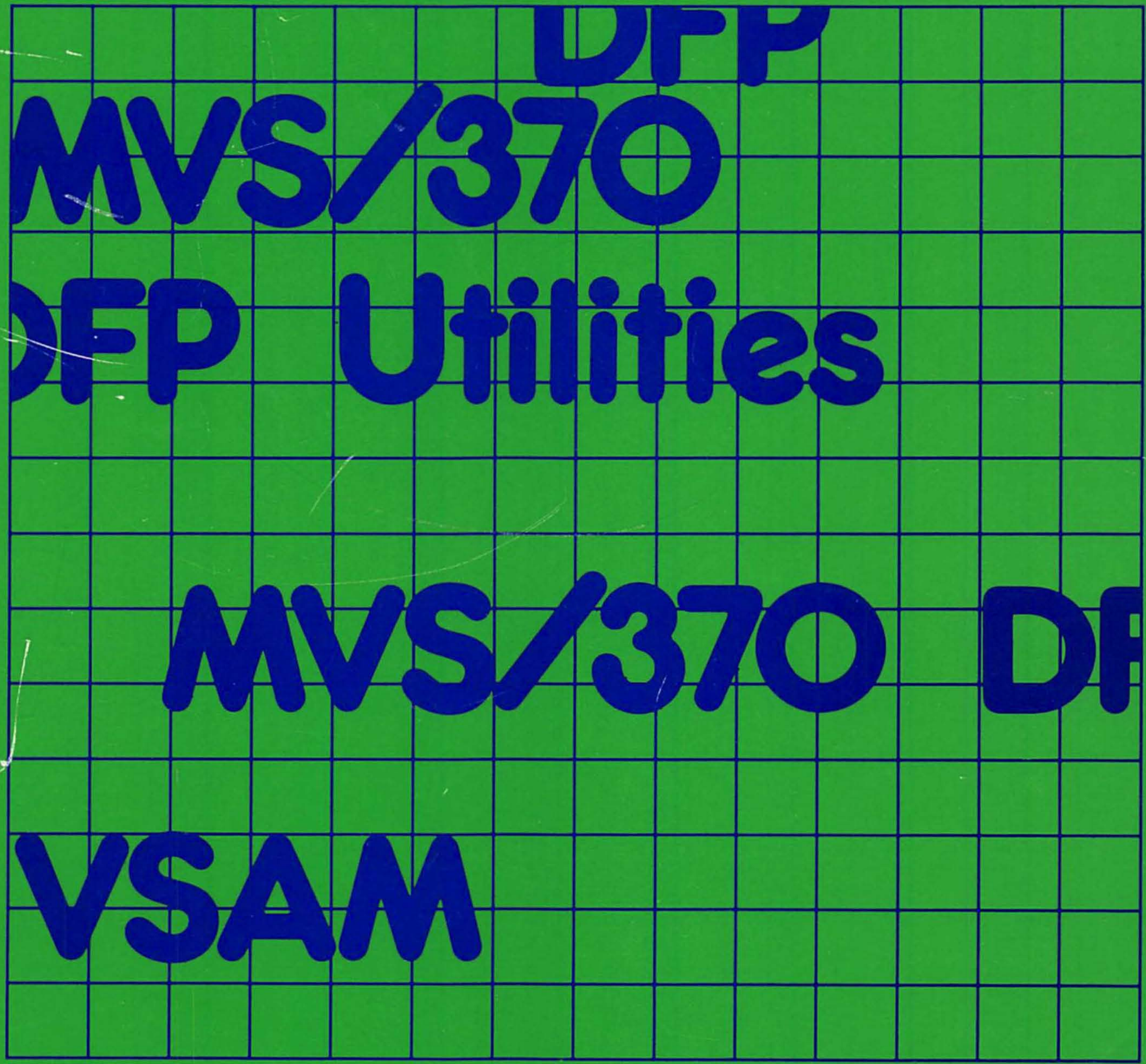
IBM

MVS/370 Access Method Services Logic Volume 2

Program Product



Order No. LY26-3930-0
File No. S370-30

Contains Restricted Materials of IBM
Licensed Materials—Property of IBM

Data Facility Product 5665-295
Release 1.1

MVS/370
Access Method Services Logic
Volume 2

Data Facility Product 5665-295
Release 1.1

LY26-3930-0

Second Edition (December 1985)

This is a major revision of, and makes obsolete, LY26-3912-0 and its technical newsletter, LN26-8088.

