|
                                    |                              |
                                    |      ┌───SPECIFY DUMP AREA   |
                                    |      V                       |   | See also
                                    └─DUMP-O─── dump definition >──┘   | Note 1 and
                                                 |          ┌───┐      | 3 at the
                                                 └──see──>│ 7 │      | end of the
                                                            └─V─┘      | dump defi-
                                                                       | nition
```

```
      ┌─── SPECIFY DUMP AREA (see Note 1)
      |
      |
      |                        ┌─── SPECIFY OFFSET RANGE
┌───┐ V                        V
| 7 |─O─partition─id───────────O......0:*.................┐
└───┘ |  (partition            ├─relad1:*───────────────>|
      |     ─id)               └─relad1:relad2──────────>|
      |                                                   |
      |                                                   |
      |                        ┌─── SPECIFY OFFSET RANGE  |
      |                        V                          |
      ├─PHase──────────────────O.......0:*...............>|
      |  ──┬──                 ├──relad1:*──────────────>|
      |    |                   └──relad1:relad2─────────>|
      |    └───see Note 2                                 |
      |                                                   |
      |                        ┌─── SPECIFY ADDRESS RANGE |
      |                        V                          |
      ├─ADDress────────────────O──────0:*───────────────>|
      |                        ├──addr1:*───────────────>|
      |                        └──addr1:addr2───────────>|
      |                                                   |
      |                                                   |
      |                        ┌─── SPECIFY DUMP RANGE    |
      |                        V                          |
      ├─REG────────────────────O───reg:bytes────────────>|
      |                                                   |
      |                                                   |
      |        ┌── SPECIFY REGISTER       ┌──SPECIFY OFFSET |
      |        |   AND OFFSET             |  AND DUMP LENGTH|
      |        V                          V               |      ┌───┐
      └─PTR──O──reg:offset──────────────O──offset:length──┴────>| 8 |
                                                                 └─V─┘
```

Notes:

1. Up to ten different areas may be specified with DUMP.

2. Can be specified only if a phase was previously defined in the area definition of the TRACE.

3. You need not specify the word OUTPut in prompt mode. SDAID prompts you for the definition of the additional output.

```
┌───┐
| 8 |
└─┬─┘
  |  ── SPECIFY OPTIONS
  |  V
  └──O........ # ..(no-option).........................┬───> ;
     |                                                  |
     |  ┌──────  ,  ───────────────────────────────┐   |
     |  V                                           |   |
     |  ┌──────HALT───────────────────────────┐    |   |
     |  |                                      |    |   |
     |  ├──────NOJCL─────────────────────────>|    |   |
     |  |                                      |    |   |
     └(─O───TERMinate────────────────────────>┼────┘─)┘
        |                ┌── SPECIFY OCCURRENCE |
        |                V   RANGE            |
        └───OCCURRENCE───O──────── # ──(all)──>|
                         ├──────────1:*───────>|
                         └──────val1:val2──────┘
```

Output Device Definition in Prompt Mode: OUTDEV Command

As a response to prompting message,

```
------------------------------------------------
 4C08D SPECIFY OUTPUT DEVICE. +
------------------------------------------------
```

enter one of the following:

Printer If printer is specified, the event records are written to a line printer at the time the particular event occurs.

Tape When tape is defined, the trace records are written to tape in the form of 3K bytes blocks.

SDAID prompts you for the address of the output device in the following way:

```
------------------------------------------------
 4C08D  SPECIFY PHYSICAL ADDRESS OF PRINTER/TAPE.
------------------------------------------------
```

Enter the device address in the form **cuu**.

BUffer writes the trace output to a wraparound buffer. SDAID prompts you for the size of the buffer as follows:

```
------------------------------------------------
 4C08D  SPECIFY SIZE OF WRAP BUFFER.  +
------------------------------------------------
```

Enter the desired size of the buffer in number of blocks of 1K bytes.

Possible Buffer Sizes:

The possible buffer sizes depend on the output device for the buffer which is defined next.

Buffer to Printer or no output device	3K - 99K
Buffer to Tape	3K - 32K

Now, SDAID prompts you as follows:

```
------------------------------------------------
 4C08D  SPECIFY OUTPUT DEVICE FOR BUFFER.  +
------------------------------------------------
```

Respond with either **Printer, Tape, or END/ENTER.**

Pressing END/ENTER causes no output device being defined.

SDAID Wait States

Wait due to OPTION=HALT

If a TRACE command is specified with OPTION=HALT then the system enters a wait state at the occurrence of the specified event. The address part of the wait PSW contains the value X'EEEE'. The operator may now respond in two ways:

1. Press the external interrupt key to continue processing. The system will enter the wait state again on the next occurrence of that traced event.

2. Enter X'FF' in storage location zero, then press the external interrupt key. That removes the OPTION=HALT specification. The system continues tracing but does no more enter the wait state X'EEEE' on the occurrence of the specified event.

Wait due to I/O Error conditions on SDAID Output Device

If an I/O error condition occurs on the SDAID output device, then SDAID performs its own error recovery. If the OUTDEV device is a printer and is not ready then SDAID waits two minutes to allow the operator to make the printer ready. On all other error conditions SDAID performs a reasonable number of retries. If the retry counter is exhausted, or the wait time expires, then SDAID loads a wait PSW with a value of X'EEEEEE' in the address part of the PSW. When the SDAID program has entered a wait state with X'EEEEEE' in the address part of the PSW then the operator should inspect bytes 0-3 of low core to find the appropriate error recovery action. Bytes 0-1 of low core display an error code of the form X'62xx'. Bytes 2-3 show the device address of the SDAID output device.

Here is a summary of all possible error codes which may be displayed in bytes 0-1 of low core.

X'62C1'	End of tape condition on output tape
X'62C2'	Device not operational
X'62C3'	Missing device end interrupt
X'62C4'	Control unit busy
X'62C5'	Intervention required - device not ready
X'62C6'	Channel error
X'62C7'	BUSOUT check
X'62C8'	UCS Parity|data converter check
X'62C9'	Sense missing or wrong

X'62D1'	Attention
X'62D2'	Command reject
X'62D3'	open error
X'62D4'	Invalid I/O function requested
X'62D5'	Stack module terminate
X'62D6'	No print buffer available
X'62D7'	max.number of write retries exceeded
X'62D8'	max.number of SIO retries exceeded
X'62D9'	Unusual command sequence

The operator should now try to lift the error condition (e.g. ready the printer or tape device, fill new paper or mount another tape reel). If this is possible then press the external interrupt key once to continue tracing operations.

If the error condition cannot be cleared (e.g. the SDAID output device remains not-operational) then press the external interrupt key twice to stop the SDAID data collection and restart VSE processing. After that (preliminary) stopping of SDAID a STOPSD or an ENDSD command should be given for a final termination of SDAID.

Types of Dump (Overview)

Initiated by/via	Output Contents	Output Device	Initiated/Controlled by
System (ABEND)	System- or Partition-Dump	Dump Sublib. or SYSLST	SYSTEM OPTIONS to Request the Dump
Operator (Console)	System- or Partition-Dump	Dump Sublib. or SYSLST	CANCEL Command
Operator (Console)	Selected Storage Areas	Tape or Printer	DUMP Command
Operator (Console)	System Storage	Tape or Printer	STAND-ALONE DUMP Program
Programm./Oper. (Defined Event)	Defined Storage Areas	Tape, Printer or Buffer	SDAID Dump Trace
Programmer (Macro)	Macro Dependent	Macro Dependent	MACROS (PDUMP, DUMP, JDUMP)

ABEND Dump Function

What is an ABEND

ABEND stands for ABnormal END of task. This means that a program (task) is terminated prior to its completion because of an error that could not be resolved by system recovery facilities.

What is an ABEND Dump

The system's ABEND dump function is called by VSE/Advanced Functions system either at the occurrence of an ABEND or else if a CANCEL command has been issued. Once the function has been called, a dump of virtual storage is provided.

- The **System-Dump** dumps the whole supervisor area and the dump symptoms besides the partition area.
- The **Partition-Dump** includes only selected VSE/Advanced Functions control blocks and the dump symptoms in addition to the partition area.

The output device for the dump data is a dump sublibrary or a printer device assigned to SYSLST.

Contents of the ABEND Dump Output

The output of the ABEND dump function either in the sublibrary or on SYSLST contains:

- A dump symptom part which is always included.
- A part which is either a system-dump or a partition-dump, depending on the options active at the time the dump was taken.

Symptom Part of the ABEND Dump

This part of the output contains

- Control data from the symptom record, like information about the environment or the failure.

System- and Partition-Dump Contents

A system dump is produced if OPTION DUMP or STDOPT DUMP=YES is active.
A partition dump is produced when option OPTION PARTDUMP or STDOPT DUMP=PART is active.
The listed VSE/AF System Areas are contained beside the symptom part in:
S - the System Dump
P - the Partition Dump

S,P The ending task PSW, general purpose and floating point registers.

S,P If the error occurred in the SVA, that part of the SVA which holds the phase responsible for the ABEND.

S,P The partition for which the ABEND dump function is active including areas acquired dynamically within the partition by GETVIS macros in your program.

S The entire supervisor area.

S The addresses of the VSE/Advanced Functions control blocks.

S The allocated portion(s) of the system GETVIS area.

P The LOWCORE (low address storage): (188 bytes - X'00' to X'BC')

P The areas containing VSE/Advanced Functions control blocks.

P The logical transient area (LTA), if the error causing the dump to be taken occured in a task owning the LTA.

The DUMP Command Dump

You can request a dump of parts of or of all the virtual storage with the attention routine command DUMP.

The Stand-Alone Dump

You can request a dump of the applicable VSE/Advanced Functions virtual storage (a stand-alone dump) with the stand-alone dump program. The program (generated by DOSVSDMP) can be called from cards, diskette or tape.

Stand-Alone Dump Program Output on Tape

The stand-alone dump program stores the dump information on the tape from where it has been loaded. The dump information on tape contains the

- symptom record which holds information on the hardware and software environment, error symptoms and control block locators.
- dump data, which are retrieved pages from processor storage, or from the page data set.
- last 200 messages from the hard copy file.

You can print the dump output either with the Info/Analysis program or with the DOSVSDMP utility.

Note: Only active address space will be dumped in a VAE-system.

The SDAID Dump

The SDAID program can also be used to dump virtual storage. You may use this program for example if you need a dump of a certain part of storage at a defined event.

Dump Requested by Macros

A dump of virtual storage can also be requested through dump macros.

DOSVSDMP AND STAND-ALONE DUMP

The DOSVSDMP Utility

The DOSVSDMP utility includes the following functions:

- Generation of the stand-alone dump program on tape, diskette, or cards.
- Print the stand-alone dump from tape.
- Print the DUMP command dump from tape.
- Format and print the SDAID output tape.

Generate the Stand-Alone Dump Program

You can generate the stand-alone dump program to reside on a **magnetic tape**, if you want to have a quick save of the dumped information for later processing.

The stand-alone dump program can also be generated to reside on **diskette** or in **cards** (no dump information saved for later retrieval since output goes to a printer).

To generate a stand-alone dump program, invoke DOSVSDMP by entering

```
// EXEC DOSVSDMP
```

The program, once it receives control, prompts you for further control information, and you select the desired option by an appropriate response to the program's prompt as shown.

```
Prompt message
xx 4G01D SELECT ONE OF THE FOLLOWING FUNCTIONS
.............................................
1 CREATE STAND ALONE DUMP PROGRAM
2 PRINT DUMP TAPE
3 PRINT SDAID TAPE
R END DOSVSDMP PROCESSING
```

Enter **1** to invoke dump program generation.

The DOSVSDMP utility responds with

```
Prompt message
xx 4G04D SPECIFY OUTPUT DEVICE IN THE FORM SYSNNN OR CUU
.........................................................
```

The device defined with SYSNNN or CUU can be a:

- Tape device
- Card punch
- Diskette device

If the selected device is a tape drive, it must be either 9-track or 7-track with data converter.

Note: Neither the utility DOSVSDMP nor the generated stand-alone dump program supports the IBM 8809 tape drive in streaming mode.

The completion message

```
┌──────────────────────────────────────────┐
│ xx 4G09I DUMP PROGRAM HAS BEEN CREATED   │
│..........................................│
└──────────────────────────────────────────┘
```

indicates the successful generation of the dump program.

You need not regenerate the stand-alone dump program when a tape has been used. The dump program remains useable for all subsequent stand-alone dump requests.

The kind of dump program generated depends on the mode of system operation. For example, a stand-alone dump program generated under 370 mode, can be used to dump also an address space of a VSE system operating in ECPS:VSE mode and vice versa.

Generating the Program on Diskette or Cards

If you use a diskette or a card unit you will be prompted for a printer address.

```
┌────────────── Prompt message ──────────────┐
│ xx 4G06D SPECIFY PRINTER ADDRESS IN FORM CUU │
│ ............................................ │
└────────────────────────────────────────────┘
```

You may respond with

- the unit address of that printer which you want to use for the dump output at stand-alone dump program execution time.
- end/enter if you want to choose the printer address at execution time.

Note: A stand-alone dump program generated under 370 mode which resides on a diskette or on cards cannot be used in E/mode and vice versa.

If you do not specify a printer address, the generated stand-alone dump program will enter a wait state until the operator starts the printer device which is to be used to print the stand-alone dump.

Dump Tape printed with DOSVSDMP

Normally Info/Analysis is used to process and print dump tapes. In exceptional cases the use of the DOSVSDMP utility may be necessary, for example, if none of your dump sublibraries is big enough to hold the stand-alone dump.

The output of a stand-alone dump taken on a central processor with ECPS:VSE can be printed by DOSVSDMP under a VSE system in 370 mode and vice versa.

The printed output of the DOSVSDMP utility contains for both, DUMP command tape or stand-alone dump tape, the following:

- Symptom record.
- Unformatted dump data.

Sample DOSVSDMP Print Set-up

```
// JOB DOSVSDMP
// EXEC DOSVSDMP
```

DOSVSDMP prompts you by messages at SYSLOG to define the operation you want to perform, with:

```
┌─────────────────── Prompt message ─────────────────────┐
│ xx 4G01D SELECT ONE OF THE FOLLOWING FUNCTIONS         │
│ ...............................................        │
│ 1 CREATE STAND ALONE DUMP PROGRAM                      │
│ 2 PRINT DUMP TAPE                                      │
│ 3 PRINT SDAID TAPE                                     │
│ R END DOSVSDMP PROCESSING                              │
└────────────────────────────────────────────────────────┘
```

Enter **2** to invoke DOSVSDMP print dump tape processing.

The DOSVSDMP utility response is:

```
┌─────────────────── Prompt message ─────────────────────┐
│ xx 4G02D SPECIFY THE TYPE OF THE DUMP                  │
│ ....................................                   │
│ 1 STAND ALONE DUMP ON TAPE                             │
│ 2 DUMP COMMAND ON TAPE                                 │
│ R END DOSVSDMP PROCESSING                              │
└────────────────────────────────────────────────────────┘
```

(ENTER ONE OPTION ONLY)

Enter **1** to print the stand-alone dump or
enter **2** to print the DUMP command dump on SYSLST.

The DOSVSDMP utility response is:

```
┌────────────────────── Prompt message ──────────────────────┐
│ xx 4G04D SPECIFY INPUT TAPE IN THE FORM SYSNNN OR CUU      │
│.............................................................│
└────────────────────────────────────────────────────────────┘
```

Enter 280, for example, if the dump tape is mounted on the tape drive 280.

The DOSVSDMP utility starts printing on the device at SYSLST if the tape contains a **stand-alone dump**.

After print completion, control is returned to Job Control.

If the dump tape contains a **DUMP command dump** the DOSVSDMP utility responds with:

```
┌─────────────────── Prompt message ─────────────────────┐
│ xx 4G30D SPECIFY FILENUMBER ON TAPE                    │
│ ...................................                    │
└────────────────────────────────────────────────────────┘
```

Enter 2, for example, if the second file contains the DUMP command output you want to print.

Now the DOSVSDMP utility starts printing the DUMP command tape on SYSLST.
After print completion, control is returned to Job Control.

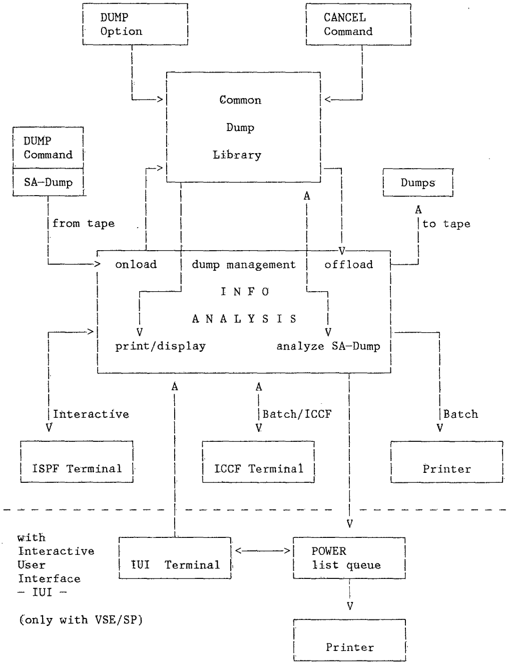
DOSVSDMP Wait Codes

The following wait state codes appear in the address portion of the current PSW when the stand-alone dump program (DOSVSDMP) terminates. The codes are shown here as they would appear in the right half of the current PSW when displayed using the ALTER/DISPLAY function.

Wait Codes (Hex)	
CE 00 00	Successful completion of a stand-alone dump request.
CE 00 01	An I/O error occured after a start I/O was issued for the dump device.
CE 00 02	The dump-output device is not operational.
CE 00 04	A channel error occured on the dump-output device.
CE 00 08	A permanent I/O error occured on the dump-output device. The original error was retried and found to be irrecoverable. When this code occurs, the stand-alone dump program stores 24 bytes of sense information at processor storage location 0.
CE 00 10	An I/O error occured during error recovery processing. This indicates an error other than that for which error recovery is being performed. When this code occurs, the stand-alone dump program stores 24 bytes of sense information at processor storage location 0.
CE 00 20	End of tape.
CE 00 40	An I/O error occured during the initial program load of the stand-alone dump program.
CE 00 80	A machine check occured.
CE 01 00	I/O error during IPL.
CE 02 00	No page frame available during IPL.
CE 04 00	A program check occured during IPL of the dump program.
CE 08 00	A program check occured during virtual storage dump preparation.
CE 10 00	A program check occured while dumping virtual storage.
CE 20 00	A program check occured while shifting the dump program (DMPPROG) to the storage limit.
CE 40 00	A program check occured while creating the VSE control block or HC message symptom record section 6 (module IJBXDM8 or 9).
CE 80 00	No E-mode page frames are available for shifting the dump program (DMPROG).

INFO/ANALYSIS

Schematic Overview



The Info/Analysis Execution Modes

B A T C H
MODE

A Input/Output via:
- ICCF Terminal

B Input:
- SYSIN/SYSLOG

Output:
SYSLST

I N T E R A C T I V E
(ISPF) MODE

Input/Output via:
- ISPF/ICCF Terminals

You initiate Info/Analysis by calling the ISPF start-up procedure. Enter the name of the procedure on the ICCF terminal command line.

Common DUMP Library SYSDUMP

The library named SYSDUMP is used to store the various dump types for further processing. It contains one or more dump sublibraries. Each dump sublibrary should be assigned to one partition and may contain one or more dumps.