This edition applies to Release 1.1 of MVS/370 Data Facility Product, Program Product 5665-295, and to any subsequent releases until otherwise indicated in new editions or technical newsletters.

The changes for this edition are summarized under "Summary of Amendments" following the preface. Specific changes are indicated by a vertical bar to the left of the change. These bars will be deleted at any subsequent republication of the page affected. Editorial changes that have no technical significance are not noted.

Changes are made periodically to this publication; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 and 4300 Processors 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 which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

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

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, P.O. Box 50020, Programming Publishing, San Jose, California, U.S.A. 95150. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

This book describes the internal logic of Data Facility Product Access Method Services routines and provides corresponding diagnostic information. The information presented is directed to support personnel and development programmers who require an indepth knowledge of a program's design, organization, and data areas. It is not required for effective use of Access Method Services.

Organization

This publication contains the following sections:

- "Data Areas," describes the control blocks and other data areas that are internal to this processor.
- "Diagnostic Aids," shows how to analyze a dump of the processor and how to find specific modules and data areas.
- Appendix A, "Portable Data Sets Created by the EXPORT Command," describes the portable data sets created by the EXPORT command.
- Appendix B, "Portable Data Sets Created by the EXPORTRA Command," describes the portable data sets created by the EXPORTRA command.
- Appendix C, "Enciphered Data Sets Created by the REPRO Command," shows the header format required to decipher an enciphered data set created by the REPRO command.

The accompanying publication, *Access Method Services Logic, Volume 1*, contains the following sections:

- "Introduction," describes the design philosophy of this processor, and defines terms used later in the book.
- "Method of Operation," describes how the program works. Emphasis is on the flow of data and the technology that is used rather than on the organization of modules.
- "Program Organization," shows how the processor is packaged into load modules. Relationships between the Access Method Services processor and the operating system are given.

- "Microfiche Directory," relates the information in this book to the listings found on microfiche.

An index is also included.

Prerequisite Knowledge

To use this publication effectively, you should be familiar with general programming techniques, VSAM concepts and use, and TSO concepts and use.

Required Publications

- *MVS/370 Access Method Services Logic, Volume 1*, LY26-3912, which complements this volume.

- Either *MVS/370 Integrated Catalog Administration: Access Method Services Reference*, GC26-4051, or *MVS/370 VSAM Catalog Administration: Access Method Services Reference*, GC26-4059, which describes the commands of Access Method Services and how they are used.

- *MVS/370 Catalog Administration Guide*, GC26-4053, which describes the services a user may want to use, and how to perform these services.

- *MVS/370 VSAM Administration Guide*, GC26-4066, which describes the use of VSAM.

- *OS/VS2 TSO Terminal User's Guide*, GC28-0645, which describes how to use TSO commands.

Related Publications

Within the text, references are made to the publications listed in the table below.

Short Title	Publication Title	Order Number
Access Method Services Logic Volume 1	*MVS/370 Access Method Services Logic Volume 1*	LY26-3912
Access Method Services Reference	*MVS/370 Integrated Catalog Administration: Access Method Services Reference* *MVS/370 VSAM Catalog Administration: Access Method Services Reference*	GC26-4051 GC26-4059
Catalog Administration Guide	*MVS/370 Catalog Administration Guide*	GC26-4053
Catalog Diagnosis Reference	*MVS/370 Catalog Diagnosis Reference*	LY27-9507
Debugging Handbook	*OS/VS2 System Programming Library: Debugging Handbook,* Volume 1	GC28-1047
Guide to PL/S II	*Guide to PL/S II*	GC28-6794
Job Management	*OS/VS2 System Programming: Job Management*	GC28-0627
Mass Storage Control Table Create Logic	*OS/VS Mass Storage Control Table Create Logic*	SY35-0016
Mass Storage System (MSS) Services Logic	*OS/VS Mass Storage System (MSS) Services Logic*	SY35-0015
System Messages	*MVS/370 Message Library: System Messages*, Volumes 1 and 2	GC28-1374 GC28-1375
TSO Command Language Reference	*OS/VS2 TSO Command Language Reference*	GC28-0646
TSO Guide to Writing a TMP or CP	*OS/VS2 TSO Guide to Writing a Terminal Monitor Program or Command Processor*	GC28-0648