```
                           SYSDUMP LIBRARY
                       ┌───────────────────────┐
                  ┌──>│ SYSDUMP.BG.....       │ <──┐
                  │   │───────────────────────│    │
  Dump            ├──>│ SYSDUMP.F1.....       │    │
  Sublibraries:   │   │───────────────────────│    │
  one per         ├──>│ SYSDUMP.F2.....       │    │
  Partition       │   │───────────────────────│    │
  (if created)    ├──>│ ......                │    │
                  │   │───────────────────────│    │
                  └──>│ ......                │    │
                       └───────────────────────┘    │
                                                    │
                        DUMP SUBLIBRARY FOR BG      │
                       ┌───────────────────────┐ ─┐ │
  Dumps stored    ┌──>│ SYSDUMP.BG.00000020   │  │ │
  in one Dump     │──>│ SYSDUMP.BG.DUMP       │  ├─┘
  Sublibrary      └──>│........               │  │
                       └───────────────────────┘ ─┘
```

Info/Analysis in Batch

You may use Info/Analysis in batch mode in two ways:

- Line mode - from the operator console or ICCF console
- Reader mode - from the system input device or ICCF virtual reader

In either case, you invoke Info/Analysis by submitting a series of job control statements (JCL) followed by control statements that request Info/Analysis functions. All output is routed to the SYSLST device or the ICCF virtual printer called the print area. The output includes the input control statements, the results of processing, and any messages issued by Info/Analysis.

You may submit the job either in line mode by entering statements on the console, or in reader mode by submitting a job to the system input device.

With JCL, you must specify any nonstandard system device assignments and preallocate and assign any files that you require other than the system libraries. A sample of the JCL for invocation is:

```
// JOB    ANALYSIS
// ASSGN  SYSLST,00E
// ASSGN  SYS016,252    " DUMP MANAGEMENT FILE "
// ASSGN  SYS017,252    " EXTERNAL ROUTINE FILE "
// EXEC   INFOANA,SIZE=400K

(Info/Analysis control statements)

/*
/&
```

Once the JCL has been processed, you are at the selection level in Info/Analysis. The program reads for your control statements. An end of input (/*) statement marks the end of these statements. To end your job, enter an end of job (/&) statement.

Entering Control Statements

The rules for entering the control statements that request Info/Analysis functions are:

- Each card or input line may contain only one control statement.
- A control statement may begin in any column.
- Control statements and their operands may be entered in uppercase or lowercase.
- Control statements must be entered in their complete form; no abbreviations are allowed.
- Each word in a control statement must be a contiguous string of characters.
- Some blanks, at least one, must appear between the words in a statement.
- A blank followed by an asterisk (' *') signifies the start of a comment. If the first non-blank character in any control statement is an asterisk, the entire statement is treated as a comment.

If you enter an invalid control statement in reader mode, the remaining control statements in the job are flushed and the session is canceled. The output indicates the erroneous statement.
In line mode, invalid input causes Info/Analysis to flush only the control statement. You may then reenter the statement correctly.

Ending the Info/Analysis Job

You end an Info/Analysis batch job by submitting the SELECT END statement while you are at the selection level. If you wish to end your session at the function level, enter RETURN followed by SELECT END.

SELECT END should be followed by an end-of-input statement (/*) and an end-of-job statement (/&). If you enter an end-of-input or end-of-job statement at any point in the sequence, the job is canceled at that point.

Command Summary

This overview contains a summary of the batch control statements for Info/Analysis. The statements are presented in a functional order.

Function Selection Level Commands:

DUMP NAME dumpname

The DUMP NAME command specifies the dump to be analyzed, onloaded, or offloaded.

If the dump name is not currently in the list of known dumps, it is added in 'TO BE ONLOADED' status.
This dump name is required for any subsequent action that requires an online dump or a dump name.

SELECT DUMP MANAGEMENT

Select this function to manage the list of dumps known to Info/Analysis.

SELECT DUMP SYMPTOMS

Select this function to print the symptom record of the dump.

SELECT DUMP VIEWING

Select this function to print the contents of the dump records or invoke an analysis routine.

SELECT DUMP OFFLOAD
Select this function to offload a dump to a tape for storage or backup.

SELECT DUMP ONLOAD
Select this function to onload a dump to the dump sub-library from a tape.

SELECT END
Select this function to cause termination of Info/Analysis.

Dump Management Commands:

DUMP NAME dumpname
The DUMP NAME command specifies the dump to be analyzed, onloaded, or offloaded.

If the dump name is not currently in the list of known dumps, it is added in 'TO BE ONLOADED' status.
This dump name is required for any subsequent action that requires an online dump or a dump name.

UTILITY
The UTILITY command provides proper initialization of a new or re-allocated dump management file.

DELETE
This command causes the currently active dump to be deleted form the dump sub-library and the list of known dumps.

PRINT DATA
The PRINT DATA command causes the list of known dumps to be printed.

Dump Symptoms Commands:

PRINT DATA
The PRINT DATA command causes the list of known dumps to be printed.

Dump Viewing Commands:

PRINT from-address to-address
This format causes dump data to be printed for the specified address range.

PRINT from-address END
This format causes dump data to be printed from the address specified to the end of the dump.

PRINT from-address FOR length
This format causes dump data to be printed from the address specified for the length specified.

PRINT FORMAT
This format causes portions of dump data to be printed according to information in the symptom record.

CALL routine-name
The CALL command initiates the execution of the specified analysis routine.

Dump Offload Commands:

VOLID volume-ID
The VOLID command specifies the 6 digit volume ID of the output tape.

BYPASS
This command bypasses the physical offloading of a dump if a valid copy already exists.

ERASE YES/NO

This command specifies if the dump is to be erased from the sub-library after completion of the offload function. The entry will remain in the list of known dumps until it is deleted.

Dump Onload Commands:

VOLID volume-ID

The VOLID command specifies the 6 digit volume ID of the input tape.

FILE file-sequence-number LAST

The FILE command specifies the sequence number of the file on the tape. The LAST operand is optional and indicates that this file is the last file to be loaded from this tape. Specifying LAST causes the tape drive to be de-allocated.

Commands valid for all Functions:

HELP

A copy of this listing is printed.

RETURN

This command causes the termination of the current function.

Control Statement Sequence Examples

```
1.  SELECT DUMP MANAGEMENT
      PRINT DATA
      RETURN
2.  DUMP NAME SYSDUMP.F6.00000007
3.  SELECT DUMP SYMPTOMS
      PRINT DATA
      RETURN
4.  SELECT DUMP VIEWING
      PRINT 0 20880              * PRINT SUPERVISOR
                                 *        DATA
      PRINT C80000 END           * PRINT SVA
      PRINT 126000 FOR 1800      * PRINT JTB TABLE
      PRINT FORMAT               * PRINT ALL FORMATTED
                                 *            DATA
      RETURN
5.  SELECT DUMP OFFLOAD
      VOLID T02512
      RETURN
6.  SELECT END
```

The example contains the following operations:

1. Select the Dump Management function and request the printing of the list of managed dumps.

2. On the selection level, specify SYSDUMP.F6.00000007 as the current dump.

3. Use the Dump Symptoms function to print the dump symptoms that are contained in the symptom record.

4. Use the Dump Viewing function to print selective areas of the dump. The assumed areas are written in the comments on each statement.

5. Use Dump Offload to offload SYSDUMP.F6.00000007 to the tape with VOLID T02512.

6. End your Info/Analysis session.

```
1.  DUMP NAME SYSDUMP.F4.DUMP0003
2.  SELECT DUMP ONLOAD
      VOLID T300U1
      FILE 003
      RETURN
3.  SELECT DUMP VIEWING
      CALL  IJBXDBUG
      RETURN
4.  DUMP NAME SYSDUMP.F5.00000002
5.  SELECT DUMP OFFLOAD
      VOLID T03417
      BYPASS
      RETURN
6.  DUMP NAME SYSDUMP.F6.DUMP0001
7.  SELECT DUMP ONLOAD
      VOLID T300U1
      FILE 1 LAST
      RETURN
8.  SELECT END
```

The example contains the following operations:

1. On the selection level, specify SYSDUMP.F4.DUMP0003 as the current dump.

2. Use Dump Onload to load the current dump (file 3 on tape T300U1) into the a dump sublibrary so that you can work with it.

3. Use Dump Viewing to call routine IJBXDBUG to analyze the stand-alone dump. Results of the routine are printed.

4. On the selection level, specify SYSDUMP.F5.00000002 as the current dump.

5. Use Dump Offload to offload SYSDUMP.F5.00000002, specifying the output volume and choosing to bypass the write operation because a valid copy of the dump already exists on tape. (The information concerning this dump in the dump management file will be kept.)

6. On the selection level, specify SYSDUMP.F6.DUMP0001 as the current dump.

7. Use Dump Onload to load the current dump (file 1 on tape T300U1) into a dump sublibrary, specifying LAST because it is the last dump to be onloaded from the tape.

8. End your Info/Analysis session.

Note: All of the functions available to the batch user are available to the interactive user. However, the dump viewing function is more flexible in the interactive environment.

Info/Analysis Interactive

Most interactions in Info/Analysis with ISPF take place through panels and responses to those panels. However, a set of commands is provided described in following overview:

Command Summary

This is a summary of the interactive commands for Info/Analysis. The "Valid Functions" column represents the functions during which the commands may be entered as follows:

M	- Dump Management
S	- Dump Symptoms
V	- Dump Viewing
OF	- Dump Offload
ON	- Dump Onload
SEL	- selection level
T	- Tutorial

COMMAND	DESCRIPTION	VALID FUNCTIONS						
		M	S	V	OF	ON	SEL	T
DOWN 'amount' UP 'MAX ' 'HALF ' 'PAGE ' 'CSR '	Scroll the specified or default value in the specified direction	X	X	X				
END	End function or session	X	X	X	X	X	X	X

Find
{'hexdata' }
{chardata }
	Locate data in display	X	X	X				

[from-addr] [from-addr to-addr] [from-addr END] [from-addr FOR length] [INCRement incr]	defines search range							
HELP	Display help	X	X	X	X	X	X	X
Log [ON] [OFF]	Record session interactions	X	X	X	X	X	X	
MASK [from-addr to-addr] [from-addr FOR length]	Overlay dump data			X				
MSG	Display messages	X	X	X	X	X	X	
PRint	Print data							
[FORMAT]				X				
[DATA]		X	X					
[from-addr to-addr]	defines the address range of dump data to be printed			X				
[from-addr END]				X				
[from-addr]				X				
[from-addr FOR length]				X				
RETURN	Return to selection level or panel from which tutorial was called	X	X	X				X

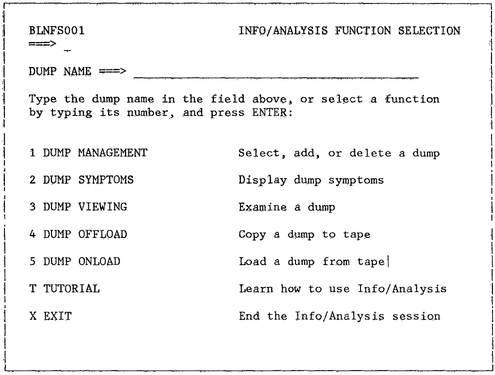
You enter a command by typing it after the arrow (===>) on the entry line and pressing the ENTER key, or by pressing a PF key if the command is represented by one.

Note: For a detailed description of the commands use the 'HELP' command.

Function Selection Panel

The function selection panel shown below is displayed when you invoke Info/Analysis. It is the focal point for all Info/Analysis activities.

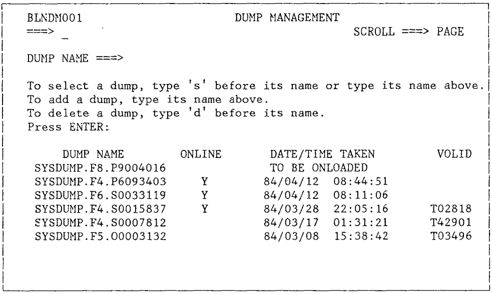
To invoke Info/Analysis, enter the name of the Info/Analysis start-up command procedure.



```
BLNFS001                      INFO/ANALYSIS FUNCTION SELECTION
===> _

DUMP NAME ===> ________________________________________

Type the dump name in the field above, or select a function
by typing its number, and press ENTER:

1 DUMP MANAGEMENT             Select, add, or delete a dump

2 DUMP SYMPTOMS               Display dump symptoms

3 DUMP VIEWING                Examine a dump

4 DUMP OFFLOAD                Copy a dump to tape

5 DUMP ONLOAD                 Load a dump from tape

T TUTORIAL                    Learn how to use Info/Analysis

X EXIT                        End the Info/Analysis session
```

The Dump Management Panel

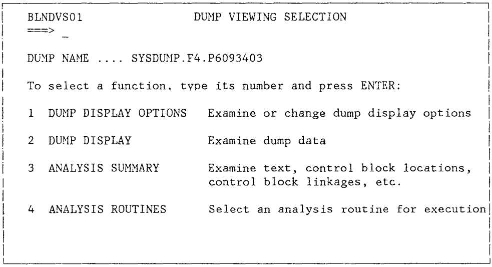
After this task is completed and at any other time that you select Dump Management during a session, Info/Analysis displays the dump management panel, which lists all of the dumps identified in the dump management file. The dumps are listed in reverse chronological order by date and time stamp. That is, the most recent dump is at the top of the list.



```
BLNDM001                      DUMP MANAGEMENT
===> _
                                                 SCROLL ===> PAGE

DUMP NAME ===>

To select a dump, type 's' before its name or type its name above.
To add a dump, type its name above.
To delete a dump, type 'd' before its name.
Press ENTER:

     DUMP NAME         ONLINE       DATE/TIME TAKEN          VOLID
 SYSDUMP.F8.P9004016                 TO BE ONLOADED
 SYSDUMP.F4.P6093403      Y         84/04/12  08:44:51
 SYSDUMP.F6.S0033119      Y         84/04/12  08:11:06
 SYSDUMP.F4.S0015837      Y         84/03/28  22:05:16      T02818
 SYSDUMP.F4.S0007812                84/03/17  01:31:21      T42901
 SYSDUMP.F5.O0003132                84/03/08  15:38:42      T03496
```

Dump Viewing Selection

This panel is the gateway to the Dump Viewing functions. On the entry line of the panel, you may enter the selection number of the function you wish to perform.



```
BLNDVS01                 DUMP VIEWING SELECTION
===> _

DUMP NAME .... SYSDUMP.F4.P6093403

To select a function, type its number and press ENTER:

1  DUMP DISPLAY OPTIONS   Examine or change dump display options

2  DUMP DISPLAY           Examine dump data

3  ANALYSIS SUMMARY       Examine text, control block locations,
                          control block linkages, etc.

4  ANALYSIS ROUTINES      Select an analysis routine for execution
```

MAINTAIN SYSTEM HISTORY PROGRAM (MSHP)

MSHP, IBM's installation and service tool for VSE Systems, is a part of VSE/Advanced Functions. Any installation or service job using the program is to be run in a partition of **640K bytes** or larger; this includes MSHP's partition GETVIS requirement of up to 92K bytes.

MSHP invokes, and supplies input to, various other system programs to perform its installation and service functions. These programs - such as the linkage editor and the librarian - run in the same partition as MSHP. They return control to MSHP when their function is complete.

MSHP records installation and service activities done under its control in a file on disk. This file, the 'System History File', is used by MSHP for maintaining system integrity. It is therefore essential that the file reflects the system's current change level at all time.

During installation and service of program products, certain functions of MSHP use an auxiliary history file, normally as a work file.

The System History File

The file is maintained under the file name IJSYSHF. To access the file, MSHP uses the IBM set default logical unit SYSREC; in other words, the same volume that contains the system's recorder file. However you can use any programmer logical unit to refer to the file if you place it on a volume other than that of SYSREC.

The file should be permanently defined by a permanent entry in the system standard label area:

```
  // DLBL IJSYSHF,'A5666301.SYSTEM.HISTORY.FILE',99/365
  // EXTENT SYSREC,vol-id,1,0,number1,number2
```

You may also define the file by using the MSHP DEFINE statement. If you use this approach and request several MSHP functions within one run, your DEFINE statement must follow the first MSHP function request. This definition is valid only until the end of this MSHP run.

The Auxiliary History File

MSHP needs this file for certain functions during the installation of a program product or of service changes.

The file, which MSHP uses primarily as a work file, is maintained on disk under the file name IJSYS02. Normally, it is created on the volume assigned to the logical unit SYS002, but you can use any other programmer logical unit if this is desirable. If you use a different programmer logical unit, this logical unit must be assigned before you submit the EXEC statement invoking MSHP.

MSHP Control Statements

MSHP has two types of control statements: functional control and detail control.

- Functional control statements define the desired MSHP functions.
- Detail control statements provide descriptive data about the requested function. Detail control statements, if required for a function, must always follow immediately the applicable function control statement.

RULES FOR WRITING MSHP CONTROL STATEMENTS

- With one EXEC MSHP, any number of function control statements can be specified.
- The function control statement you use determines which detail control statement(s) must or may follow.
  If statements must be submitted in a specific sequence, this is noted in the description for the statement(s).
- Detail control statements can follow only a function control statement or another detail control statement.
- MSHP control statements are of free form. The statement's operation code may begin in any input character position.
- An input line for MSHP control statements represents the first 72 characters of a card image input record or 120 characters for console input.
- Operation codes and operand keywords may be abbreviated. In the statement descriptions, permissible minimum abbreviations are shown as uppercase character strings, followed by the remainder of the keyword in lowercase. For example, INSTall may be coded as INST, INSTALL, or anything within these limits.
- A value contained within brackets [....] may be included or omitted, depending on the requirements of the program.
  Two or more values contained within brackets and separated by an | sign represent alternatives, one (and only one) of which may be chosen.
  For Example:

  ```
  [IRRevokable|REVokable]
  ```

  If you specify nothing, MSHP assumes the underlined option as default.
- Options contained within braces ....} and separated by an | sign represent alternatives, one of which must be chosen.
  For Example:

  ```
  PRODuct|SYSres}
  ```

- The operands of a statement are separated from one another by:
  - One or more blanks
  - A comment (which is text within /* and */)
  - A comma (which may be surrounded by one or more blanks or comments)
- An all-comment input line is allowed; however, not beginning in column 1.
- Words given in all lowercase letters represent information that must be supplied by the user.
- The equal sign (=), the plus sign (+), the colon (:), and the single quotes (' ') must be coded as shown; they may be surrounded by one or more blanks, except for the (+) sign, which must not be preceded or followed by a blank.
- An ellipsis (a series of three periods) indicates that a list of up to 100 items (such as PTF numbers) may be specified within parentheses. For Example:

  ```
  (UD27484,UD13528,...)
  ```

  However, a single item does not have to be enclosed in parentheses.
- The individual values in a list can be separated from each other by:
  - One or more blanks
  - A comment (text within /* and */)
  - A comma (which may be surrounded by one or more blanks or comments)

  Commas and blanks as separators may be intermixed in a given list.
- A control statement (function or detail) ends with the end of the input line, unless it is explicitly continued by means of a dash (-), followed by at least one blank. It may also end with a semicolon.

The continuation dash must also be preceded by at least one blank, except after a

- Comma
- Parenthesis
- Equal sign
- Comment end
- Quoted string end

For function control statements, not more than six continuation lines may be specified.

A pair of values connected by a colon, as in

```
APPLY 5666-301-01:UD12345
```

cannot be broken by a line end; nor can a keyword itself, a number, or a quoted or unquoted string be continued on a subsequent line.

- More than one control statement may be coded on the same line. In this case, preceding statements on the line have to be terminated by a semicolon (;). For Example:

  ```
  PTF UD00001; PTF UD00002
  ```

- From the console, MSHP control statements may be entered in uppercase or in lowercase.
- An MSHP statement entered from the console may be canceled by entering two question marks (??).

SYSIN Service Tape Format

From VSE/SP 2.1 (VSE/AF 2.1) up, the format of the distribution media for preventive and corrective PTFs will change to a common SYSIN format. This 'SYSIN' service tape format replaces the old 'PUT Tape' format.

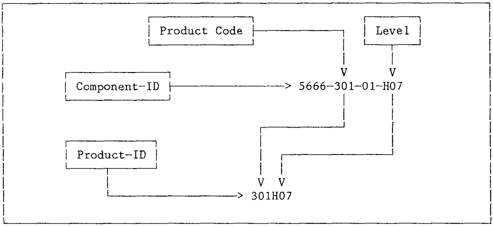
The format of the service tape is as follows:

- FILE 1 : History file or null file
- FILE 2 : Preventive service document
- FILE 3 : Not used by MSHP
- FILE 4 : EXCLUDE-List or null file
- FILE 5 : Coverletters or null file
- FILE 6 : PTFs
- FILE 7 : Not used by MSHP
- FILE 8 : Not used by MSHP

The files which are not used by MSHP are present for compatibility reasons.

The 'old' format PTF Tapes are not longer valid, because when PTFs will be installed, MSHP skips automatically to FILE 6.

Component Identification



Definitions

component: The term 'component' stands for the component identification number of a component, for example:

5666-301-01

A component may occur in more than one product, in which case it is further qualified by a 'level' indication.

Note: When installing PTFs, the level is required too.

level: Denotes the level of a component. It is a string of three alphameric characters which identifies the component uniquely if, for example, the component is shared by several different products.

To indicate that a component belongs to a certain product, the level number of that product is appended (with a -) to the component's name.

For example:

5666-301-01-H07

identifies component 5666-301-01 at level H07.

product: The term 'product' stands for the 6-character product identification number of a product, for example:

301H07

where the first three characters are derived from the product's program number (for example, 301 for VSE/Advanced Functions), and the remaining characters are the level number of the product, formerly known as feature number or release number.

This level number is also the level number of any component(s) belonging to the product.

Restore System History

```
// JOB RESTORE SYSTEM HISTORY
// ASSGN SYS006,cuu
// MTC REW,SYS006
// MTC FSF,SYS006,2
// EXEC MSHP
RESTORE HIST SYS
/*
/&
```

Note:
SYS006 (IJSYS06) - Tape drive on which the distribution tape is mounted.
MTC commands are used to position the distribution tape to the beginning of the system history file.

Personalize the System History

```
// JOB PERSONAL SYSTEM HISTORY
// EXEC MSHP
PERSONALIZE 'customer name' -
     ADDR='location' -
     PHONE='extension' -
     PROGR='programmer name' -
     ENV='environment'
/*
/&
```

Note:
'customer name' may be up to 20 characters.
'location' is the address not longer than 45 characters.
'extension' is the telephone number up to 17 characters.
'programmer name' may be up to 24 characters.
'environment' is system related information. e.g. The software with the component level. (up to 62 characters.)

Install SYSRES from Tape (Install New System)

```
// JOB INSTALL NEW SYSTEM
// ASSGN SYS006,cuu
// ASSGN SYS007,cuu
// ASSGN SYS008,cuu
// DLBL IJSYSR1,'A5666301.PRODUCTION.LIBRARY',99/365
// EXTENT ,SYSRES,1,0,number1,number2
// DLBL GENLIB,'A5666301.GENERATION.LIBRARY'
// EXTENT ,WRKFL1,,,0,1700
// EXEC LIBR
DEFINE LIB=IJSYSR1
DEFINE LIB=GENLIB
/*
// EXEC MSHP
INSTALL SYSRES FROMTAPE ID=301H07 -
               PRODUCTION INTO=IJSYSR1 -
               GENERATION INTO=GENLIB
/*
/&
```

Note:
SYS006 (IJSYS06) - Tape drive on which the distribution tape is mounted.
SYS007 (IJSYS07) - New SYSRES (Production Library) device
SYS008 (IJSYS08) - New GENERATION library device

Tailor a Supervisor (with Generation Feature)

```
// JOB $$A$SUP1
// ASSGN SYS001,cuu
// ASSGN SYS002,cuu
// ASSGN SYS003,cuu
// ASSGN SYS004,cuu
// ASSGN SYSLNK,cuu
// ASSGN SYSPCH,IGN
// OPTION CATAL
// EXEC MSHP
TAILOR 5666-301-06-H07 PHASE=$$A$SUP1 KEEPDATA
RESOLVES 'SUPERVISOR 1 GENERATION'
EXECUTE (ASSEMBLY LNKEDT) XREF
          TITLE 'VSE SUPERVISOR SOURCE CODE FOR $$A$SUP1'
          SUPVR ID=1,
                MICR=1419,
                ... ... ...
          FOPT  DASDSHR=YES,
                TTIME=BG,
                ... ... ...
          IOTAB IODEV=254,
                NPGR=600
          END
/$
/*
/&
```

KEEPDATA, an optional specification, causes the source code to be stored in the system's history file. MSHP uses that code if retailoring is to be done later on.

Note:
SYS001 (IJSYS01) - used for Assembler and Linkage Editor
SYS002 (IJSYS02) - used for Assembler
SYS003 (IJSYS03) - used for Assembler
SYS004 (IJSYS04) - used for MSHP
SYSLNK (IJSYSLN) - used for Linkage Editor

Retailor a Supervisor (with Generation Feature)

```
// JOB $$A$SUP1
// ASSGN SYS001,cuu
// ASSGN SYS002,cuu
// ASSGN SYS003,cuu
// ASSGN SYS004,cuu
// ASSGN SYSLNK,cuu
// ASSGN SYSPCH,IGN
// OPTION CATAL
// EXEC MSHP
TAILOR 5666-301-06-H07 PHASE=$$A$SUP1 KEEPDATA
/$
/*
/&
```

The options for the generation macros will be taken from the system's history file (from a previous tailor job).

Note:
SYS001 (IJSYS01) - used for Assembler and Linkage Editor
SYS002 (IJSYS02) - used for Assembler
SYS003 (IJSYS03) - used for Assembler
SYS004 (IJSYS04) - used for MSHP
SYSLNK (IJSYSLN) - used for Linkage Editor

Install a Program Product

```
// JOB INSTALL PROGRAM PRODUCT
// ASSGN SYS006,cuu
// OPTION STDLABEL
// DLBL PRPNLIB,'PROGRAM.LIBRARY.PRODUCT'
// EXTENT ,PRPLB1,,,1,1899
// DLBL PRGNLIB,'PROGRAM.LIBRARY.GENER'
// EXTENT ,PRPLB2,,,1,1899
// OPTION USRLABEL
// LIBR
DEFINE LIB=PRPNLIB
DEFINE LIB=PRGNLIB
/*
// EXEC MSHP
INSTALL PRODUCT FROMTAPE ID=tapefile-id -
                PRODUCTION INTO=PRPNLIB.SLIB1 -
                GENERATION INTO=PRGNLIB.SLIB2
RETRACE PRODUCT
/*
/&
```

Note:
SYS006 (IJSYS06) - Tape drive on which the distribution tape is mounted.

Backup a Program Product

```
// JOB BACKUP PROGRAM PRODUCT
// ASSGN SYS006,cuu
// EXEC MSHP
BACKUP PRODUCT=product-id -
       ID=tape-id -
       PRODUCTION
/*
/&
```

Note:
SYS006 (IJSYS06) - Tape drive on which the output tape is mounted.

List Service Information

```
// JOB LIST SERVICE INFORMATION
// ASSGN SYS006.cuu
// EXEC MSHP
LIST SERVICE TAPE -
     DOCUMENT COVER SEPARATE
(PTF=UDnnnnn,UDnnnnn,...) - optional if only specified PTFs should
                            be printed. If this statement is omitted,
                            all PTFs will be printed.
/*
/&
```

Note:
SYS006 (IJSYS06) - Tape drive on which the service tape is mounted.

Install Service

```
// JOB INSTALL SERVICE
// ASSGN SYS002,cuu
// ASSGN SYS006,cuu
// EXEC MSHP
INSTALL SERVICE RESTART -
        TAPES=2
INCLUDE PTF=(UDnnnnn,UDnnnnn,...)
DEFINE HISTORY AUX EXT=number1:number2 UNIT=SYS002
/*
/&
```

Note:
SYS002 (IJSYS02) - Disk drive on which the aux history is located.
SYS006 (IJSYS06) - Tape drive on which the service tape is mounted.

Restart PTF Installation

The installation of PTFs frequently requires modules to be link-edited into phases. Before this link-editing under MSHP control starts, MSHP takes a checkpoint. Should this link-editing fail, then MSHP terminates PTF installation, but allows you to set up the install job again at the recorded checkpoint. To restart the installation process at this checkpoint, submit a job similar to the one shown below:

```
// JOB INSTALL SERVICE
// EXEC MSHP
INSTALL SERVICE RESTART
/*
/&
```

Note:
For a restart, MSHP needs no input other than the INSTALL statement as shown in the above sample job.

Install a Backout PTF

```
// JOB INSTALL BACKOUT PTF
// ASSGN SYS006,cuu
// EXEC MSHP
INSTALL BACKOUT
INCLUDE PTF=UD12345
/*

- - - - - - - MSHP  generated - - - - - - -
:  // OPTION CATAL                         :
:  // EXEC MSHP                            :
:  REVOKE 5666-SCI-01-H07 : UD12345        :
:  DATA                                    :
:    ... ... ...                           :
:  END                                     :
:  /$                                      :
:  /*                                      :
:  /&                                      :
- - - - - - - - - - - - - - - - - - - - - -
```

Note:
SYS006 (IJSYS06) - Tape drive on which the backout tape is mounted.

Correct a Phase

```
// JOB CORRECT PHASE
// OPTION CATAL
// EXEC MSHP
CORRECT 5666-273-01-A45 : DY21001
AFFECTS PHASES=IPW$$OT EXPAND=100
ALTER F0 9200B0F8:92F180F8
ALTER 6FA 00000000:4700C426
RESOLVES 'ERROR ON TAPE OPEN'
/*
/&
```

Remove the Fix

```
// JOB UNDO FIX
// OPTION CATAL
// EXEC MSHP
UNDO 5666-273-01-A45 : DY21001
/*
/&
```

Correct a Module

```
// JOB CORRECT RELOCATABLE MODULE
// OPTION CATAL
// EXEC MSHP
CORRECT 5666-301-02-H07 : DY19227
AFFECTS MODULE=IJWCCD2 ESDID=1
ALTER 1048 47F0F000:47F0F800
RESOLVES 'CLEAR DISK ERROR'
INVOLVES LINK = IJWCCD
/*
/&
```

Correct a Source Macro

```
// JOB CORRECT SOURCE MACRO
// OPTION CATAL
// EXEC MSHP
CORRECT 5666-301-02-H07 : DY17291
AFFECTS MACROS=CDLOAD
VERIFY 007100
          AIF    (K'&PHASE LE 8).FOUR
INSERT 7100
          AGO    STOP
/$
INSERT 9100
          STOP   ANOP
/$
CORRECT 5666-301-02-H07 : DY18456
AFFECTS MACROS=SETL
REPLACE : 300000+21
          RETURN ANOP
/$
AFFECTS MACROS=SECHECK
DELETE 071000 : 072000
/*
/&
```

Correct Unedited Source Macro

```
// JOB CORRECT UNEDITED SOURCE MACRO
// OPTION CATAL
// EXEC MSHP
CORRECT 5746-XX-100-H57 : PP73336
AFFECTS MACROS=DLZCKOPT TYPE=A
DELETE : 000400
INSERT 450
          LCLB@B(9),@NGP
@B(9)    SETB  (@PIO(@P))
/$
/*
/&
```

Archive Update