Short Title	Publication Title	Order Number
TSO Terminal Monitor Program and Service Routines Logic	*OS/VS2 TSO Terminal Monitor Program and Service Routines Logic*	SY28-0650
TSO Terminal User's Guide	*OS/VS2 TSO Terminal User's Guide*	Gc28-0645
VSAM Administration Guide	*MVS/370 VSAM Administration Guide*	GC26-4066
VSAM Logic	*MVS/370 VSAM Logic*	LY26-3928

Summary of Amendments

Release 1.1 Library Update, December 1985

Service Changes

Access Method Services Logic has been divided into two volumes:

- Volume 1 covers "Introduction," "Method of Operation," "Program Organization," and "Microfiche Directory."
- Volume 2 covers "Data Areas," "Diagnostic Aids," and all the appendixes.

All MVS/370 titles referred to in this publication have been changed to their corresponding MVS/XA titles. Order numbers of the MVS/370 books remain the same.

Information has been added to reflect technical service changes.

Enhancements

- The following messages have been added under "Processor Messages" in "Diagnostic Aids" section.
  - IDC0896I
  - IDC2895I
  - IDC3558I
  - IDC3897I
  - IDC31376I
  - IDC31377I
  - IDC31442I
- The IOCSEX (Input/Output Communication Structure Extension) and OPNAGL (Open Argument List) data areas have been updated.
- A new access method services function, EXAMINE, has been added, incorporating three function support routines: IDCXM01, IDCXM02, and IDCXM03. The description of EXAMINE logic affects the following sections of this manual: "Program Organization," "Data Areas," and "Diagnostic Aids."

Release 1.1, October 1983

New Programming Support

A new message (IDC01551I) indicating the status of caching activity has been added.

Contents

Figures

Data Areas

The data areas in this chapter are described in four columns, which are interpreted as follows:

Offset: The numeric address of the field relative to the beginning of the area. The first number is the offset in decimal, followed (in parentheses) by the hexadecimal equivalent.

Bytes and Bit Pattern: The size (number of bytes) of the field and its alignment relative to the fullword boundary. A *v* indicates variable length.

Examples:

4 A 4-byte field beginning on a word boundary.

. . 3 A 3-byte field beginning on a halfword boundary and running into the next word.

This column also shows the bit patterns of a byte when they are significant (as in a flag byte). When the column is used to show the state of the bits (0 or 1) in a flag byte, it is shown as follows:

.... The 8 bit positions (0 through 7) in a byte. For ease of scanning, the high-order (leftmost) 4 bits are separated from the low-order 4 bits.

x... .... A reference to bit 0.

1... .... Bit 0 is on.

0... .... Bit 0 is off.

.... ..xx A reference to bits 6 and 7.

Bit settings that are significant are shown and described. Bit settings that are not shown are considered to be reserved and set to zero.

Field Name: A name that identifies the field and appears in the assembly listings. A subfield or value name is indented from the field's name. An * indicates the field is not named.

Description: Content, Meaning, Use: A description of the use of the field.

ALLAGL

Allocation Argument List

The allocation argument list is passed whenever a UALLOC or UDEALLOC macro is issued. It contains information needed to dynamically allocate or deallocate data sets or to mount or demount volumes.