```
// JOB ARCHIVE UPDATE
// EXEC MSHP
ARCHIVE 5666-301-08-H07 APAR=LF00003
/*
/&
```

MSHP FUNCTION CONTROL STATEMENTS SUMMARY

Function Control Statement*	Purpose
APply	Install a PTF and record it in the system history file.
ARChive	Enter, into the history file, information relating to products, components, PTFs, and local or APAR fixes.
BACKup	Copy an auxiliary or system history file from disk to magnetic tape. Produce a backup copy of a product.
COPy	Copy a history file from disk to disk.
CORrect	Install a local or APAR fix.
CReate	Pre-format a history file and reserve space for the PERsonalize function (see below).
DUMP	Produce a formatted printout of a system or auxiliary history file.
INCorporate	Install a component distributed in SYSIN format.
INSTall SYsres/ PRoduct	Install a system (SYsres) or product (PRoduct).
INSTall SErvice/ BAckout	Apply preventive and corrective service from the service tape (SErvice) or a backout tape (BAckout).
LIST	Write information from a service tape onto SYSLST.
Lookup	Display, on SYSLOG, selected information from the system history file.
MERge	Put entries of a history file into another history file.
PERsonalize	Identify a system history file in relation to a specific user.
REMove	Erase entries from the system history file.
RESIdence	Specify names of sublibraries in which a product resides
RESTore	Restore a complete shipment package or a history file from magnetic tape to disk.
RETRace	Print information from the system history file on SYSLST
REVoke	Restore an operational system to the status existing prior to a PTF installation.
SELect	Select individual tailor jobs from the generation file (for retailoring).
TAILor	Identify and initiate the generation (or regeneration) of a library member.
UNdo	Remove an initiated local or APAR fix to re-establish the previous library status.

*Note: The capitalized part of the statement is the minimum allowed abbreviation.

MSHP DETAIL CONTROL STATEMENTS SUMMARY

Detail Control Statement*	Application
AFFects	Library members that are affected by a PTF or local fix.
ALter	Specify text modifications for library members.
COMPAtible	Products that are compatible with the shipped product.
COMPRises	Identify the component, phases, modules, and/or macros that comprise a product, and enter the information in the history file.
DATA	Delimit input to the LIBR and LNKEDT programs.
DEFine	Create label/extent definitions for the history file.
DELete	Delete lines from source book when applying a local fix.
EXCLude	Exclude one or more products, components, or PTFs from a service application.
EXECute	Call one or more system programs (for example, assembler) required for tailoring.
GENerate	Specify a phase, module, or macro for regeneration.
INCLude	Include one or more products, components, or PTFs in a service application.
INFluences	Identify those generated phases, modules, or macros that are affected by a PTF or local/APAR fix and that have to be regenerated if the fix is applied.
INsert	Insert lines in a source book when applying a local fix.
INVolves	Explicitly request link-editing when installing a product or applying service.
OR	Delimit a set of requirements (initiated with the REQuires statement) and test the requirements.
PTF	List the PTFs whose cover letters are to be printed.
REPlace	Delimit begin and end of replacement lines for local or APAR fixes; initiate the replacement of the source text.
REQuires	Specify the requirements for successfully installing a shipment package or applying service.
RESolves	Associate a comment with a PTF, a product, an APAR or a local fix, or a generated member.
RESTart	Indicates, for macro updates, that a new sequence number series starts after the specified statement.
SCan	Scan a phase for a specified string, or display 16 bytes of a phase.
SUPersedes	Record the PTFs that are superseded by a given PTF.
VERify	Specify where a verification is to be made for a local or APAR fix correction.

*Note: The capitalized part of the statement is the minimum allowed abbreviation.

Operation	Operands
APply	component[-level] : ptf–number [IRRevokable|REVokable] [INDirect]

The APPLY statement is used to install a single PTF to your system and to record the installation in the system history file.

Logical Unit Assignments

Required:

SYS001 Work file used by MSHP.
SYSLNK Linkage editor input file; needed to catalog phases supplied by IBM in object format.
SYSLST System printer.

Optional:

SYSPCH Needed if a backout PTF is to be generated (via the REVOKABLE operand).
SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

AFFECTS, DATA, RESOLVES APARS=apar-number

Optional:

DEFINE, INFLUENCES, INVOLVES, OR,
REQUIRES, SUPERSEDES

Description of Operands

component[-level]:
Specifies the component to which the PTF is to be installed.

Level specifies the level number (formerly release number) of the applicable component.

ptf-number:
Specifies the number of the PTF to be installed.

IRRevokable:
When installing the PTF, MSHP will not produce any backout PTF jobs. The PTF cannot be revoked, that is, the status before the installation of the PTF cannot be recreated at a later point in time.

REVokable:
When installing the PTF, a backout PTF is to be generated on SYSPCH.

Restrictions: Do not specify REVOKABLE for a PTF that is a pre- or co-requisite for other PTFs or has comparable local/APAR fix dependencies.
Do not specify REVOKABLE if the PTF contains new or additional modules or macros that are not part of the current component release.

INDirect:
Specifies a PTF for indirect application via the Service Dialog of VSE/SP.

Note: For compatibility reasons, the APPLY statement is still accepted in the 'old' format.

Operation	Operands
APply	component[(level)] : ptf-number [RELease=(release-number,...)]

(level):
Indicates the old three-character alphameric identifier of the component.

RELease=(release-number,...):
Specifies the release(s) of the component to which the PTF is to be installed. This operand applies to old-format statements only and is ignored if level was specified. If level was not specified, MSHP converts the release number into a level number.

<table>
<tr><th>Operation</th><th>Operands</th></tr>
<tr><td>ARChive</td><td>product</td></tr>
<tr><td>ARChive</td><td>component-level [PTF=ptf-number|APAR=apar-number
|SOFTreject]</td></tr>
</table>

The ARChive statement is used to make entries in the system history file.

Logical Unit Assignments

Required:

SYSLST System printer.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

ARCHIVE...	Prod.	Comp.	PTF	APAR
AFFECTS			R	R
ALTER				O
COMPRISES	R			
DEFINE	O	O	O	O
DELETE				O
INSERT				O
INVOLVES	O	O	O	
OR	O	O	O	
REPLACE				O
REQUIRES	O	O	O	
RESOLVES	R		R	O
SUPERSEDES			O	
R = Required	O = Optional			

Description of Operands

product:
Specifies that an entry for the named product is to be made in the system history file.

component-level:
Specifies that an entry for the named component is to be made in the system history file.

If a PTF or APAR is specified, 'component' identifies the component to which the particular PTF, local fix, or APAR fix to be archived applies.

PTF=ptf-number:
Identifies the PTF for which an entry is to be made in the history file.

APAR=apar-number:
Identifies the local or APAR fix for which an entry is to be made in the history file.

SOFTreject:
Specifies that a PTF which may have to be installed to the named component is to be installed even if, as a result, a local or APAR fix would be partially overwritten. (The same applies to a PTF that may have to be revoked.) For a component that is archived without SOFTREJECT specified, MSHP automatically rejects the installation (revocation) of a PTF that partially overwrites a local or APAR fix.

Operation	Operands
BACKup	History [SYStem¦AUXiliary]

The BACKUP HISTORY statement requests MSHP to copy a history file located on disk onto magnetic tape.

Logical Unit Assignments

Required:

SYS006 The tape onto which MSHP writes the backup copy of the history file.
SYSLST System printer.

Required for BACKUP HISTORY AUXILIARY:

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required for BACKUP HISTORY SYSTEM if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required: none

Optional: DEFINE

Description of Operands

History SYStem:
Specifies that the system history file is to be copied to tape.

History AUXiliary:
Specifies that the auxiliary history file is to be copied to tape.

Operation	Operands
BACKup	PRoduct=(product,...) ID='tapefile-id' [HEADer=member-name] [PRODuction] [GENEration]

The BACKUP PRODUCT statement is used to produce, on magnetic tape, a backup copy of the named product(s). This backup copy consists of the production and generation sublibraries of the product(s), together with the pertinent history file containing product-related entries.

Logical Unit Assignments

Required:

SYS006	The tape onto which MSHP writes the backup copy of the named product(s).
SYSLST	System printer.

Related Detail Control Statements

Required: none

Optional: DEFINE HISTORY SYSTEM

Description of Operands

PRoduct=(product,...):
Specifies the product(s) for which a backup copy is to be produced.
All requested products must reside in the same set of production and generation sublibraries, since MSHP copies only entire sublibraries.

ID='tapefile-id':
Specifies that an identification is to be given to the created backup file, which MSHP can later use during RESTORE to locate the backup file.

HEADer=member-name:
Specifies an additional library member that is to be written as a header file onto the backup tape (as the very first file created with this BACKUP statement).

PRODuction:
Specifies that only the production sublibrary of the named product(s) is to be copied.

GENEration:
Specifies that only the generation sublibrary of the named product(s) is to be copied.

Operation	Operands
COPy	History {SYStem AUXiliary|AUXiliary SYStem}

The COPY statement requests MSHP to copy a history file from disk to disk:

- The sequence 'SYSTEM AUXILIARY' creates a copy of the system history file for use as an auxiliary history file.
- The sequence 'AUXILIARY SYSTEM' copies an auxiliary history file to the system history file.

Logical Unit Assignments

Required:

SYSLST System printer.
SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required: none

Optional: DEFINE HISTORY

Description of Operands

History SYStem AUXiliary:
Creates a copy of the system history file for use as an auxiliary history file.

History AUXiliary SYStem:
Copies an auxiliary history file to the system history file.

Restriction

If the history file extent resides on a newly defined VM minidisk, this extent should be initialized by:

- CMS command FORMAT, followed by
- VM disk initialization program IBCDASDI, or
- Device Support Facilities INIT command with parameter 'Mimic(Mini(n))'.

Alternatively, you can use the MSHP RESTORE statement to restore a history file from tape to the minidisk extent.

Operation	Operands
CORrect	component[-level] : apar-number [REVokable|IRRevokable]

The CORRECT statement specifies that a local or APAR fix is to be installed to a component.

Note: To avoid an unintended removal of a fix due to linkage editor or assembly runs after the application of the fix, a correction should always be made in all applicable library members (phases, modules, macros).

Logical Unit Assignments

Required:

SYSLST System printer.

Optional:

SYSLNK Linkage editor input file; needed if the correction requires link-editing.
SYSPCH Needed when correcting a macro.

SYS001,
SYS004 Needed as work files if the correction involves
a) modules or expandable phases
b) macros.

SYS002,
SYS003 Needed as work files if corrections to macros are involved.

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

AFFECTS

Optional:

ALTER, DEFINE HISTORY SYSTEM, DELETE,
INFLUENCES, INSERT, INVOLVES, REPLACE,
REQUIRES, OR, RESOLVES 'comment',
RESTART, SCAN, VERIFY

The detail control statements must be entered in the following sequence:

1. DEFINE HISTORY SYSTEM
2. REQUIRES, OR
3. RESOLVES
4. AFFECTS
5. ALTER, DELETE, INSERT, REPLACE, RESTART, SCAN, and VERIFY, if used, must be coded after the AFFECTS statement.
6. INFLUENCES, INVOLVES.

Description of Operands

component[-level]:
Specifies the component that is to be corrected by the local or APAR fix.

If only one level is installed, MSHP applies the fix to this one; otherwise, MSHP informs you which levels are installed and asks you for the requested one.

apar-number:
Specifies the number of the local or APAR fix that contains the correction(s).

REVokable:
Specifies that corrections made to phases or modules can be removed by using the UNDO function.

For corrections to macros, the REVOKABLE option causes a job to be created on SYSPCH with the UNDO control statement.

IRRevokable:
Specifies that corrections cannot be revoked.

Note: For compatibility reasons, the CORRECT statement is still accepted in the 'old' format.

Operation	Operands
CORrect	component[(level)] : apar-number ...

(level):
Indicates the old three-character alphameric feature identifier of the component.

Operation	Operands
CReate	History [SYStem|AUXiliary]

The CREATE statement requests MSHP to preformat a history file. (For defining a history file, that is, creating extent definitions, refer to the DEFINE HISTORY detail control statement.)

Logical Unit Assignments

Required:

SYSLST System printer.

Required for CREATE HISTORY AUXILIARY:

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required for CREATE HISTORY SYSTEM if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required: none

Optional: DEFINE

Description of Operands

History SYStem:
Specifies that a system history file is to be created.

History AUXiliary:
Specifies that an auxiliary history file is to be created.

Restriction

If the historv file extent resides on a newlv defined VM minidisk. this extent should be initialized by:

- CMS command FORMAT, followed by
- VM disk initialization program IBCDASDI, or
- Device Support Facilities INIT command with parameter 'Mimic(Mini(n))'.

Alternatively, you can use the MSHP RESTORE statement to restore a history file from tape to the minidisk extent.

Operation	Operands
DUMP	History [SYStem|AUXiliary]

The DUMP statement requests MSHP to produce a formatted hexadecimal printout of a historv file on SYSLST.

Logical Unit Assignments

Required:

SYSLST System printer.

Required for DUMP HISTORY AUXILIARY:

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required for DUMP HISTORY SYSTEM if the device on which the system historv file resides is soecified in the UNIT ooerand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required: none

Optional: DEFINE

Description of Operands

History SYStem:
Specifies that the system history file is to be dumped.

History AUXiliary:
Specifies that the auxiliary history file is to be dumped.

Operation	Operands
INCorporate	component[-level] [RELease=release-number]

The INCORPORATE statement is used to install a component distributed in SYSIN format.

Loaical Unit Assianments

Required:

SYSLNK Linkage editor input file.
SYS001 Linkage editor work file.
SYSLST System printer.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required: DATA

Optional: DEFINE, INVOLVES, OR, REQUIRES.

Description of Operands

component[-level]:
Identifies the component to be incorporated.

RELease=release-number:
Specifies the release of the component to be incorporated (only applicable if 'level' is not specified in the component operand). MSHP converts the release number into a level number.

Operation	Operands

INSTall	{PRoduct|SYsres}
	[FROMTape[ID='tapefile-id']|FROMDisk] {INto=lib|
	[PRODuction INto=lib[.sublib]]
	[GENEration INto=(lib[.sublib],...)]}

The INSTALL statement requests MSHP to install either a product, such as VSE/VSAM, or a SYSRES package, such as VSE/Advanced Functions.

The history file that accompanies the programming support reflects the contents of the shipment package; it may contain information on any requirements that have to be met prior to installation (for example, pre-requisite components or PTFs). When executing the INSTALL function, MSHP informs you of any missing requirements.

MSHP also determines (by analyzing the shipment history file and your system's history file) which of the products already installed in your system are compatible with the shipped product and which are superseded.

Logical Unit Assignments

Required:

SYS006 Distribution tape.
SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required if the device on which the system history resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required: none

Optional: DEFINE HISTORY, COMPATIBLE

Description of Operands

Product:
Specifies that a product (non-SYSRES package) is to be installed.

SYsres:
Specifies that a SYSRES package is to be installed.

FROMTape:
Specifies the distribution tape.

MSHP restores the history file from the distribution tape into an auxiliary history file. You may either use the standard SYS002 work file for the auxiliary history file or define it with a DEFINE HISTORY AUXILIARY statement.

With the restored auxiliary history file, checking for pre-, co-, and negative-requisites is performed. If all checks and verifications prove satisfactory, the distribution libraries are restored into the specified target libraries, and the restored distribution history file is merged with the current system history file.

ID='tapefile-id':
Indicates that the distribution tape is to be searched for the denoted tapefile-id, which was specified with the BACKUP statement. If the tape is not correctly positioned, it will be scanned (forward only) for the specified id and correctly positioned before installation.

FROMDisk:
FROMDISK must be specified to support the INSTALL function for a system without magnetic tape.

Defaults:

- For INSTALL SYSRES, you must always specify the names of the target libraries (lib) and provide the necessary label information (DLBL/EXTENT). If the libraries do not exist, MSHP creates them. If you do not specify a sublibrary (sublib) name, MSHP takes the name of the shipment sublibrary, which is SYSLIB.
- For INSTALL PRODUCT, you need not specify the names of the target libraries and sublibraries, if these libraries exist (online) and label information is available in the label area.

INto=lib:
Specifies, for INSTALL PRODUCT only, installation of both the production and the generation part of the shipment package into the library denoted by 'lib'. (For INSTALL SYSRES, the generation part must be installed into a different target library than the production part.)

PRODuction INto=lib[.sublib]:
Specifies installation of the executable (production) part of the shipment package, which consists of all phases, procedures (and some modules/macros) needed for daily operation of your system or product. The production part must be installed before the generation part.

GENEration INto=(lib[.sublib],...):
Specifies installation of the generation part of the shipment package, which contains those modules and, possibly, macros that are needed for the regeneration of the product.

Operation	Operands
INSTall	{SErvice[REVokable|REStart] [TAPes=no.-of-tapes] [SD] |BAckout[REStart]}

The INSTALL SERVICE statement requests MSHP to install PTFs from one or more service tapes.

The INSTALL BACKOUT statement requests MSHP to install one or more backout PTFs, which amounts to recataloging the library member(s) replaced by installing the corresponding PTF(s). The statement works in the same way as the INSTALL SERVICE statement, except that it reads the PTF information from the backout tape, which is created when you specify the REVOKABLE operand in the INSTALL SERVICE statement.

Via the INCLUDE and EXCLUDE detail control statements you can specify that only certain products, components, or PTFs are to be included or excluded during the service application. If you omit the EXCLUDE and INCLUDE statements, all service tape PTFs which are applicable to your system will be selected for service installation.

MSHP prints a list of all PTFs that are to be installed and asks you for confirmation before it replaces the affected library members in your sublibraries and updates the history file.

Logical Unit Assignments

Required:

SYS006	Service/backout tape.
SYS002	Work file used by MSHP.
SYSLST	System printer.

Optional:

SYSxxx	Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.
SYS004	Needed if backout PTF jobs are to be created (by specifying the REVOKABLE operand).

Related Detail Control Statements

Required:	none	
Optional:	INCLUDE,	EXCLUDE (for INSTALL SERVICE onlv)

Description of Operands

SErvice:
Specifies that MSHP is to install PTFs from the service tape(s), as detailed in any INCLUDE or EXCLUDE detail control statements.

REVokable:
Specifies that backout jobs are to be created for all PTFs that are to be installed. The backout jobs are MSHP jobs with the REVOKE function control statement included. The backout jobs are written in blocked format onto a tape which must be assigned as SYS004.

BAckout:
Specifies that MSHP is to install one or more backout PTFs from the backout tape which is produced by the INSTALL SERVICE function with the REVOKABLE option. The PTFs to be installed can be selected via the INCLUDE statement.

REStart:
Requests MSHP to restart a previous INSTALL SERVICE/BACKOUT or APPLY/REVOKE job whose final link step failed. MSHP scans the history file entries for those PTFs that were correctly cataloged, but not yet linked, and invokes the linkage editor to complete the final link step.

TAPes=no.-of-tapes:
Indicates to MSHP the number of tape volumes that have to be scanned for the particular service installation. If you know that prerequisite PTFs exist on other service tapes and that these PTFs are not yet installed, have MSHP scan these additional tape volumes for the prerequisite PTFs and have them retrieved for installation.
The additional tapes (maximum number=9) are to be mounted, on the tape drive assigned to SYS006, in response to MSHP's mount request.
If the operand is omitted, one tape volume is assumed.

SD:
This operand indicates that service is to be applied via the VSE/SP Service Dialog. For those PTFs that are flagged with the INDIRECT option (in the APPLY statement), the library members affected by the service application are first linked to a reserved sublibrary $$MSHPIL before they are finally moved into the system sublibrary IJSYSRn.SYSLIB. This is to protect the IPLed SYSLIB in case the final link step fails.

Operation	Operands
LIST	SERVICEtape [DOCument|NODOCument][XREF|NOXREF] [COVer[CONTinuous|SEParate]|NOCOVer]

The LIST statement requests MSHP to print, on SYSLST, information from a service tape.

Logical Unit Assignments

Required:

SYS006	Service tape.
SYSLST	System printer.

Optional:

SYS002	Needed as work file if COVER is specified.
SYSxxx	Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:	none
Optional:	PTF (if COVER is specified)

Description of Operands

SERVICEtape:
Specifies that information from a service tape is to be printed.

DOCument:
Specifies printing of the service tape documentation, which contains information on how to apply corrective and preventive service from the service tape.

NODOCument:
Suppresses the DOCUMENT function.

XREF:
Specifies printing of the cross-reference list of all PTFs shipped on the service tape.

NOXREF:
Suppresses the XREF option.

COVer:
Specifies printing of the cover letters of those PTFs that are listed on an associated PTF detail control statement. If no PTF statement is given, the cover letters of all PTFs on the service tape are printed.

NOCOVer:
Suppresses the COVER function.

CONTinuous:
Specifies that the cover letters of the PTFs are to be printed without starting a new page for each PTF.

SEParate:
Causes a new page to be started for each PTF cover letter that is to be printed.

Operation	Operands
Lookup	PRoduct=product
Lookup	[component[-level]] [PTF=ptf-number |APAR=apar-number |PHAse=member-name[DATA] |MODule=member-name[DATA] |MACro=member-name [TYpe=member-type][DATA]]

The LOOKUP statement requests MSHP to display, on SYSLOG, selected information from the system history file.

Logical Unit Assignments

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:	none
Optional:	DEFINE HISTORY

Description of Operands

PRoduct=product:
Indicates that the following information is to be displayed for the specified product-id:

- Date of installation.
- Requirements to be met for installation.
- Component(s) contained in the product.
- Comments, if any.

component[-level]:
Specifies the component for which information is to be displayed.

If level is omitted and more than one level of the component is installed, all levels of the component information will be displayed.

PTF=ptf-number:
Indicates that, for the given ptf-number, the history file information is to be displayed.

APAR=apar-number:
Indicates that, for the given apar-number, the history file information is to be displayed.

PHAse=member-name:
Indicates that, for the given phase-name, the history file information is to be displayed.

MODule=member-name:
Indicates that, for the given module-name, the history file information is to be displayed.

MACro=member-name:
Indicates that, for the given macro-name, the history file information is to be displayed.

TYpe=member-type:
Indicates the type of the specified macro. Member-type may be one character only.
If the operand is omitted, type E is assumed.

DATA:
Specifies that the source data from which the phase/module/macro was generated (with TAILOR KEEPDATA) is to be displayed.

Operation	Operands
MERge	History {SYStem AUXiliary|AUXiliary SYStem}

The MERGE statement requests MSHP to insert entries of one history file into another history file.

The sequence of the keywords SYSTEM and AUXILIARY defines the direction of the merge operation. The first keyword specifies the source history file, and the second the target history file. The two keywords must be specified adjacent to each other.

Restriction: Both the source and the target history files must reside on disk.

Logical Unit Assignments

Required:

SYSLST System printer.

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:	none
Optional:	DEFINE

Description of Operands

History SYStem AUXiliary:
Specifies that entries from the system history file are to be merged into an auxiliary history file.

History AUXiliary SYStem:
Specifies that entries from an auxiliary history file are to be merged into the system history file.

Operation	Operands
PERsonalize	['customer-name'] [ADDRess='customer-address'] [PHone='phone-number'] [PRogrammer='programmer-name'] [ENVironment='description']

The PERSONALIZE statement is used to identify a history file and relate it to a specific user.

Restrictions:

- To personalize your system's history file, MSHP needs at least one operand.
- If the history file has not been personalized before, specification of customer-name and customer-address is mandatory.
- The first personalization of a history file changes the dates of all PTF entries to the date when the PERSONALIZE statement is given.

Logical Unit Assignments

Required:

SYSLST System printer.

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:	none
Optional:	DEFINE HISTORY

Description of Operands

'customer-name':
Specifies the user's name that is to be entered in the history file. (1 to 20 characters, enclosed in quotes).

Restriction: If the history file has not been personalized before, customer-name must be specified.

ADDRess='customer-address':
Specifies the address that is to be entered in the history file. (1 to 45 characters, enclosed in quotes.)

Restriction: If the history file has not been personalized before, customer-address must be specified.

PHone='phone-number':
Specifies the phone-number that is to be entered in the history file. (1 to 17 characters, enclosed in quotes.)

A null string ('') is accepted; it erases a previously specified number.

PRogrammer='programmer-name':
Specifies the programmer name that is to be entered in the history file. (1 to 24 characters, enclosed in quotes.)
A null string ('') is accepted; it erases a previously specified name.

ENVironment='description':
Specifies any additional information (for example, the release level) that is to be entered in the history file.
(1 to 62 characters, enclosed in quotes.)
A null string ('') is accepted; it erases a previously specified description.

Operation	Operands
REMove	product
REMove	component-level [PTF=ptf-number |APAR=apar-number |PHAse=member-name |MODule=member-name |MACro=member-name [TYpe=member-type]}

The REMOVE statement requests MSHP to erase entries from the system history file. The space of the removed history file entries is freed for future use.

Note: MSHP does not remove an APAR that was archived as a 'resolved' APAR in conjunction with a PTF.
However, you may still receive messages