Created by	Modified by	Used by	Size
All routines	IDCSA02	IDCSA02	36

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	ALLDSN	For data set allocation or deallocation, this contains the address of a 44-byte data set name, left-justified and padded with blanks. This field is not used for volume mounting or demounting.
4(4)	8	ALLDDN	After successful data set allocation or volume mounting, this field contains the ddname for the data set or volume. The ddname is left-justified and padded with blanks. If several volumes were mounted, the ddname is the first of several concatenated DD statements. For data set deallocation and volume demounting, and if ALLDSN is zero, this field contains the ddname describing the data set or volume. The ddname is left-justified and padded with blanks. It may be the first of concatenated DD statements.
12(C)	4	ALLULP	For volume mounting, this field contains the address of a table.
The table contains:			
	2	*	Number of entries in the following array.
Each entry consists of:			
	2	*	Number of nonblank characters in the following field.
	8	*	Unit name or device type left-justified and padded with blanks.
16(10)	4	ALLVLP	For volume mounting, this field contains the address of a table.

The table contains:

	2	*	Number of entries in the following array:

Each entry consists of:

	2	*	Number of nonblank characters in the following field.
	6	*	Volume serial number left-justified and padded with blanks. Volume serials in this array are matched one to one with the units in the array in ALLULP. The last unit is matched with all the unmatched volume serials. This field is used only for volume mounting.
20(14)	4	ALLPWD	For allocation of password protected data sets, this field contains the address of an 8-byte password, left-justified, and padded with blanks. If ALLPWD contains zero, no password is supplied.
24(18)	1	ALLSTS	For data set allocation, this field means the following data set status:
	.... 1...	ALLSTSSR	Status of SHR.
	.... .1..	ALLSTSNW	Status of NEW.
	.... ..1.	ALLSTSMD	Status of MOD.
	1	ALLSTSOD	Status of OLD.
25(19)	. 1	ALLDSP	For data set allocation, this field indicates the current data set disposition. For data set deallocation, this field indicates a disposition that overrides any other disposition.
	.... 1...	ALLDSPKP	Disposition of KEEP.
	.... .1..	ALLDSPDE	Disposition of DELETE.
	.... ..1.	ALLDSPCG	Disposition of CATLG.
	1	ALLDSPUN	Disposition of UNCATLG.
	0		No overriding disposition.
26(1A)	. . 2	ALLORG	For data set allocation, the UALLOC puts the data set organization in this field:
	. . 1		
	1...	ALLORGIS	Indexed sequential.
	.1..	ALLORGPS	Physical sequential.
	..1.	ALLORGDA	Direct access.
	...1	ALLORGCX	Telecommunications line group.
	.... 1...	ALLORGCQ	Direct-access message queue.
	.... .1..	ALLORGMQ	Data control block for message transfer.
	.... ..1.	ALLORGPO	Partitioned organization.
	1	ALLORGUN	Unmovable data set.

27(1B)	... 1	ALLORG (cont.)	
	1...	ALLORGGS	Graphic data control block.
	.... 1...	ALLORGVS	VSAM.
	0		Dynamic allocation cannot determine the data set organization.
28(1C)	1	ALLOPT	Indicates the type of request either data set allocation or volume mounting, and allocation option for private mounting and deallocate option to unallocate.
	1...	ALLOPTVL	Request for volume mounting.
	.1..	ALLOPTDS	Request for data set allocation.
	..1.	ALLOPTPV	Request for private volume.
	...1	ALLOPTUN	Request for unallocate option during deallocation.
29(1D)	.1	ALLVLCNT	Volume count.
30(1E)	.. 1	ALLUNCNT	Unit count.
31(1F)	1		Reserved.
32(20)	4	ALLMEN	Pointer to partitioned data set member name.

BUFS

Buffer Pool Control Block

The buffer pool control block is used by EXPORTRA to control I/O buffers. It is passed from IDCRC01 through field management (IDCRC04) to IDCRC03.

Created by	Modified by	Used by	Size
IDCRC01	IDCRC03	IDCRC03	28

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	BUFPOOL	Address of first buffer.
4(4)	4	BUFPL	Address of chain of buffers.
8(8)	4	BUFIOCS	Address of the IOCSTR.
12(C)	4	BUFGDT	Address of the GDT.
16(10)	4	BUFCTT	Address of the CTT.
20(14)	4	BUFWKARA	Address of the work area.
24(18)	2	BUFSIZE	Size of buffer pool.
26(1A)	.2	BUFSWS	Indicator flags.
	1...	BUFORMAT	1=Buffer pool formatted 0=Buffer pool not formatted.
	.xxx xxxx	*	Reserved.
	xxxx xxxx	*	Reserved.