```
    M028I REMOVE IN PROGRESS and
    M041I FUNCTION COMPLETED
```

Logical Unit Assignments

Required:

SYSLST System printer.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required: none

Optional: DEFINE HISTORY

Description of Operands

product:
Indicates that the entry for the specified product is to be removed from the history file.

component-level:
Indicates that the entry for the specified component is to be removed (if no further operand follows).
If followed by another operand, 'component' indicates the component to which the specified PTF, APAR, or library member refers.

PTF=ptf-number:
Indicates that the entry for the specified PTF number is to be removed.

APAR=apar-number:
Indicates that the entry for the specified APAR number is to be removed.

PHAse=member-name:
Indicates that the entry for the specified phase name is to be removed.

MODule=member-name:
Indicates that the entry for the specified module name is to be removed.

MACro=member-name:
Indicates that the entry for the specified macro name is to be removed.

TYpe=member-type:
Indicates the type of the macro to be removed. (may be 1 char. only.)
If the operand is omitted, type E is assumed.

Operation	Operands
RESIdence	PRODuct=(product,...) PRODuction=lib.sublib [GENEration=(lib.sublib,...)]

The RESIDENCE statement defines the names of the production and generation sublibraries in which the named products are to reside. This information is recorded in the history file for any follow-on activities, such as service applications, tailoring, installation, or product backup.
Any sublibrary names previously recorded in the history file (via another RESIDENCE or an INSTALL statement) are erased.

Logical Unit Assignments

Required:

SYSLST System printer.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required: none

Optional: DEFINE HISTORY SYSTEM

Description of Operands

PRODuct=(product,...):
Specifies the name(s) of the product(s) whose residence is to be defined.

PRODuction=lib.sublib:
Indicates that the production part of the product(s) is to reside in the named library and sublibrary.

GENEration=(lib.sublib,...):
Indicates that the generation part of the product(s) is to reside in the named library and sublibrary.

<table>
<tr><th>Operation</th><th>Operands</th></tr>
<tr><td>RESTore</td><td>{PRoduct|SYsres} {INto=lib|
PRODuction INto=lib[.sublib]
GENEration INto=(lib[.sublib],...)}
[ID='tapefile-id']</td></tr>
</table>

The RESTORE statement is used to restore a complete shipment tape (production part, generation part, and shipment history file) onto disk; however, without any checks or updates of the system history file.

Logical Unit Assignments

Required:

SYS006 Distribution tape.
SYSLST System printer.
SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required: none

Optional: DEFINE HISTORY AUXILIARY

Description of Operands

PRoduct:
Specifies that a product (non-SYSRES package) is to be restored.

SYsres:
Specifies that a SYSRES shipment package is to be restored.

INto=lib:
Specifies, for RESTORE PRODUCT only, that both the production and the generation part of the shipment package are to be restored into the library denoted by lib.

PRODuction INto=lib[.sublib]:
Specifies that the production part of the shipment package is to be restored to the named library (and sublibrary).

GENEration INto=(lib[.sublib],...):
Specifies that the generation part of the shipment package is to be restored to the named library or libraries.

ID='tapefile-id':
Indicates that the shipment tape is to be searched for the denoted tapefile-id, which was specified in the BACKUP statement. If the tape is not correctly positioned, it is scanned (forward only) for the specified tapefile-id and correctly positioned.
The tapefile-id can be one to 16 characters.

Operation	Operands
RESTore	History [SYStem|AUXiliary]

The RESTORE HISTORY statement requests MSHP to write a history file located on magnetic tape onto disk. For processing, the tape must be positioned to the file containing the history file.

Logical Unit Assignments

Always required:

SYS006 The tape containing the history file.
SYSLST System printer.

Required for RESTORE HISTORY AUXILIARY:

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Optional: DEFINE HISTORY

Description of Operands

History SYStem:
Specifies that the history file is to be copied to the system history file (a disk file with the file name IJSYSHF).

History AUXiliary:
Specifies that the history file is to be copied to the auxiliary history file (a disk file with the file name IJSYS02).

Operation	Operands
RETRace	[PRODucts |COMPonents[IDentifier=component[-level]] |PTFs|APARs|MEMbers]

The RETRACE statement requests MSHP to print information from the system history file on SYSLST.

Defaults: If RETRACE is specified without any keywords, MSHP writes, to SYSLST, an overview report on the system's current service level.

Logical Unit Assignments

Required:

SYSLST System printer.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required: none

Optional: DEFINE HISTORY

Description of Operands

PRODucts:
Requests MSHP to print a list of all installed products.

COMPonents:
Requests MSHP to print a list of all component records from the history file.

COMPonents IDentifier=component[-level]:
Requests MSHP to print information for the specified component only. If level is omitted, MSHP prints information for all installed levels of the component.

PTFs:
Specifies that all applied PTFs are to be listed (in ascending sequence).

APARs:
Specifies that all APARs are to be listed (in ascending sequence) which were corrected by a PTF or local/APAR fix.

MEMbers:
Specifies that all phases, modules, and macros that are affected by a PTF or local fix are to be listed.

Note: Since RETRACE MEMBERS does not indicate whether an APAR, PTF, or component is incorrect or incomplete, use RETRACE APARS|PTFS|COMPONENTS instead.

Operation	Operands
REVoke	component[-level] : ptf-number

The REVOKE statement initiates a backout PTF job that contains the phases, modules, and macros as they were before the named PTF was installed. This backout PTF job (with the initial REVOKE statement) is generated by the APPLY or INSTALL SERVICE statements if REVOKABLE was specified.
A PTF cannot be revoked if it is a prerequisite for another PTF that has not been revoked previously.

Logical Unit Assignments

Required:

SYSLNK	Linkage editor input file.
SYS001	Linkage editor work file.
SYSLST	System printer.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:	DATA
Optional:	none

Description of Operands

component[-level]:
Identifies the component for which the backout PTF was generated.

ptf-number:
Identifies the PTF that is to be revoked.

Operation	Operands
SELect	GENFile COMPonent=component[-level]

The SELECT statement identifies the generation file, from which individual phases, modules, or macros can be regenerated (with the GENERATE detail control statement) after a service application.

Logical Unit Assignments

Required:

SYS005	Generation file.
SYSLNK	Linkage editor input file.
SYS001	Linkage editor work file.
SYSLST	System printer.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:	GENERATE
Optional:	DEFINE HISTORY

Description of Operands

GENFile:
Indicates the generation file.

component[-level]:
Identifies the component to which the members to be regenerated belong. The level indication must be the same as that of the corresponding TAILOR job.

Operation	Operands
TAILor	component[-level] PHAse=member-name |MODule=member-name |MACro=member-name [TYpe=member-type]} [KEEPdata]

The TAILOR statement (together with the EXECUTE detail control statement) is used to generate (tailor) library members of components that are shipped in source-macro format and that have to be assembled and link-edited according to the specific needs of your installation (for example, supervisor macros).

Logical Unit Assignments

Required:

SYSLNK	Linkage editor input file.
SYS001	Linkage editor/assembler work file.
SYS002, SYS003	Assembler work files.
SYSLST	System printer.

Optional:

SYS004	Work file needed by MSHP if MODULE= or MACRO= is specified.
SYSxxx	Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:	EXECUTE	
Optional:	RESOLVES	DEFINE

Description of Operands

component[-level]:
Identifies the component containing the macro, module, or phase to be tailored.

PHAse=member-name:
Specifies the name of the phase to be generated. (See Note, below.) For retailoring, generic names such as DFH* are allowed. **MODule=member-name:**
Specifies the name of the module to be generated. (See Note, below.) For retailoring, generic names such as DFH* are allowed.

MACro=member-name:
Specifies the name of the macro (definition) to be assembled. (See Note, below.) For retailoring, generic names such as DFH* are allowed.

TYpe=member-type:
Indicates the type of the macro to be assembled. Member-type may be one character only.
If the operand is omitted, type E is assumed.

KEEPdata:
Specifies that the source code processed by the invoked control program(s) is to be stored in the system history file.

Note: The operands PHASE=, MODULE=, or MACRO= do not generate any PHASE or CATALOG statements. You have to include these statements after the EXECUTE detail control statement.

Operation	Operands
UNdo	component[-level] : apar-number

The UNDO statement is used to re-establish the status of a library member as it existed before a local or APAR fix was applied with the CORRECT...REVOKABLE statement.

Restriction: If a phase has been expanded by a local or APAR fix, this expansion cannot be removed. The phase remains expanded.

For phases and modules, MSHP may be invoked with an UNDO statement that refers (by component and apar-number) to the correction as specified in the CORRECT statement.

For macros, the UNDO statement is included in the job created (on SYSPCH) by CORRECT...REVOKABLE.

Logical Unit Assignments

Same as for CORRECT.

Related Detail Control Statements

Required:	none	
Optional:	DEFINE HISTORY SYSTEM	DATA

Description of Operands

component[-level]:
Specifies the component from which the local or APAR fix (initiated by CORRECT) is to be removed.

apar-number:
Specifies the local or APAR fix (initiated by CORRECT) that is to be removed.

DETAIL CONTROL STATEMENTS

Operation	Operands

AFFects	[PHAses=(member-name,...) [EXPand=size-increment]}
	[MODules=(member-name,...) [LIOCS][ESDid=esd-number]
	[EXPand=size-increment]}
	[MACros=(member-name,...} [TYpe=member-type]}

The AFFECTS statement identifies the phases, modules, and macros that are affected by a PTF or local fix application.

Restrictions:

- One AFFECTS statement may not refer to more than a total of 100 phases, modules, and macros.
- If AFFECTS is used as a detail control statement to CORRECT, or when archiving a local/APAR fix and the fix information itself, only one phase, module or macro may be specified.

Description of Operands

PHAses=(member-name,...):
Specifies the affected phase(s).

EXPand=size-increment:
Indicates that the specified phase or module (see below) is to be made larger by the number of bytes specified in size-increment (1 to 6 decimal digits), so that fix code can be added at the end of the phase or module.

EXPAND may be specified only when applying a local/APAR fix (with CORRECT) or archiving a local/APAR fix.

MODules=(member-name,...):
Specifies the affected module(s).

LIOCS:
Indicates that a LIOCS module is affected by the PTF.

ESDid=esd-number:
Indicates that a change applies to the specified ESD.

MACros=(member-name,...):
Specifies the affected macro(s).

TYpe=member-type:
Indicates the type of the affected macro.
If the operand is omitted, type E is assumed.

Operation	Operands
ALter	address old-text : new-text

The ALTER statement identifies the modifications that are to be made to a phase or module. This includes verification of the alteration for phases and, optionally, for modules.

Description of Operands

address:
Specifies the (relative) address where the new-text is to begin to replace the old-text. (1 to 6 hexadecimal digits.)

old-text:
Specifies the text that is to be replaced.

MSHP checks the text in the phase or module at the specified address whether it is identical with the old text; replacement by new text takes place only if the text is identical.

Restrictions: Old-text must be specified when modifying a phase; it may be specified when modifying a module.

Old-text can be any of the three formats described in the MSHP USERs GUIDE under the description for the ALter Detail Control Statement.

new-text:
Specifies the text that is to replace the text at the specified address.

For new-text the same rules are valid as for old-text.

If old-text is specified, new-text must have the same length (in bytes).

If new-text is specified without old-text, the colon must be specified at the beginning of the new-text line.

Operation	Operands
COMPAtible	WITH=(product,...)

The COMPATIBLE statement is used to indicate to MSHP at installation time those products that are compatible with the shipped product(s).

Compatible products are usually based on the same base products, contain the same component(s) as the shipped product(s), and may run concurrently with each other. Compatible products may also be stored in the same sublibrary.

Description of Operands

WITH=(product,...):
Specifies the name(s) of the compatible product(s).

Operation	Operands
COMPRises	component [PHAses=(member-name,...)] [MODules=(member-name,...)] [MACros=(member-name,...) [TYpe=member-type]]

The COMPRISES statement is used to specify the component(s) comprised in the shipped product and the library members that make up the component(s). The information is entered in the system history file. A separate COMPRISES statement must be issued for each component contained in the shipped product.

Restriction: One COMPRISES statement may not refer to more than a total of 100 phases, modules, and macros.

Description of Operands

component:
Specifies the component comprised in the shipped product.

PHAses=(member-name,...):
Specifies the phases of the named component.

MODules=(member-name,...):
Specifies the modules of the component.

MACros=(member-name,...):
Specifies the macros of the component.

TYpe=member-type:
Indicates the type of the affected macro.
If the operand is omitted, type E is assumed.

Operation	Operands
DATA	no operands

The DATA statement in conjunction with /$ delimits input that is to be passed by MSHP to the linkage editor or the librarian.

Restrictions:

- A DATA statement (with its corresponding terminating delimiter /$) may be followed only by another DATA statement, not by any other detail control statement.
- The end-of-data indicator (/$) is valid only when input is entered via SYSIPT. Substitute this delimiter by hitting END/ENTER if input is entered from the console.
- Input for the linkage editor must not contain 'named INCLUDE' statements; however, this is not checked by MSHP.

Linking from a link-book (where link-book is an object module that contains LNKEDT control statements) must be requested with the MSHP statement INVOLVES.

MSHP checks the first line after the DATA statement. If this is a linkage editor control statement, all input up to the next /$, is passed to the linkage editor. If it is a librarian control statement, it is passed to the librarian program.

MSHP internally converts any old MAINT CATALR and CATALS statements into the catalog statement.

Operation	Operands
DEFine	History [AUXiliary|SYStem] EXTent=start-track:tracks [SPlit=split-track] [UNIT=SYSxxx] [IDentifier='file-identifier']

The DEFINE statement is used to create extent definitions for a history file in the user label area of the partition in which MSHP is executed.

Restrictions:

- If you use IBM-supplied standard labels or if your own standard label set contains DLBL and EXTENT statements for the system history file (filename IJSYSHF), do not use DEFINE HISTORY SYSTEM in any MSHP job accessing the system history file.
- A DEFINE statement, if used, must immediately follow the applicable function control statement; it may not be placed at the end of several functions or at the end of the job stream.

Description of Operands

History:
Specifies that a history file is to be defined.

AUXiliary:
Specifies that an auxiliary history (work) file is to be defined.

The auxiliary history file is maintained under the filename IJSYS02 on the default logical unit SYS002. MSHP normally uses this file as a history work file.
The DEFINE HISTORY AUXILIARY statement allows you to explicitly define a temporary auxiliary history file (in the user label area) on the logical unit indicated in the UNIT operand.

SYStem:
Specifies that the system history file is to be defined.

The system history file is part of the IBM-distributed system and is maintained under the filename IJSYSHF. To access the file, MSHP uses the IBM-set default logical unit SYSREC. However, you can use any programmer logical unit to refer to the file if you place it on a volume other than that of SYSREC. The history file should be permanently defined. If it is to be on the SYSREC volume, supply the following statements:

```
  // DLBL IJSYSHF,'A5666301.SYSTEM.HISTORY.FILE',99/365
  // EXTENT SYSREC,,1,0,start-address,number-of-tracks/blocks
```

With the DEFINE HISTORY SYSTEM statement you can define a temporary system history file on the logical unit specified in the UNIT operand. This definition is, of course, valid only for the duration of the applicable MSHP job.

Since the system history file normally contains all the status information of the system, you should always keep a backup copy of it.

EXTent=start-track:tracks:
Specifies the extent information for the history file.

Start-track specifies the sequential number of the track (relative to zero) where the extent is to begin.
For FBA devices, start-track indicates the block number at which the extent is to start.

Tracks specifies the number of tracks (or FBA blocks) to be allocated to the history file.

The following list gives suggested sizes of the extents required for the various device types; the values apply to both system and auxiliary history files. (The values in brackets show the maximum sizes MSHP uses.)

2314/2319:	80 (289) tracks
3330/3333:	38 (145) tracks
3340/3344:	96 (289) tracks
3350:	30 (109) tracks
3375:	15 (61) tracks
3380:	15 (46) tracks
FBA DASDs:	900 (3460) FBA blocks

SPlit=split-track:
Specifies, for CKD devices, which track is the last one in each cylinder to be allocated to the history file. (The first cylinder occupied by the file is the one in which the "start-track" lies, and the last cylinder is determined by the number of tracks specified.)

Split-track is a decimal integer equal to the number of tracks per cylinder minus one.

UNIT=SYSxxx:
Specifies the logical unit (other than SYSREC) on which the history file is to reside.

Defaults: If not specified, MSHP takes the following defaults:

- For a system history file: SYSREC
- For an auxiliary history file: SYS002

IDentifier='file-identifier':
Specifies the history file identification that is to be entered in the VTOC. (1 to 44 alphameric characters enclosed in quotes.)

Defaults: If the operand is not specified, MSHP takes the following defaults:

- For an auxiliary history file: 'DOS.AUXILIARY.HISTORY.FILE'
- For the system history file: 'DOS.SYSTEM.HISTORY.FILE'

Operation	Operands
DELete	[from-line[+rel]] : to-line[+rel]

The DELETE statement indicates the lines to be deleted from a macro (definition) when applying a local/APAR fix.

Description of Operands

from-line:
Specifies the line-number (in columns 73 through 78 in the macro) where deletion begins. The from-line is the first line to be deleted.

Default: If omitted, from-line is assumed to be equal to the to-line value. This means that only the line designated by to-line is deleted.

+rel:
Identifies the position of the statement relative to the from-line number and is an integer of one or two digits.
It applies to E- or F-type macros only.

to-line:
Identifies the last line of the lines to be deleted. The value of to-line must be equal to or greater than the value given in from-line.

+rel:
Identifies the position of the statement relative to the to-line number and is an integer of one or two digits.
It applies to E- or F-type macros only.

Operation	Operands
EXCLude	{PRoduct=(product,...) |COmponent=(component[-level],...) |PTf=(ptf-number,...)}

The EXCLUDE statement is used to exclude specific products, components or PTFs from a service application (with the INSTALL SERVICE statement). This implicitly includes service for all other products, components, or PTFs shipped on the tape.

Description of Operands

PRoduct=(product,...):
Specifies the product(s) that are not to be serviced.

COmponent=(component[-level],...):
Specifies the component(s) that are not to be serviced. If level is not specified, MSHP excludes all levels of the component.

PTf=(ptf-number,...):
Lists the PTFs that are not to be installed.

<table>
<tr><th>Operation</th><th>Operands</th></tr>
<tr><td>EXECute</td><td>(control-program,...)[Xref|NOXref]</td></tr>
</table>

The EXECUTE statement is used to indicate which system programs (assembler, librarian or linkage editor) are to be called in which order to process the data submitted with the TAILOR statement. The data to be processed must immediately follow the EXECUTE statement and be terminated by /$.

Description of Operands

(control-program,...):
MSHP calls the specified system program(s) in the submitted order to process the data which immediately follows the EXECUTE statement (and is terminated by /$).

If two programs are specified, the output of the first program is taken as input to the second without any modification.

Any mismatch between the program and the data (for example, an object deck as input for ASSEMBLY) is not checked by MSHP, but results in an error situation diagnosed by the called program.

The following programs or program combinations can be specified:

- EXEC ASSEMBLY
  EXEC LNKEDT
- EXEC ASSEMBLY
  EXEC LIBR
- EXEC ASSEMBLY,LNKEDT
- EXEC ASSEMBLY,LIBR
- EXEC LNKEDT
- EXEC LIBR

Xref:
Specifies that the cross-reference list of included macros as given by the ASSEMBLY program is to be recorded in the history file.

NOXref:
Specifies that the cross-reference list of included macros is not to be recorded in the history file.

<table>
<tr><th>Operation</th><th>Operands</th></tr>
<tr><td>GENerate</td><td>{PHAse=member-name|MODule=member-name
|MACro=member-name [TYpe=member-type]}</td></tr>
</table>

The GENERATE statement is used as a detail control statement to the SELECT statement to regenerate (retailor) individual phases, modules, or macros from the generation file.

Description of Operands

PHAse=member-name:
Indicates to MSHP the name of the phase that is to be regenerated.

MODule=member-name:
Indicates to MSHP the name of the module that is to be regenerated.

MACro=member-name:
Indicates to MSHP the name of the macro that is to be regenerated.

TYpe=member-type:
Indicates the type of the affected macro (can be one character only). If the operand is omitted, type E is assumed.

<table>
<tr><th>Operation</th><th>Operands</th></tr>
<tr><td>INCLude</td><td>{PRoduct=(product,...)
|COmponent=(component[-level],...)
|PTf=(ptf-number,...)}</td></tr>
</table>

The INCLUDE statement is used to indicate to MSHP that only the named products, components, or PTFs are to be included in a service application (with INSTALL SERVICE). This implicitly excludes service for all other products, components, or PTFs shipped on the service tape.

Description of Operands

PRoduct=(product,...):
Specifies the product(s) to which service is to be applied.

COmponent=(component[-level],...):
Specifies the component(s) to which service is to be applied.

If level is omitted, all levels of the component are serviced.

PTf=(ptf-number,...):
Lists the PTFs that are to be applied.

<table>
<tr><th>Operation</th><th>Operands</th></tr>
<tr><td>INFluences</td><td>{PHAse=(member-name,...)|MODule=(member-name,...)
|MACro=(member-name,...) [TYpe=member-type]}</td></tr>
</table>

The INFLUENCES statement identifies which generated phases, modules, or macros of the serviced component are affected by a PTF or local/APAR fix and have to be regenerated.

Restriction: One INFLUENCES statement may not refer to more than a total of 100 phases, modules, and/or macros.

Description of Operands

PHAse=(member-name,...):
Names the phases to be regenerated.

MODule=(member-name,...):
Names the modules to be regenerated.

MACro=(member-name,...):
Names the macros to be regenerated.

TYpe=member-type:
Indicates the type of the affected macro (can be one character only). If the operand is omitted, type E is assumed.

Operation	Operands
INsert	after-line[+rel]

The INSERT statement identifies where, in a source book (macro), additions are to be made when archiving a local/APAR fix or when initiating a local or APAR fix by means of the CORRECT statement. The statement further serves as the initiating delimiter (with /$ or a blank line being the terminating delimiter) for the input line to be inserted.

Description of Operands

after-line:
Specifies the line number in the macro (in columns 73 through 78) after which the source input (following the INSERT statement up to the next /$) is to be inserted (1 to 6 digits).

+rel:
Specifies the position of the source input relative to the after-line number (1 or 2 digits).

<u>Restriction</u>: rel applies to E- or F-type macros only.

Operation	Operands
INVolves	LINK=(link-book,...)

The INVOLVES statement explicitly requests link-editing to be performed when installing an archived product, or when applying PTFs from a service tape.

As a detail control statement to APPLY, INCORPORATE, and CORRECT, it indicates that, as the final step of the particular function, a link-edit run must be performed.

Description of Operands

LINK=(link-book,...):
Link-book specifies the name of a module that is to be included in the link-edit step. (1 to 8 characters, the first one must <u>not</u> be an asterisk.)

If you specify several link-books, the linkage editor includes the named modules in the same sequence as they occur in the list. You can specify up to 100 link-books.

Operation	Operands
OR	no operand

The OR statement initiates a set of alternative REQUIRES statements that are to be checked in case the preceding set of requirements is not met.

Two or more REQUIRES statements following each other immediately are considered to be in an 'AND' relation. This means that the REQUIRES check is successful only if the prerequisites, corequisites, and negative prerequisites of the whole set of REQUIRES statements are met.

Operation	Operands
PTF	=(ptf-number,...)

The PTF statement is used as a detail control statement to the LIST SERVICETAPE COVER statement to print selected cover letters.

Description of Operands

=ptf-number:
Identifies the PTF whose cover letter is to be printed.

Operation	Operands
REPlace	[from-line[+rel]] : to-line[+rel]

The REPLACE statement is used when applying (CORRECT) or archiving (ARCHIVE) a local or APAR fix to define where replacement of lines in a source macro must begin and end.
The replacing data must follow immediately the REPLACE statement and is to be terminated by an input line containing /$ in columns 1 and 2 (or a blank line if entered from SYSLOG).

Description of Operands

from-line:
Specifies, by the line-number in columns 73 through 78 in the macro, the first line to be deleted and to be replaced by the first (if any) input line. (Input refers to data that follows the REPLACE statement.)

Default: If from-line is not specified, it is assumed to be equal to to-line.

+rel:
Specifies the position of the line relative to the from-line number (1 or 2 digits).

Restriction: rel applies to E- or F-type macros only.

to-line:
Specifies that, beginning with from-line, all lines in the macro are to be deleted up to and including the line indicated by 'to-line'. 'To-line' is the line-number contained in columns 73 through 78 of the macro to be modified.

+rel:
Specifies the position of the line relative to the to-line number (1 or 2 digits).

Restriction: rel applies to E- or F-type macros only.

Operation	Operands
REQuires	[component][PRE=req-list][CO=req-list][NOT=req-list]

The REQUIRES statement is used to specify the requirements (such as prerequisite PTFs) that must be met to successfully install a shipment package or apply service in PTF or local/APAR fix format.
The specified requirements are entered in the history file that accompanies the programming package.

You may connect several requirements (with an 'AND' relation) by specifying several REQuires statements in succession.

You can also delimit such a set of REQUIRES statements from a preceding set by means of the OR statement.
If the preceding set of requirements (at least one) fails, MSHP tests the set of requirements initiated by OR.
If that test is successful, all requirements are considered to be met.

Restrictions: The number of requirements per PTF, local/APAR fix, component, or product specified in one or more REQUIRES statements must not exceed 88.
At least one of the operands PRE=, CO=, or NOT= must be present.

Description of Operands

component:
If the requirements specified in the req-lists are PTFs or local/APAR fixes, then component specifies the component to which the PTF or local/APAR fix belongs.

Default: If component is omitted, then the PTFs or local/APAR fix(es) specified as requirements are assumed to belong to the component to which the "requiring" PTF or local/APAR fix is applied.

Restrictions: Component must not be specified if the requirement in a req-list is neither a PTF nor a local/APAR fix.
Component must always be specified if REQUIRES is used in conjunction with the ARCHIVE statement.

PRE:
Indicates that the requirements specified in the req-list have to be installed prior to the requested service application or installation function.

A prerequisite condition is also considered as being met if a prerequisite PTF has been superseded by another, installed PTF.

CO.
Indicates that the requirements specified in the req-list have to be applied together with the requested service application or installation function.

If REQUIRES is used as a detail control statement to CORRECT, CO= indicates that the requesting local/APAR fix will be applied even though the requirements specified in the req-list are not met; however, MSHP issues a warning message.

NOT:
Indicates that the requirements specified in the req-list must not be installed prior to the requesting service application or installation function.

req-list:
A 'req' is one of the following:

- PTF number or local/APAR fix number
- component[-level]
- product (or old feature number)

Restriction: In a requirements list, all items must be of the same type: PTF numbers, APAR numbers, components, and products may not be mixed.

Operation	Operands
RESolves	['comment'][APARs=(apar-number,...)]

The RESOLVES statement associates a comment with a product, a PTF, a local/APAR fix. or a generated member: it is also used to indicate which APARs are fixed bv a PTF.

Restriction: Only one comment per associated product (or fix or member) can be recorded in the history file.
If more than one RESOLVES 'comment' statement is specified, the last one will be recorded.

Description of Operands

'comment'.
Specifies that a comment relating to a PTF. a local/APAR fix. a product. or a generated member is to be inserted in the history file.
The maximum length is 55 characters if the comment is associated with a local/APAR fix; it is 57 for any other comment (enclosed in quotes).

APARs=(apar-number,...):

Specifies the APAR number(s) corrected by a given PTF.

Restrictions:

- This operand must be specified in a RESOLVES statement that relates to a PTF (RESOLVES being used in conjunction with ARCHIVE PTF and APPLY component:ptf-number).
- The maximum number of APARs that can be specified in one RESOLVES statement is 100.

Operation	Operands
RESTart	restart-line[+rel]

The RESTART statement is used for the correction of edited macros (with the CORRECT statement). It indicates that a new sequence number series starts after the specified statement.

Description of Operands

restart-line:

Specifies the sequence number of the statement after which the new sequence number series starts.

+rel:

Specifies the position of the desired statement relative to 'restart-line'.

Operation	Operands
SCan	[offset] [ARGument= 'char-string'|hex-string}]

The SCAN statement is used when correcting a phase from the console (after AFFECTS PHASES=...) to search for a specified string in a phase and to display 16 bytes of the phase.

Description of Operands

offset:

Specifies the displacement (relative to the beginning of the phase) where, in the phase, the search for the specified ARGUMENT string is to be started. If the ARGUMENT=string operand is omitted, MSHP displays 16 bytes of the phase, starting at 'offset'. 'offset' is a number of up to six hexadecimal digits; leading zeros may be omitted.

ARGument=['char-string'|hex-string]:

Specifies the string that is to be searched in the phase. It can be in one of the following formats:

- A string of 1 to 16 characters, enclosed in quotes, where each character represents one byte in the phase.
- An even string of 2 to 32 hexadecimal digits, where each pair of hexadecimal digits describes one byte in the phase.

The table shows the results of specifying the two operands 'offset' and 'ARGUMENT' in various combinations.

	offset	ARG	Result
First scan after AFFECTS PHASES	–	–	Invalid; error message.
	–	x	Scanning for specified string from offset 0.
	x	–	Display of 16 bytes from specified offset.
	x	x	Scanning for indicated string from specified offset.
Subsequent scan	–	–	Scanning from current offset for old argument string, which must be known from preceding scan request.
	–	x	Scanning for specified string from current offset.
	x	–	Display of 16 bytes from specified offset.
	x	x	Scanning for indicated string from specified offset.

Operation	Operands
SUPersedes	(ptf–number,...)

The SUPERSEDES statement identifies which PTFs are superseded by a given PTF when that PTF is being built.
MSHP requires the list of superseded PTFs to be complete.
For example: If PTF2 supersedes PTF1, and subsequently a PTF3 is issued that supersedes PTF2, then PTF3 must be specified as also superseding PTF1.

Description of Operands

(ptf-number,...):
Specifies the PTF(s) that are superseded.

Restriction: The maximum number of PTFs that can be specified as superseded in one SUPERSEDES statement is 100. The total number that can be specified is 255.

Operation	Operands
VERify	verify–line[+rel]

The VERIFY statement designates where, in a source book, a verification is to be made for a local or APAR fix correction.
The VERIFY statement must be followed by a single line of text.
MSHP checks whether this text is present in the statement indicated by 'verify-line'.

Description of Operands

verify-line:
Specifies the sequence number of the source statement to be verified (1 to 6 decimal digits).

rel:
Specifies the position of the desired statement in relation to the statement number indicated for verify-line (1 or 2 digits).

Restriction: rel applies to E- or F-type books only.

APPENDIX A. RETURN CODES

Job Control Return Codes

Return Codes from $JOBEXIT

```
R15: zero     = continue processing
     not zero = treat statement as comment
```

Note: If R5 is not zero, the 80-byte long continuation card(s) is (are) located adjacent to the 80-byte long statement.

Return Codes from Phase $IJBASGN in Register 15

00 - Assignment successful
04 - No free LUB entry found
08 - Device cuu not found in PUB table
0C - Device cuu is not a disk
10 - Device cuu is down
18 - No free tape unit found
1C - Invalid logical unit for unassignment
20 - Device reserved by volume statement or by mount request from another partition
24 - Invalid function code
28 - No GETVIS space available
2C - Device to be unassigned is not assigned

Return Codes form Phase $IJBCJC in Register 15

00 - Function successfully executed
04 - Wanted information not available
08 - Invalid parameter field
0C - No GETVIS space available

Return Codes from Phase IJBPROC in Register 15

1. If called by macro PARMMAC

   00 - Request was successful
   08 - Invalid length in LENFLD
   0C - Invalid pointer for a buffer parameter
   10 - Parameter not defined in GETVAL-request
   14 - SETPDF-request occurred twice
   18 - SETPDF-request occurred after second GETREC
   1C - Space block chain exhausted
   20 - No system GETVIS space available
   2C - No partition GETVIS space available
   40 - Invalid function

2. If called by macro PROCMAC

   00 - Request was successful
   04 - Procedure not found
   08 - EOPREQ was given on level 0
   0C - GETREC was given on level 0
   10 - ACCESS exceeds nesting level of 15
   14 - Duplicate procedure name in nested stack
   18 - Request outside member
   1C - Invalid pointer to buffer
   20 - No system GETVIS space available
   24 - Librarian error
   28 - Conflict in nested stack related to DATA=YES/NO option
   2C - No partition GETVIS space available
   30 - Error in LABEL request
   34 - Partition FREEVIS failed
   40 - Invalid function

Librarian Return Codes

0 .... requested function successfully completed

4, 8, 12 .... function-specific exceptional conditions

16 .... external error (e.g. system resources exhausted, invalid parameter list when called)

20 .... internal error (e.g. processing consistency check failed)

32 .... security violation

36 .... library chaining error (library block identification number changes within a chain)

Additional to return codes 16 and 20, a feedback code is returned in the corresponding parameter list (usually in the LRPL) denoting the reason for the failure.

Return codes which are equal or greater than 16 are accompanied by a message which is returned in the message area for SYSLST and SYSLOG and printed or logged according to the message routing indicator if the corresponding DTFs are provided via the LAMB. If neither DTFs nor message areas are specified in the LAMB, only the message number is returned in the LAMB.

Linkage Editor Return Codes

00 - Successful completion

04 - Warning or error message issued, but phases are cataloged

16 - Severe error occurred, phases are not cataloged

For return code 0 one of the following warning messages might occur:

- ROOT STRUCTURE OVERLAID BY SUCCEEDING PHASE
- POSSIBLE INVALID ENTRY POINT DUPLICATION IN INPUT
- INVALID TRANSFER LABEL ON END OR ENTRY STATEMENT IGNORED
- CONTROL SECTION OF ZERO LENGTH IN INPUT
- and message 2139I

For return code 4 one of the following error messages did occur:

- One of the messages 2100I through 2170I (except 2139I) or
- ** MODULE xxxxxxxx NOT FOUND or
- UNRESOLVED EXTERNAL REFERENCES
- and messages 2187I and 2199I

For return code 16 one of the following error messages did occur:

- One of the messages 2180I through 2197I (except 2187I) or
- a librarian message LxxxI.

APPENDIX B. SENSE INFORMATION SUMMARY

1287 - Optical Reader

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4: data check
	5: overrun
	6: non-recovery
	7: keyboard correction (tape only)
1	Bit0: tape mode
	1: late stacker reset
	2: no document found
	3: --
	4: invalid option
	5: --
	6: --
	7: --

1288 - Optical Reader

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4: data check
	5: overrun
	6: non-recovery
	7: --
1	Bit0: --
	1: end-of-page
	2: no document found
	3: --
	4: invalid option
	5: --
	6: --
	7: --

1403 - Printer

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4: data check
	5: parity check (UCS storage)
	6: --
	7: channel 9
1	Not used
2	Bit0: chain interlock
	1: forms check
	2: coil protect check
	3: subscan ring check
	4: chain buffer address register check
	5: --
	6: any hammer on check
	7: --
3	Not used

1403 - Printer (cont....)

Sense Byte	Designation
4	Bit0: hammer reset failure check
	1: no fire check
	2: misfire check
	3: print data buffer parity check
	4: check bit buffer parity check
	5: chain buffer parity check
	6: buffer address register parity check
	7: clock check
5	Bit0: open hammer coil check
	1-7: --

1419 - PCU-MICR

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: --
	4: data check
	5: overrun
	6: autoselect
	7: --
1	Bit0-1: --
	2: document under read head
	3: amount field valid
	4: process-control field valid
	5: account-number field valid
	6: transit field valid
	7: serial-number field valid

1419 - SCU-MICR

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3-4: --
	5: late stacker select
	6: autoselect
	7: operator attention

1442 - Card Read-Punch/Card Punch

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4: data check
	5: overrun
	6-7: --

1443 - Printer

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4: type bar
	5: type bar
	6-7: --

2260 - Display Station

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4-7: --

2311 - Disk Storage

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4: data check
	5: overrun
	6: track condition check
	7: seek check
1	Bit0: data check in count area
	1: track overrun
	2: end of cylinder
	3: invalid sequence
	4: no record found
	5: file protect
	6: missing address marker
	7: overflow incomplete
2	Bit0: unsafe
	1: --
	2: serializer check
	3: --
	4: ALU check
	5: unselected file status
	6: --
	7: --
3	Bit0: ready
	1: online
	2: unsafe
	3: --
	4: online
	5: end of cylinder
	6: --
	7: seek incomplete
4	Bit0-7: --
5	Bit0-7: command in progress when overflow incomplete occurs

2314/2319 - Direct Access Storage

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4: data check
	5: overrun
	6: track condition check
	7: seek check
1	Bit0: data check in count area
	1: track overrun
	2: end of cylinder
	3: invalid sequence
	4: no record found
	5: file protect
	6: missing address marker
	7: overflow incomplete
2	Bit0: unsafe
	1: --
	2: SERDES check
	3: --
	4: ALU check
	5: unselected file status
	6: --
	7: --

2314/2319 - Direct Access Storage (cont....)

Sense Byte	Designation
3	Bit0: busy
	1: online
	2: unsafe
	3: write current sense
	4: pack change
	5: end of cylinder
	6: multy-module select
	7: seek incomplete
4	Bit0: wrong length record (2314 with multiplex storage control feature only)
	1: pending status (2314 with multiplex storage control feature only)
	2: --
	3: --
	4-7: Module identification bits 4567 physical drive 0000 A 0001 B 0010 C 0011 D 0100 E 0101 F 0110 G 0111 H 1000 J 1111 module not defined
5	Bit0-7: command in progress when overflow incomplete occurs

2400 - Magnetic Tape

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4: data check
	5: overrun
	6: word count zero
	7: data converter check
1	Bit0: noise
	1-2: B'00'= not existent B'01'= not ready B'10'= ready and not rewinding B'11'= ready and rewinding
	3: seven track
	4: at loadpoint
	5: selected and write status
	6: file protect
	7: not capable
2	Bit0-7: contains the track-in-error indicator bits that are set at the end of a read, or read-backward command if a data check has been encountered. Bits 6 and 7 on together indicate either more than one error or no record found.
3	Bit0: R/W VRC
	1: LRCR
	2: skew
	3: CRC
	4: skew register VRC
	5: phase encoding
	6: backward
	7: C compare

2400 - Magnetic Tape (cont....)

Sense Byte	Designation
4	Bit0: echo check
	1: reject TU
	2: read clock error
	3: write clock error
	4: delay counter
	5: sequence indicator C
	6: sequence indicator B
	7: sequence indicator A

2501 - Card Reader

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4: data check
	5: overrun
	6: --
	7: --

2520 - Card Read-Punch/Card Punch

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4: data check
	5: overrun
	6: --
	7: --

2540 - Card Reader/Card Punch

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4: data check
	5: --
	6: unusual command
	7: --

2560 - Multifunction Card Machine

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: --
	3: equipment check
	4: data check
	5: feed/Machine check
	6: no card available
	7: print operation in progress
1	Bit0: cover interlock/punch pusher check
	1: jam bar check
	2: corner station check
	3: call 8 to 9 feed check
	4: print station feed check
	5: punch station feed check
	6: read station feed check
	7: input station SC1 exposed

2560 - Multifunction Card Machine (cont....)

Sense Byte	Designation
2	Location of individual card: Bit0: secondary select 1: card in punch station 2: preprint SC7 exposed 3: prepunch SC5 exposed 4: prepunch SC4 exposed 5: preread SC3 exposed 6: preread SC2 exposed 7: input station SC1 exposed
3	Stacker Select Information Bit0: primary card 0) 1: binary value 4) card at primary 2: binary value 2) prepunch station 3: binary value 1) 4: secondary card 1) 5: binary value 4) card at secondary 6: binary value 2) prepunch station 7: binary value 1)
4	Stacker Select Information Bit0: primary (0)) secondary (1)) 1: binary value 4) card at punch or 2: binary value 2) preprint station 3: binary value 1) 4: primary (0)) secondary (1)) 5: binary value 4) card after print 6: binary value 2) station 7: binary value 1)
5	Stacker Select Information Bit0: primary (0)) secondary (1)) 1: binary value 4) card at corner 2: binary value 2) station 3: binary value 1) 4: primary (0)) secondary (1)) 5: binary value 4) card in stacker pocket 6: binary value 2) (was just stacked) 7: binary value 1)
6	Card column in which first (possibly only) error was detected: Bit0: multi data check 1: binary value 64 2: binary value 32 3: binary value 16 4: binary value 8 5: binary value 4 6: binary value 2 7: binary value 1

2596 - Card Read-Punch

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: bus-out check 3: equipment check 4: data check 5: overrun 6: -- 7: --

3203 - Printer

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: --
	3: equipment check
	4: data check
	5: chain buffer parity check
	6: no channel found
	7: channel 9
1	Bit0-7: --
2	Bit0: interlock (chain gate open)
	1: forms check (jam)
	2: coil protect check
	3: subscan ring check
	4: chain buffer address register check
	5: hammer unit shift check (model 1 only)
	6: any hammer on check
	7: device ready check
3	Bit 0: --
	1: --
	2: --
	3: carriage inhibit check
	4: --
	5: --
	6: step check
	7: move check
4	Bit0: hammer reset failure check
	1: no fire check
	2: misfire check
	3: print data buffer parity check
	4: check bit buffer parity check
	5: chain buffer parity check
	6: buffer address register check
	7: clock check
5	Bit0: open coil check
	1-7: --

3210/3215 - Console Printer Keyboard

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: --
	3: equipment check
	4-7: --

PRT1 (3203-4,3203-5,3211,3289,4248)

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4: data check
	5: buffer parity check
	6: load check
	7: channel 9
1	Bit0: command parity
	1: print check
	2: print quality
	3: line position check
	4: forms check
	5: command suppress
	6: mechanical motion
	7: --

PRT1 (3203-4,3203-5,3211,3289,4248) (cont....)

Sense Byte	Designation
2	Bit0: carriage failed to move
	1: carriage sequence check
	2: carriage stop
	3: platen failed to advance
	4: platen failed to retract
	5: forms jam
	6: ribbon motion
	7: train overload
3	Bit 0: UCSB parity
	1: PLB parity
	2: FCB parity
	3: coil protect
	4: hammer fire check
	5: service aid
	6: UCSAR sync check
	7: PSE sync check
4	Bit0-7: information used by service personnel
5	Bit0-7: --

3272 - (3270 Local)

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4: data check
	5: unit specify
	6: control check
	7: operation check

FBA (3310 and 3370) Disk Storage

Sense Byte	Designation	Format
0	Bit0: command reject	0
	1: intervention required	1
	2: bus-out parity (not used)	0
	3: equipment check	1
	4: data check	4
	5: overrun	0
	6: (unused)	-
	7: (unused)	-
1	Bit0: permanent error	any
	1: (unused)	-
	2: (unused)	-
	3: (unused)	-
	4: (unused)	-
	5: file protected	0
	6: write inhibit	-
	7: operation incomplete	0
2	Bit0: check data error	4
	1: correctable	5
	2: (unused)	-
	3: environmental data present	6,4,1
	4: (unused)	-
	5: (unused)	-
	6: (unused)	-
	7: only logging required	-
3-6	physical address	
7	Bit0-3: hex: format number	
	4-7: hex: message code	
8-23	only for diagnose information	

3330 - Disk Storage

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: bus-out check 3: equipment check 4: data check 5: overrun 6: -- 7: --
1	Bit0: permanent error 1: invalid track format 2: end of cylinder 3: -- 4: no record found 5: file protected 6: write inhibited 7: operation incomplete
2	Bit0: -- 1: correctable 2: -- 3: environmental data present 4: -- 5: -- 6: -- 7: --
3	Bit0-7: restart command
4	Bit0-1: storage control identification 2-7: physical drive identification bits 4567 physical drive 111000 A 110001 B 101010 C 100011 D 011100 E 010101 F 001110 G 000111 H
5	Bit0-7: identify the eight low-order bits of the cylinder address in the most recent seek argument
6	Bit0: reserve 1: cylinder number (high order bit of cylinder address) 2: difference 3: 16) 4: 8) 5: 4) head number 6: 2) 7: 1)
7	Bit0-3: format type of remaining sense bytes (8-23) 4-7: encoded error message
8-23	Meaning depends on format type

3340 - Disk Storage

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: bus-out check 3: equipment check 4: data check 5: overrun 6: track condition check 7: seek check

3340 - Disk Storage (cont....)

Sense Byte	Designation
1	Bit0: permanent error 1: invalid track format 2: end of cylinder 3: -- 4: no record found 5: file protected 6: write inhibit 7: operation incomplete
2	Bit0: RPS feature present 1: correctable 2: -- 3: environmental data present 4: -- 5: data module size 70F Fixed Head 6: data module size 70MB 6: data module size 35MB
3	Bit0-7: restart command
4	Physical drive identification Bit0: drive A 1: drive B 2: drive C 3: drive D 4: drive E 5: drive F 6: drive G 7: drive H
5	Bit0-7: identifies the eight low-order bits of the cylinder address in the most recent seek argument
6	Bit0-2: identifies the three high-order bits of the cylinder address 3: -- 4: 8) 5: 4) head number 6: 2) 7: 1)
7	Bit0-3: format type of remaining sense bytes (8-23) 4-7: encoded error message
8-23	Meaning depends on format type

3344 - Disk Storage (3340 Mode)

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: bus-out check 3: equipment check 4: data check 5: overrun 6: track condition check 7: seek check
1	Bit0: permanent error 1: invalid track format 2: end of cylinder 3: -- 4: no record found 5: file protected 6: write inhibit 7: operation incomplete

3344 - Disk Storage (cont....)

Sense Byte	Designation
2	Bit0: RPS feature present 1: correctable 2: -- 3: environmental data present 4: Compatibility mode / 3344 5: HDA size 3 bit 6: HDA size 2 bit 7: HDA size 1 bit