CIRAGL

Catalog Interface Argument List

The catalog interface argument list is required each time a UCIR macro is issued. The CIRAGL contains information about the qualifiers for which data set names are returned.

Created by	Modified by	Used by	Size
All routines	IDCSA02	IDCSA02	32

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	CIRTYP	Type of UCIR request:
	1...		"One" qualifier function. This function returns data set names whose qualifiers match the qualifiers provided in CIRAGL and have only one qualifier between the header and trailer qualifiers supplied in CIRAGL.
	.1..		"All" qualifier function. This function returns all data set names whose qualifiers match the qualifiers provided in CIRAGL.
1(1)	3	*	Reserved.
4(4)	2	CIRHLN	Contains the length of the header qualifiers.
6(6)	2	CIRTLN	Contains the length of the trailer qualifiers. If no trailer qualifiers are supplied, this field must contain zero.
8(8)	4	CIRHDR	Address of the header qualifier. The header qualifier must end with a period.
12(C)	4	CIRTLR	Address of trailer qualifier. The trailer qualifier must start with a period.
16(10)	4	CIRWKP	Address of a fullword where the UCIR macro puts the address of the work area containing the qualified data set names.
20(14)	4	CIRCAT	If data set names are to be found only in one catalog, this field contains the address of a 44-byte catalog name. If data set names are to be found in more than one catalog, this field contains zero.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
24(18)	4	CIRPWD	If the catalog addressed in CIRCAT is password protected, this field may contain the address of a 8-byte password. If no password is supplied for a password protected catalog, the operator will be prompted for it.
28(1C)	4	CIRPID	Contains a 4-byte UGPOOL identifier that is used to obtain storage for the returned work area. The caller must free this work area.

CKAGL

Check UCB Argument List

The CKAGL is passed when a UMSSUNIT is issued for a CHECK request. It defines a request to interrogate a specified UCB for information.

Created by	Modified by	Used by	Size
Calling routine	IDCSA06	IDCSA06	29

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	CKHEAD	Set with 'CHECK' by IDCSA06. Set with 'CHECKERR' if an invalid option is set in CKFLAGS.
8(8)	4	CKUCBPTR	Address of an area containing the UCB address or zeros if the UCB address is unknown. If the UCB address is zeros, on return the UCB address will be put in the area.
12(C)	8	CKDDNAME	DDNAME of the volume.
20(14)	4	CKDATYPE	Address of the 4-byte area to be posted with the characters 'VIRT' or 'REAL' if the field CKRETTYP is set. The area is set only when the UCB is a 3330.
24(18)	4	CKLABELP	Address of the 6-byte volume serial number if CKDYALOC is set.
28(1C)	1	CKFLAGS	Flags indicating the type of test.
	1...	CKTESTVT	Indicates that the request is for a 3330 virtual UCB. This field and CKRETTYP are mutually exclusive.
	.1..	CKRETTYP	Indicates that the test is for a 3330 Model I (real) or 3330V (virtual) UCB and to return the type to the area addressed by CKDATYPE.
	..1.	CKDMTABL	Indicates that the test is for a real or virtual 3330 UCB that is demountable, using the CKDDNAME to locate the UCB pointer.

CLST

Compare List

The CLST contains names of all data sets to be compared by DIAGNOSE.

Created by	Modified by	Used by	Size
IDCDA02	IDCDA01 IDCDA02 IDCDA03	IDCDA01 IDCDA02 IDCDA03	Variable

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	CLSTID	ID='CLST'.
4(4)	1	*	Reserved.
5(5)	3	CLSTLEN	Length for FREEMAIN.
8(8)	4	CLSTAVAL	Address of next CLST entry.
12(C)	4	CLSTLAST	Address of last CLST entry.
16(10)	4	CLSTNEXT	Address of next CLST block.
20(14)	var.	CLSTNAMS	Entries in CLSTY.