3	Bit0-7: restart command
4	Controller device address Bit0: Controller addr. bit 2 1: Controller addr. bit 1 2:) 3:) Controller Device Address 4:) 5: Device addr. bit 4 6: Device addr. bit 2 7: Device addr. bit 1
5	Bit0-7: identifies the eight low-order bits of the cylinder address
6	Bit0-3: identifies the four high-order bits of the cylinder address 0: 1024 1: 512 2: 256 3: 2048 4: 8) 5: 4) head number 6: 2) 7: 1)
7	Bit0-3: format type of remaining sense bytes (8-23) 4-7: encoded error message
7-23	Meaning depends on format type

3350 - Direct Access Storage

Sense Byte	Designation
1	Bit0: command reject 1: intervention required 2: channel bus-out parity 3: equipment check 4: data check 5: overrun 6: not used 7: not used
1	Bit0: permanent error 1: invalid track format 2: end of cylinder 3: not used 4: no record found 5: file protected 6: write inhibit 7: operation incomplete
2	Bit0: not used 1: correctable 2: not used 3: environmental data present 4: computability mode 5: not used 6: not used 7: not used

3350 - Direct Access Storage (cont....)

Sense Byte	Designation
3	Bit0-7: restart command
4	Physical drive identification
	Bit0: drive A
	1: drive B
	2: drive C
	3: drive D
	4: drive E
	5: drive F
	6: drive G
	7: drive H
5	Bit0-7: low order logical cylinder address
	0: 128
	1: 64
	2: 32
	3: 16
	4: 8
	5: 4
	6: 2
	7: 1
6	Bit0-2: identifies the three high-order bits of the cylinder address
	Bit0: CE cylinder
	1: 3330-11 = 512
	3330- 1 = 256
	2: 3330-11 = 256
	3330- 1 = 0
	3: 16)
	4: 8)
	5: 4) head number
	6: 2)
	7: 1)
7	Bit0-3: format type of remaining sense bytes (8-23)
	4-7: encoded error message
8-23	Meaning depends on format type

3370 Disk Storage

Sense Byte	Designation	Format
0	Bit0: command reject	0
	1: intervention required	1
	2: bus-out parity (not used)	0
	3: equipment check	1
	4: data check	4
	5: overrun	0
	6: (unused)	-
	7: (unused)	-
1	Bit0: permanent error	any
	1: block size exception	-
	2: (unused)	-
	3: operator message	-
	4: (unused)	-
	5: file protected	0
	6: write inhibit	-
	7: operation incomplete	0
2	Bit0: check data error	4
	1: correctable	5
	2: first logged error (logging mode)	-
	3: environmental data present	6,4,1
	4: (unused)	-
	5: (unused)	-
	6: (unused)	-
	7: (unused)	-

3370 Disk Storage (cont....)

Sense Byte	Designation	Format
3	Bit0-7: identifies the high-order cylinder address of the most recent seek	
4	Bit0-7: identifies the low-order cylinder address of the most recent seek	
5	Bit0-7: head address, IAR, or diskette check	
6	Bit0-7: block no., IAR, or storage director ID	
7	Bit0-3: hex: format number 4-7: hex: message code	
8-23	only for diagnose information	

3375 Disk Storage

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: channel bus-out parity 3: equipment check 4: data check 5: overrun 6: not used 7: not used
1	Bit0: permanent error 1: invalid track format 2: end of cylinder 3: message to operator 4: no record found 5: file protected 6: write inhibit 7: not used
2	Bit0: not used 1: correctable data check 2: first logged error (logging mode) 3: environmental data present 4: intent violation (speed matching buffer) 5: imprecise ending (speed matching buffer) 6: not used 7: not used
3	Bit0-7: controller identification
4	Bit0: dual controller 1: not used 2: not used 3-4: controller physical address

Bits 3 4	address
0 0	0
0 1	1
1 0	2
1 1	3

5-7: device physical address

Bits 5 6 7	actuator	module
0 0 0	0	1
0 0 1	1	1
0 1 0	2	2
0 1 1	3	2
1 0 0	4	3
1 0 1	5	3
1 1 0	6	4
1 1 1	7	4

5	Bit0-7: format 1,4,5 - low cylinder address 0-7: format 6 - number of diskette checks

3375 Disk Storage (cont...)

Sense Byte	Designation
6	Bit0-7: format 1,4,5 - high cylinder/head address 0-7: format 6 - storage control identification
7	Bit0-3: format 4-7: message code
8-23	Meaning depends on format type

3380 Disk Storage

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: bus-out parity 3: equipment check 4: data check 5: overrun 6: not used 7: not used
1	Bit0: permanent error 1: invalid track format 2: end of cylinder 3: message to operator 4: no record found 5: file protected 6: write inhibit 7: not used
2	Bit0: not used 1: correctable data check 2: first logged error (logging mode) 3: environmental data present 4: intent violation 5: imprecise ending 6: write operation 7: not used
3	Controller ID if format 1, 2, 6, 7, 8 not used for format 0, 3, 4, 5 Bit0: logical address of the controller 1-6: not used 7: indicates A2 controller
4	Bit0: indicates dynamic path selection function 1: not used 2: logical path error 3: not used 4-7: device address
5	Bit0-7: cylinder-low address
6	Bit0-7: cylinder-high and head

Cylinder address	Head address
0= 0	4= 8
1= 0	5= 4
2= 512	6= 2
3= 256	7= 1

Sense Byte	Designation
7	Bit0-3: hex: format number 4-7: hex: message code
8-23	only for diagnose information

3410/3411 - Magnetic Tape

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: bus-out check 3: equipment check 4: data check 5: overrun 6: word count zero 7: data converter check
1	Bit0: noise 1-2: B'00'= not existent B'01'= not ready B'10'= ready and not busy B'11'= ready and busy 3: seven track 4: at loadpoint 5: write status 6: file protected 7: not capable
2	Bit0-7: track-in-error bits
3	Bit0: VRC 1: multiple track error (PE) or LRC (NRZI) 2: skew 3: end data check (PE) or CRC (NRZI) 4: envelope check (PE only) 5: phase encoding 6: backward 7: --
4	Bit0: tape unit positioning check 1: tape unit reject 2: end of tape 3: -- 4: -- 5: diagnostic track check 6: tape unit check 7: illegal command
5	Bit0: new subsystem 1: new subsystem 2: write tapemark check 3: PE identification burst 4: PE compare 5: tachometer check 6: false end mark 7: RPQ
6	Bit0: seven track 1: short gap mode 2: dual density 3: NRZI density 4-7: tape unit model
7	Bit0: lamp check 1: left column check 2: right column check 3: ready reset 4: data security erase 5-7: --
8	Bit0: -- 1: feedthrugh 2: -- 3: end velocity check 4: no read-back data 5: start velocity check 6: -- 7: --

3420/3803 - Magnetic Tape

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: bus-out check 3: equipment check 4: data check 5: overrun 6: word count zero 7: data converter check
1	Bit0: noise 1-2: B'00'= not exist. B'10'= ready and not rew. B'01'= not ready B'11'= ready and rew. 3: seven track 4: at loadpoint 5: write status 6: file protected 7: not capable
2	Bit0-7: track-in-error bits
3	Bit0: VRC 1: multiple track error (PE) or LRC (NRZI) 2: skew 3: end data check (PE) or CRC (NRZI) 4: envelope check (PE only) 5: phase encoding 6: backward 7: C-compare
4	Bit0: ALU hardware error 1: reject tape unit 2: tape indicate 3: write trigger VRC 4: microprogram detected error 5: LWR 6: tape unit check 7: RPQ
5	Bit0: new subsystem 1: new subsystem 2: write tapemark check 3: PE ID burst check 4: start read check 5: partial record 6: excessive postable or tape mark 7: RPQ
6	Bit0: seven track 1: write current failure 2: dual density 3: NRZI density 4-7: tape unit model
7	Bit0: lamp failure 1: tape bottom left 2: tape bottom right 3: reset key 4: data security erase 5: erase head 6: air bearing pressure 7: load failure
8	Bit0: IBG drop while writing 1: feedthrugh check 2: SDR counter 3: early begin readback check 4: early ending readback check 5: slow begin readback check 6: slow ending readback check 7: velocity retry/restart

3420/3803 - Magnetic Tape (cont....)

Sense Byte	Designation
9	Bit0: SDR counter 1: velocity change during write 2-3: SDR counter 4: -- 5: -- 6: -- 7: tape control reserved
10	Bit0: command status reject 1: -- 2: control status reject 3: no block on record readback check 4: WTM no detected block 5: tachometer start fail 6: -- 7: velocity check
11	Bit0: B bus parity error, ALU 1 1: -- 2: low ROS parity/low IC/parity on br. instr. 3: high IC/high ROS reg parity 4: microprogram detected hardware error 5: D bus parity error, ALU 1 6: -- 7: branch condition error, ALU 2
12	Bit0: B bus parity error, ALU 2 1: -- 2: low ROS parity/low IC/parity on br. instr. 3: high IC/BC/high ROS reg parity 4: microprogram detected hardware error 5: D bus parity error, ALU 2 6: -- 7: branch condition error, ALU 2
13	Bit0-1: tape control desity 2-7: tape control unique ID high
14	Bit0-7: tape control unique ID low
15	Bit0-7: tape unit unique ID
16	Bit0-7: tape unit unique ID
17	Bit0: two channel switch 1-3: tape control device switch features 4-7: EC level of tape control
18	Bit0: power check/air flow 1-3: -- 4-7: EC level of tape unit
19	Bit0: primed for device end unit 7 1: primed for device end unit 6 2: primed for device end unit 5 3: primed for device end unit 4 4: primed for device end unit 3 5: primed for device end unit 2 6: primed for device end unit 1 7: primed for device end unit 0
20	Bit0: primed for device end unit F 1: primed for device end unit E 2: primed for device end unit D 3: primed for device end unit C 4: primed for device end unit B 5: primed for device end unit A 6: primed for device end unit 9 7: primed for device end unit 8

3420/3803 - Magnetic Tape (cont....)

Sense Byte	Designation
21	Bit0: load button depressed 1: left reel turning 2: right reel turning 3: tape present 4: reels loaded 5: load rewind 6: load complete 7: load check
22	Bit0-7: FRU identifiers for tape control
23	Bit0-7: FRU identifiers for tape control

Note: Some bits may have different meaning depending on the model.

3430 - Magnetic Tape

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: bus-out check 3: equipment check 4: data check 5: overrun 6: word count zero 7: always zero
1	Bit0: noise 1-2: B'00'= not present B'01'= not ready B'10'= ready and not rewinding B'11'= ready and rewinding 3: always zero 4: beginning of tape 5: write status 6: file protected 7: not capable
2	Bit0-7: track-in-error bits
3	Bit0: VRC 1: multiple track error 2: skew 3: end data check/GCR/CRC check 4: envelope check (PE/GCR mode) 5: 1600 BPI 6: backward 7: write tape mark check
4	Bit0: BOT failure 1: tape unit reject 2: end of tape (EOT) 3: vacuum failure 4: power driver failure 5: LWR check 6: tape unit positioning check 7: no readback data
5	Bit0: new subsystem 1: new subsystem 2: control unit 4 check 3: ID burst check 4: control unit 1 5: control unit 3 6: false end mark 7: tachometer check
6	Bit0-7: fault symptom code (FSC)

3430 - Magnetic Tape (cont....)

Sense Byte	Designation
7	Bit0-3: FSC pointer 4: data security erase 5: write/erase head check 6: nonreportable checks 7: capstan direction
8	Bit0: load failure 1: short gap mode 2: force log mode 3: velocity check 4: reserved 5: control unit 2 6: left column check 7: right column check

3505/3525 - Card I/O

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: bus-out check 3: equipment check 4: data check 5: -- 6: abnormal format reset 7: permanent error key
1	Bit0: permanent error 1: automatic retry 2: motion malfunction 3: retry after intervention complete 4-7: --
2-3	Used for diagnostic purposes only

3540 - Diskette

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: bus-out check 3: equipment check 4: data check 5-7: --
1	Bit0: permanent error 1: automatic retry 2: motion malfunction 3: retry after intervention complete 4: special record transferred 5-7: --
2	Used for diagnostic purposes only
3	Bit0-7: cylinder address in binary
4	Bit0-7: head address, must be binary zero
5	Bit0-7: record address in binary

3800 - Printer Subsystem

Sense Byte	Designation
0-2	Condition defined by byte 0 is further defined by bits turned on in bytes 1 and 2.
0	Bit0: command reject
1	Bit0: invalid command 1-7: reserved
2	Bit0-7: reserved

3800 - Printer Subsystem (cont....)

Sense Byte	Designation
0	Bit1: intervention required
1	Bit0: not ready
	1: operation check
	2: toner collector full
	3: toner supply empty
	4: developer replacement required
	5: end of forms
	6: output full
	7: reserved
2	Bit0: forms overlay check
	1: transfer check
	2: fuser check
	3: CFS check
	4: process check
	5: BTS check
	6: reserved
	7: line overrun
0	Bit2: bus out parity
1	Bit0: command code
	1: data byte
	2-7: reserved
2	Bit0-7: reserved
0	Bit3: equipment check
1	Bit0: hardware error
	1: permanent error
	2: internal log full
	3: cancel key
	4-7: reserved
2	Bit0-7: reserved
0	Bit4: data check
1	Bit0: unprintable character
	1: reserved
	2: no translate table
	3: no FCB channel code match
	4: multiple characters
	5-7: reserved
2	Bit0-7: reserved
0	Bit5: reserved
1	Bit0-7: reserved
2	Bit0-7: reserved
0	Bit6: load check
1	Bit0: incorrect length
	1: incorrect multiple of 6, 8, 12LPI
	2: FCB 1/2-inch error
	3: invalid FCB channel codes
	4: FCB length check
	5: WCGM not loaded
	6: unassigned graphic character
	7: reserved
2	Bit0: invalid WCGM ID
	1: no ID for WCGM 00
	2: invalid copy modification
	3: invalid forms overlay sequence
	4: invalid graphic modification
	5: WCGM data parity error
	6-7: reserved
0	Bit7: channel 9
1	Bit0-7: reserved
2	Bit0-7: reserved

3881 - Optical Mark Reader

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: bus-out check 3: equipment check 4-5: -- 6: unusual command sequence 7: --

3886 - Optical Character Reader

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: bus-out check 3: equipment check 4-5: -- 6: non-initialized 7: RCP error
1	Bit0: -- 1: mark check 2: invalid format 3: -- 4: incomplete scan 5: -- 6: non-recovery 7: outboard

4245 - Printer

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: bus-out check 3: equipment check 4: data check 5: buffer parity check 6: load FCB check 7: channel 9
1	Bit0-2: not used 3: line position check 4: forms check 5: command suppress 6: controller check 7: not used
2	Bit0: carriage fail to move 1: carriage motion check 2-4: not used 5: forms jam 6: not used 7: band velocity check
3	Bit0-2: not used 3: coil protect 4: hammer fire check 5: not used 6: sync check 7: ribbon check
4	Always X'84'

4248 - Printer

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4: data check
	5: buffer parity check
	6: load check
	7: channel 9
1	Action code
	Hex 00: no error exists
	01: data or program check
	02: intervention required
	03: normal intervention
	04. channel check
	05. error logging crippled
	06: microcode error, recoverable
	07: microcode error, unrecoverable
	0A: channel 9 detected
	0B: retries exceeded threshold
	10: unrecoverable error in rotate mode
2	Unit Code Area
	Hex 00: not defined
	01: power
	02: IPL task
	03: level 0 handler & microcode hardw.
	04: 804B display unit
	05: not defined
	06: printer adapter & CAC
	07: printer adapter & CAC
	08: PCA & CAC
	09: channel task
	0A: buffer sync task (print task)
	0B: not defined
	0C: not defined
	0D: SLIM file & CAC
	0E: internal u-code
	0F: printer maintenance task
3	Error Code

5424/5425 - Multifunction Card Unit

Sense Byte	Designation
0	Bit0: command reject
	1: intervention required
	2: bus-out check
	3: equipment check
	4: data check
	5: --
	6: no card available
	7: --
1	Bit0: read check
	1: punch check
	1: --
	3: print data check
	4: print clutch check
	5: hopper check
	6: feed check
	7: --

5424/5425 - Multifunction Card Unit (cont....)

Sense Byte	Designation
2	Bit0: -- 1: -- 2: card in primary wait station 3: card in secondary wait station 4: NPRO allowed 5: hopper cycle not complete 6: card in transport counter bit 2 7: card in transport counter bit 1
3	Contains a hexadecimal number whose value can represent feedchecks and emitter checks in the 5425