Each CLSTNAMS entry contains the following:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	1	CNFLAG1	Flags field.
	1...	CNVALID	Entry is valid.
	.1..	CNUALLOC	UALLOC was done, do a UDEALLOC.
	..1.	CNOPEN	UOPEN was done, need a UCLOSE.
	...1	CNOERR	UOPEN error, skip further OPEN.
	.... xxx.	*	Reserved.
	1	CNFOUND	This entry found by search.
	1	*	Reserved.
	6	CNVOLSER	Volser of entry.
	44	CNDSN	Data set name of entry.
	8	CNDDN	DD name for entry.
	4	CNIOCAD	IOCS for this entry.

Command Descriptor

There is a command descriptor for each verb supported by this processor. The command descriptor is a load module that contains directions for parsing the command, performing semantic checking, and building an FDT from the commands. The name of the load module for each verb is found in a directory, which is itself a load module named IDCRILT.

The name of each load module and the corresponding verb, as supplied by IBM, is as follows:

VSAM		CHECKPOINT/RESTART	
IDCCDAL	ALTER	IDCCDCK	CHKLIST
IDCCDBI	BLDINDEX		
IDCCDCC	CNVTCAT		
IDCCDDA	DIAGNOSE		
IDCCDDE	DEFINE		
IDCCDDL	DELETE		
IDCCDLC	LISTCAT		
IDCCDLR	LISTCRA		
IDCCDMP	IMPORT		
IDCCDPM	PARM		
IDCCDPR	PRINT		
IDCCDRC	EXPORTRA		
IDCCDRM	IMPORTRA		
IDCCDRP	REPRO		
IDCCDRS	RESETCAT		
IDCCDVY	VERIFY		
IDCCDXP	EXPORT		

Each command descriptor consists of several variable length data areas. The data areas are divided into two groups. The first group is used to build the function data table, FDT. If access method services is invoked with a batched job, IDCRI01 also uses the first group to parse commands. The second group is used to parse a command if access method services is invoked interactively with TSO.

The first group is described in the following table and consists of two main data areas. The first data area, verb data area, names the FSR load module to use for this command. Appendages to the verb data area define positional parameters, default parameters, groups of needed parameters, and groups of incompatible parameters. The second data area, parameter data area, describes parameters the command may have. If several parameters have the same syntax attributes, one parameter data area can describe the parameters. Appendages to each parameter data area define constants and data for the parameter(s). The command descriptor assigns an identification number, ID, to each parameter the command may have. Both reader/interpreters use the ID numbers to reference the parameter data area that describes a parameter.

The second group of data areas in a command descriptor consists of parameter control lists, PCL. The PCLs are arranged in the command descriptor as follows: First is the PCL that describes all the nonrepeated parameters. Second is one PCL for each set of repeated parameters. Third is one PCL for each set of parameters that may be prompted for. IDCRI04 passes the PCLs to TSO so TSO can parse the command. TSO returns a parameter descriptor list, PDL. Each parsed parameter is described by its own section of the PDL called the parameter descriptor entry, PDE. For a description of the PCL, PDL, and PDE, see *TSO Guide to Writing a Terminal Monitor Program or a Command Processor.*

Created by	Modified by	Used by	Size
IBM-supplied	None	IDCRI01 IDCRI04	Variable

Verb Data Area

A command descriptor always begins with the verb data area. This data area names the FSR for this command, gives the total number of parameters, and provides offsets to other data areas in the command descriptor.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	DESCID	Descriptor identification, contains the last four letters of the command descriptor module name. For example, 'CDAL' for the alter command descriptor, IDCCDAL.
4(4)	2	PCLDSPL1	Number of bytes from the beginning of the verb data area to the first parameter control list.
6(6)	..2	VDATALEN	Number of halfwords in verb data area (used to compute the address of the first parameter data area).
6(6)	2	PARMCNT	Number of parameter data areas in this command descriptor.
10(A)	..2	MAXID	Largest parameter ID number that is used in this command descriptor.
12(C)	8	LOADNAME	Load module name of FSR that processes this command.
20(14)	1	POSDSPL	Number of halfwords from the beginning of the verb data area to positional parameter appendage of the verb data area.
21(15)	.1	DGRPDSPL	Number of halfwords from the beginning of the verb data area to default parameter appendage of the verb data area.
22(16)	..1	VNGRPDSP	Number of halfwords from the beginning of the verb data area to needed parameters appendage of the verb data area.
23(17)	...1	NTGRPDSP	Number of halfwords from the beginning of the verb data area to incompatible parameters appendage of the verb data area.