4	Defines the card column group and tier where the error was detected which caused the first read check or punch check of a card cycle Bit0: multiple error 1-2: B'00' tier 1 B'01' tier 2 B'10' tier 3 3-7: B'00000' column group 1 B'00001' column group 2 ''''' ''''' B'11111' column group 32
5	Bit0: D row miscompare 1: C row miscompare 2: B row miscompare 3: A row miscompare 4: 8 row miscompare 5: 4 row miscompare 6: 2 row miscompare 7: 1 row miscompare
6-10	Forms a table of the five most recent command strings Bit0: secondary 1: print four lines 2: stacker select M2 3: stacker select M3 4: punch 5: feed command sample 6: print 7: read

8809 - Tape Unit

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: -- 3: equipment check 4: data check 5: overrun 6-7: --
1	Bit0: noise 1: tape unit status A 2: tape unit status B 3: -- 4: at loadpoint 5: write status 6: file protected 7: not capable

8809 - Tape Unit (cont....)

Sense Byte	Designation
2	Bit0-7: represent track-in-error pointers
3	Bit0: 128 ERP number 1: 64 ERP number 2: 32 ERP number 3: 16 ERP number 4: 8 ERP number 5: 4 ERP number 6: 2 ERP number 7: 1 ERP number
4	Bit0: -- 1: -- 2: tape indicate 3: permanent error 4: host detected error 5: loop write to read error 6-7: not used
5	Bit0-2: -- 3: PE ID burst check 4-7: --
6	Contains all zeroes
7	Bit0: 8 format code 1: 4 format code 2: 2 format code 3: 1 format code 4: data security erase 5-7: --
8-31	Sense bytes 8-31 are only used for hardware diagnostics

DOC - Display Operator Console

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: -- 3: equipment check 4: -- 5: -- 6: -- 7: operation check

TP - Teleprocessing Devices

Sense Byte	Designation
0	Bit0: command reject 1: intervention required 2: bus out check 3: equipment check 4: data check 5: overrun 6: lost data 7: timeout

APPENDIX C. HARD AND SOFT WAIT CODES

BYTE 0	BYTE 1	BYTE 2	BYTE 3	EXPLANATION
MCH/CCH Hard Wait Code placed in low address storage				
X'C1'	X'00'	A,I,S	Not used	Irrecoverable machine check.
X'C2'	X'00'	A,I,S	Not used	Irrecoverable channel check during fetch.
X'C3'	X'00'	A,I,S	Not used	Irrecoverable channel check on paging channel.
X'C5'	X'00'	A,I,S	Not used	No ECSW stored.
X'C7'	X'00'	A,I,S	Not used	Channel failure; channel address invalid.
X'C8'	X'00'	A,I,S	Not used	Channel failure on SYSLOG.
Notes: A C'C1' - SYSREC recording unsuccessful (NO record written) I C'C9' - SYSREC recording incomplete (Not all records written) S C'E2' - SYSREC recording successfully completed				
IPL Hard Wait Codes				
X'07'	X'E6'	Channel	Unit or X'00'	IPL input/output error: I/O error on SYSRES I/O error on communication device Equipment malfunction during STORE-CHANNEL-ID
X'C1'	X'E2'	not used	not used	Irrecoverable machine check
X'cc'	X'00'	X'0F'	X'D0'	Error during IPL. IPL canceled (cc = cancel code)
X'F0'	X'C9'	X'F0'	X'F0'	see message 0I00
X'F0'	X'C9'	X'F0'	X'F2'	Requested Supervisor cannot be loaded (see message 0I03)
X'F0'	X'C9'	X'F0'	X'F6'	Device type of SYSRES cannot be identified. The volume label (VOL1) or format-4 record contains invalid information. The pack was not initialized correctly.
X'F0'	X'C9'	X'F0'	X'F7'	see message 0I07
X'F0'	X'C9'	X'F0'	X'F8'	see message 0I08
X'F0'	X'C4'	X'F3'	X'F8'	see message 0D38
X'F0'	X'D1'	X'F5'	X'F0'	Unsopported SYSLOG device see message 0J50
Device Error Recovery Wait Codes. Refer to OP... messages				
X'08' to X'60'	X'C1' or X'C4'	Channel	Unit	Error recovery message
SDAID Soft Wait Codes (identified by EEEE in addr. part of WAIT PSW.				
X'62'	X'C5'	not used	not used	SDAID output device unready. Make printer ready and press EXTERNAL INTERRUPT key.
X'00'	X'00'	X'00'	X'00'	SDAID stop on event. To continue press EXT. INTERRUPT.

HARD AND SOFT WAIT CODES (cont....)

BYTE 0	BYTE 1	BYTE 2	BYTE 3	EXPLANATION
General Hard Wait Codes				
X'00'	X'00'	X'0C'	X'CC'	No recovery possible from CRT errors.
X'00'	X'00'	X'0F'	X'ED'	System error condition (e.g. control block inconsistency) GReg 5 contains the address of the location where the system inconsistency was determined.
X'00'	X'00'	X'0F'	X'F1'	System error detected by the page manager.
X'00'	X'00'	X'0F'	X'F2'	Unused
X'00'	X'00'	X'0F'	X'F3'	Unused
X'00'	X'00'	X'0F'	X'F4'	$$A transient not found (the transient name can be found in ERBLOC).
X'00'	X'00'	X'0F'	X'F5'	TFIX count outside limits.
X'00'	X'00'	X'0F'	X'F6'	I/O error during SLD update.
X'00'	X'00'	X'0F'	X'F7'	No copy blocks available for BTAM appendage I/O request.
X'00'	X'00'	X'0F'	X'F8'	CRT phase not found.
X'00'	X'00'	X'0F'	X'F9'	Paging I/O error.
X'00'	X'00'	X'0F'	X'FA'	Translation specification exception.
X'00'	X'00'	X'0F'	X'FB'	Page fault in Supervisor routine with identifier RID=X'00'.

ERROR BYTES AND INFORMATION IN LOW REAL STORAGE
(in variation to permanent main storage assignment
- see System/370 Reference Summary)

Storage Loc.		
Hex	Dec	
0 - 3	0 - 3	Hard Wait Message Code (MCH,CCH,IPL)
0 - 4	0 - 4	Device error message codes if I/O error and SYSLOG device is also in error.
10 - 13	16 - 19	In a System with ACF/VTAM, the address of the VTAM communications vector table (ATCVT)
14 - 17	20 - 23	Address of Communication Region for active partition (COMREG)
80 - 83	128-131	Address of System Communication Region (SYSCOM)
84 - 85	132-133	External interrupt information
90 - 93	144-147	Address which caused a page fault
94 - 95	148-149	Monitor class number
9C - 9F	156-159	Monitor call address field

APPENDIX D. ASCII CONVERSION TABLES

ASCII to EBCDIC Correspondance 0/0 to 3/9

ASCII				EBCDIC			
Charact.	Col	Row	Bit Pattern	Col	Row	Bit Pattern	Comments
				(Hex)			
NUL	0	0	0000 0000	0	0	0000 0000	
SOH	0	1	0000 0001	0	1	0000 0001	
STX	0	2	0000 0010	0	2	0000 0010	
ETX	0	3	0000 0011	0	3	0000 0011	
EOT	0	4	0000 0100	3	7	0011 0111	
ENQ	0	5	0000 0101	2	D	0010 1101	
ACK	0	6	0000 0110	2	E	0010 1110	
BEL	0	7	0000 0111	2	F	0010 1111	
BS	0	8	0000 1000	1	6	0001 0110	
HT	0	9	0000 1001	0	5	0000 0101	
LF	0	10	0000 1010	2	5	0010 0101	
VT	0	11	0000 1011	0	B	0000 1011	
FF	0	12	0000 1100	0	C	0000 1100	
CR	0	13	0000 1101	0	D	0000 1101	
SO	0	14	0000 1110	0	E	0000 1110	
SI	0	15	0000 1111	0	F	0000 1111	
DLE	1	0	0001 0000	1	0	0001 0000	
DC1	1	1	0001 0001	1	1	0001 0001	
DC2	1	2	0001 0010	1	2	0001 0010	
DC3	1	3	0001 0011	1	3	0001 0011	
DC4	1	4	0001 0100	3	C	0011 1100	
NAK	1	5	0001 0101	3	D	0011 1101	
SYN	1	6	0001 0110	3	2	0011 0010	
ETB	1	7	0001 0111	2	6	0010 0110	
CAN	1	8	0001 1000	1	8	0001 1000	
EM	1	9	0001 1001	1	9	0001 1001	
SUB	1	10	0001 1010	3	F	0011 1111	
ESC	1	11	0001 1011	2	7	0010 0111	
FS	1	12	0001 1100	1	C	0001 1100	
GS	1	13	0001 1101	1	D	0001 1101	
RS	1	14	0001 1110	1	E	0001 1110	
US	1	15	0001 1111	1	F	0001 1111	
SP	2	0	0010 0000	4	0	0100 0000	
|	2	1	0010 0001	4	F	0100 1111	Logical OR
"	2	2	0010 0010	7	F	0111 1111	
#	2	3	0010 0011	7	B	0111 1011	
S	2	4	0010 0100	5	B	0101 1011	
]	2	5	0010 0101	6	C	0110 1100	
&	2	6	0010 0110	5	0	0101 0000	
'	2	7	0010 0111	7	D	0111 1101	
(	2	8	0010 1000	4	D	0100 1101	
)	2	9	0010 1001	5	D	0101 1101	
*	2	10	0010 1010	5	C	0101 1100	
+	2	11	0010 1011	4	E	0100 1110	
,	2	12	0010 1100	6	B	0110 1011	
—	2	13	0010 1101	6	0	0110 0000	Hyphen, Minus
.	2	14	0010 1110	4	B	0100 1011	
/	2	15	0010 1111	6	1	0110 0001	
0	3	0	0011 0000	F	0	1111 0000	
1	3	1	0011 0001	F	1	1111 0001	
2	3	2	0011 0010	F	2	1111 0010	
3	3	3	0011 0011	F	3	1111 0011	
4	3	4	0011 0100	F	4	1111 0100	
5	3	5	0011 0101	F	5	1111 0101	
6	3	6	0011 0110	F	6	1111 0110	
7	3	7	0011 0111	F	7	1111 0111	
8	3	8	0011 1000	F	8	1111 1000	
9	3	9	0011 1001	F	9	1111 1001	

ASCII to EBCDIC Correspondance 3/10 to 7/8

ASCII				EBCDIC			
Charact.	Col	Row	Bit Pattern	Col Row (Hex)		Bit Pattern	Comments
:	3	10	0011 1010	7	A	0111 1010	
;	3	11	0011 1011	5	E	0101 1110	
<	3	12	0011 1100	4	C	0100 1100	
=	3	13	0011 1101	7	E	0111 1110	
>	3	14	0011 1110	6	E	0110 1110	
[	3	15	0011 1111	6	F	0110 1111	
@	4	0	0100 0000	7	C	0111 1100	
A	4	1	0100 0001	C	1	1100 0001	
B	4	2	0100 0010	C	2	1100 0010	
C	4	3	0100 0011	C	3	1100 0011	
D	4	4	0100 0100	C	4	1100 0100	
E	4	5	0100 0101	C	5	1100 0101	
F	4	6	0100 0110	C	6	1100 0110	
G	4	7	0100 0111	C	7	1100 0111	
H	4	8	0100 1000	C	8	1100 1000	
I	4	9	0100 1001	C	9	1100 1001	
J	4	10	0100 1010	D	1	1101 0001	
K	4	11	0100 1011	D	2	1101 0010	
L	4	12	0100 1100	D	3	1101 0011	
M	4	13	0100 1101	D	4	1101 0100	
N	4	14	0100 1110	D	5	1101 0101	
O	4	15	0100 1111	D	6	1101 0110	
P	5	0	0101 0000	D	7	1101 0111	
Q	5	1	0101 0001	D	8	1101 1000	
R	5	2	0101 0010	D	9	1101 1001	
S	5	3	0101 0011	E	2	1110 0010	
T	5	4	0101 0100	E	3	1110 0011	
U	5	5	0101 0101	E	4	1110 0100	
V	5	6	0101 0110	E	5	1110 0101	
W	5	7	0101 0111	E	6	1110 0110	
X	5	8	0101 1000	E	7	1110 0111	
Y	5	9	0101 1001	E	8	1110 1000	
Z	5	10	0101 1010	E	9	1110 1001	
[	5	11	0101 1011	4	A	0100 1010	
\	5	12	0101 1100	E	0	1110 0000	Reverse Slant
]	5	13	0101 1101	5	A	0101 1010	
¬	5	14	0101 1110	5	F	0101 1111	Logical NOT
_	5	15	0101 1111	6	D	0110 1101	Underscore
`	6	0	0110 0000	7	9	0111 1001	Grave Accent
a	6	1	0110 0001	8	1	1000 0001	
b	6	2	0110 0010	8	2	1000 0010	
c	6	3	0110 0011	8	3	1000 0011	
d	6	4	0110 0100	8	4	1000 0100	
e	6	5	0110 0101	8	5	1000 0101	
f	6	6	0110 0110	8	6	1000 0110	
g	6	7	0110 0111	8	7	1000 0111	
h	6	8	0110 1000	8	8	1000 1000	
i	6	9	0110 1001	8	9	1000 1001	
j	6	10	0110 1010	9	1	1001 0001	
k	6	11	0110 1011	9	2	1001 0010	
l	6	12	0110 1100	9	3	1001 0011	
m	6	13	0110 1101	9	4	1001 0100	
n	6	14	0110 1110	9	5	1001 0101	
o	6	15	0110 1111	9	6	1001 0110	
p	7	0	0111 0000	9	7	1001 0111	
q	7	1	0111 0001	9	8	1001 1000	
r	7	2	0111 0010	9	9	1001 1001	
s	7	3	0111 0011	A	2	1010 0010	
t	7	4	0111 0100	A	3	1010 0011	
u	7	5	0111 0101	A	4	1010 0100	
v	7	6	0111 0110	A	5	1010 0101	
w	7	7	0111 0111	A	6	1010 0110	
x	7	8	0111 1000	A	7	1010 0111	

ASCII to EBCDIC Correspondance 7/9 to 7/15

ASCII			EBCDIC			
Charact.	Col	Row	Bit Pattern	Col Row (Hex)	Bit Pattern	Comments
y	7	9	0111 1001	A 8	1010 1000	
z	7	10	0111 1010	A 9	1010 1001	
}	7	11	0111 1011	C 0	1100 0000	
|	7	12	0111 1100	6 A	0110 1010	Vertical Line
{	7	13	0111 1101	D 0	1101 1101	
°	7	14	0111 1110	A 1	1010 1010	Tilde
DEL	7	15	0111 1111	0 7	0000 0111	

EBCDIC to ASCII Correspondance 0/0 to 5/D

EBCDIC				ASCII			
Charact.	Col (Hex)	Row	Bit Pattern	Col	Row	Bit Pattern	Comments
NUL	0	0	0000 0000	0	0	0000 0000	
SOH	0	1	0000 0001	0	1	0000 0001	
STX	0	2	0000 0010	0	2	0000 0010	
ETX	0	3	0000 0011	0	3	0000 0011	
HT	0	5	0000 0101	0	9	0000 1001	
DEL	0	7	0000 0111	7	15	0111 1111	
VT	0	B	0000 1011	0	11	0000 1011	
FF	0	C	0000 1100	0	12	0000 1100	
CR	0	D	0000 1101	0	13	0000 1101	
SO	0	E	0000 1110	0	14	0000 1110	
SI	0	F	0000 1111	0	15	0000 1111	
DLE	1	0	0001 0000	1	0	0001 0000	
DC1	1	1	0001 0001	1	1	0001 0001	
DC2	1	2	0001 0010	1	2	0001 0010	
DC3	1	3	0001 0011	1	3	0001 0011	
BS	1	6	0001 0110	0	8	0000 1000	
CAN	1	8	0001 1000	1	8	0001 1000	
EM	1	9	0001 1001	1	9	0001 1001	
FS	1	C	0001 1100	1	12	0001 1100	
GS	1	D	0001 1101	1	13	0001 1101	
RS	1	E	0001 1110	1	14	0001 1110	
US	1	F	0001 1111	1	15	0001 1111	
LF	2	5	0010 0101	0	10	0000 1010	
ETB	2	6	0010 0110	1	7	0001 0111	
ESC	2	7	0010 0111	1	11	0001 1011	
ENQ	2	D	0010 1101	0	5	0000 0101	
ACK	2	E	0010 1110	0	6	0000 0110	
BEL	2	F	0010 1111	0	7	0000 0111	
SYN	3	2	0011 0010	1	6	0001 0110	
EOT	3	7	0011 0111	0	4	0000 0100	
DC4	3	C	0011 1100	1	4	0001 0100	
NAK	3	D	0011 1101	1	5	0001 0101	
SUB	3	F	0011 1111	1	10	0001 1010	
SP	4	0	0100 0000	2	0	0010 0000	
[	4	A	0100 1010	5	11	0101 1011	
.	4	B	0100 1011	2	14	0010 1110	
<	4	C	0100 1100	3	12	0011 1100	
(	4	D	0100 1101	2	8	0010 1000	
+	4	E	0100 1110	2	11	0010 1011	
|	4	F	0100 1111	2	1	0010 0001	Logical OR
&	5	0	0101 0000	2	6	0010 0110	
]	5	A	0101 1010	5	13	0101 1101	
S	5	B	0101 1011	2	4	0010 0100	
*	5	C	0101 1100	2	10	0010 1010	
)	5	D	0101 1101	2	9	0010 1001	

EBCDIC to ASCII Correspondance 5/E to D/9

EBCDIC				ASCII			
Charact.	Col (Hex)	Row	Bit Pattern	Col	Row	Bit Pattern	Comments
;	5	E	0101 1110	3	11	0011 1011	
¬	5	F	0101 1111	5	14	0101 1110	Logical NOT
-	6	0	0110 0000	2	13	0010 1101	Hyphen, Minus
/	6	1	0110 0001	2	15	0010 1111	
|	6	A	0110 1010	7	12	0111 1100	Vertical Line
,	6	B	0110 1011	2	12	0010 1100	
]	6	C	0110 1100	2	5	0010 0101	
_	6	D	0110 1101	5	15	0101 1111	Underscore
>	6	E	0110 1110	3	14	0011 1110	
[	6	F	0110 1111	3	15	0011 1111	
`	7	9	0111 1001	6	0	0110 0000	Grave Accent
:	7	A	0111 1010	3	10	0011 1010	
#	7	B	0111 1011	2	3	0010 0011	
@	7	C	0111 1100	4	0	0100 0000	
'	7	D	0111 1101	2	7	0010 0111	
=	7	E	0111 1110	3	13	0011 1101	
"	7	F	0111 1111	2	2	0010 0010	
a	8	1	1000 0001	6	1	0110 0001	
b	8	2	1000 0010	6	2	0110 0010	
c	8	3	1000 0011	6	3	0110 0011	
d	8	4	1000 0100	6	4	0110 0100	
e	8	5	1000 0101	6	5	0110 0101	
f	8	6	1000 0110	6	6	0110 0110	
g	8	7	1000 0111	6	7	0110 0111	
h	8	8	1000 1000	6	8	0110 1000	
i	8	9	1000 1001	6	9	0110 1001	
j	9	1	1001 0001	6	10	0110 1010	
k	9	2	1001 0010	6	11	0110 1011	
l	9	3	1001 0011	6	12	0110 1100	
m	9	4	1001 0100	6	13	0110 1101	
n	9	5	1001 0101	6	14	0110 1110	
o	9	6	1001 0110	6	15	0110 1111	
p	9	7	1001 0111	7	0	0111 0000	
q	9	8	1001 1000	7	1	0111 0001	
r	9	9	1001 1001	7	2	0111 0010	
°	A	1	1010 1010	7	14	0111 1110	Tilde
s	A	2	1010 0010	7	3	0111 0011	
t	A	3	1010 0011	7	4	0111 0100	
u	A	4	1010 0100	7	5	0111 0101	
v	A	5	1010 0101	7	6	0111 0110	
w	A	6	1010 0110	7	7	0111 0111	
x	A	7	1010 0111	7	8	0111 1000	
y	A	8	1010 1000	7	9	0111 1001	
z	A	9	1010 1001	7	10	0111 1010	
{	C	0	1100 0000	7	11	0111 1011	
A	C	1	1100 0001	4	1	0100 0001	
B	C	2	1100 0010	4	2	0100 0010	
C	C	3	1100 0011	4	3	0100 0011	
D	C	4	1100 0100	4	4	0100 0100	
E	C	5	1100 0101	4	5	0100 0101	
F	C	6	1100 0110	4	6	0100 0110	
G	C	7	1100 0111	4	7	0100 0111	
H	C	8	1100 1000	4	8	0100 1000	
I	C	9	1100 1001	4	9	0100 1001	
}	D	0	1101 1101	7	13	0111 1101	
J	D	1	1101 0001	4	10	0100 1010	
K	D	2	1101 0010	4	11	0100 1011	
L	D	3	1101 0011	4	12	0100 1100	
M	D	4	1101 0100	4	13	0100 1101	
N	D	5	1101 0101	4	14	0100 1110	
O	D	6	1101 0110	4	15	0100 1111	
P	D	7	1101 0111	5	0	0101 0000	
Q	D	8	1101 1000	5	1	0101 0001	
R	D	9	1101 1001	5	2	0101 0010	

EBCDIC to ASCII Correspondance E/0 to F/9

EBCDIC			ASCII		
Charact.	Col Row (Hex)	Bit Pattern	Col Row	Bit Pattern	Comments
\	E 0	1110 0000	5 12	0101 1100	Reverse Slant
S	E 2	1110 0010	5 3	0101 0011	
T	E 3	1110 0011	5 4	0101 0100	
U	E 4	1110 0100	5 5	0101 0101	
V	E 5	1110 0101	5 6	0101 0110	
W	E 6	1110 0110	5 7	0101 0111	
X	E 7	1110 0111	5 8	0101 1000	
Y	E 8	1110 1000	5 9	0101 1001	
Z	E 9	1110 1001	5 10	0101 1010	
0	F 0	1111 0000	3 0	0011 0000	
1	F 1	1111 0001	3 1	0011 0001	
2	F 2	1111 0010	3 2	0011 0010	
3	F 3	1111 0011	3 3	0011 0011	
4	F 4	1111 0100	3 4	0011 0100	
5	F 5	1111 0101	3 5	0011 0101	
6	F 6	1111 0110	3 6	0011 0110	
7	F 7	1111 0111	3 7	0011 0111	
8	F 8	1111 1000	3 8	0011 1000	
9	F 9	1111 1001	3 9	0011 1001	

INDEX

VSE/Advanced Functions
Handbook
Order No. LY33-9121-0

READER'S COMMENT FORM

This sheet is for comments and suggestions about this manual. We would appreciate *your* views, favorable or unfavorable, in order to aid us in improving *this* publication. This form will be sent directly to the author's department. Please include your name and address if you wish a reply. Contact your IBM branch office for answers to technical questions about the system or when requesting additional publications. Thank you.

Your comments* and suggestions:

***We would especially appreciate your comments on any of the following topics:**

Clarity of the text
Organization of the text
Accuracy
Cross-references
Index
Tables
Illustrations
Examples
Appearance
Printing
Paper
Binding

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 6R1BP
180 Kost Road
Mechanicsburg, PA 17055

IBM®

LY33-9121-00

LY33-9121-0

(File No. S370/4300-40)
Printed in U.S.A.