Positional Parameter Appendage

This appendage contains the parameter ID number of each positional parameter that is not a subparameter of other parameters. This appendage may follow the verb data area or any verb data area appendage.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	VPOSCNT	Number, *n*, of ID numbers that follow:
2(2)	2x*n*	VPOSID*n*	List of ID numbers for positional parameters.

Default Parameter Appendage

This appendage contains the parameter ID number of each default parameter. The parameter IDs are grouped into arrays. The first parameter in each array is the default if none of the parameters in that array is supplied in the command. This appendage may follow the verb data area or any verb data area appendage.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	DGRPTOT	Number of arrays that follow.
Each array contains:			
	2	DGRPCNT	Number, *n*, of ID numbers that follow:
	2x*n*	DGRPID*n*	List of ID numbers.

Needed Parameters Appendage

This appendage contains the parameter ID number of any necessary parameter that is not a subparameter of another parameter. The parameter IDs are grouped into arrays. At least one of the parameters in each array must be supplied through the command. This appendage may follow the verb data area or any verb data area appendage.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	VNGRPTOT	Number of arrays that follow:
Each array contains:			
	2	VNGRPCNT	Number, *n*, of ID numbers that follow:
	2x*n*	VNGRPID*n*	List of ID numbers.

Incompatible Parameters Appendage

This appendage contains the parameter ID numbers for each parameter in groups of incompatible parameters. The parameter IDs are grouped into arrays. Only one parameter in each array may be supplied through the command.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	NTGRPTOT	Number of arrays that follow:
Each array contains:			
	2	NTGRPCNT	Number, *n*, of ID numbers that follow:
	2x*n*	NTGRPID*n*	List of ID numbers.

Parameter Data Area

The parameter data area follows the verb data area and describes the syntax and subparameters of a parameter. Usually there is one parameter data area for each parameter. However, one parameter data area can describe several parameters if the parameters have the same syntax and data.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	PDEFLEN	Number of halfwords in this parameter data area including appendages.
1(1)	3	OCCURNUM	Number of times this parameter can be repeated in the command.
4(4)	1	IDDSPL	Number of halfwords from the beginning of this parameter data area, to the ID appendage.
5(5)	1	KWDDSPL	Number of halfwords from the beginning of this parameter data area to the keyword appendage.
6(6)	1	NOTDSPL	Number of halfwords from the beginning of this parameter data area to the conflicting parameters appendage.
7(7)	1	NGRPDSPL	Number of halfwords from the beginning of this parameter data area to the necessary parameters appendage.
8(8)	1	PDEDSPL	Number of halfwords from the beginning of this parameter data area to the prompt appendage.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
9(9)	1	KWDGRPID	Identification number of a TSO keyword needed when this parameter is part of a group of mutually exclusive parameters.
10(A)	1	*	Reserved.
11(B)	1	FLAGS	Flags:
	1...	SCLRDATA	Indicates the user supplies data with this parameter.
	.1..	LEVEL1	Indicates this parameter is not a subparameter.
	..1.	REPEATED	Indicates the user may repeat the subparameters of this parameter.
	...1	SCALAR	Indicates the user supplies a single constant with this parameter.
	.... 1...	LIST	Indicates the user may supply several "like" constants with this parameter.
	.... .1..	DEFAULT	Indicates this parameter has a default value.
	.... ..1.	SUBLIST	Indicates this parameter has subparameters.
	1	PLUS256	Indicates that the REPMAX value is REPMAX plus 256.

No Constant Appendage

This appendage follows the above section if the parameter has subparameters. In other words, if SUBLIST=1, this appendage immediately follows the FLAGS field described above.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
12(C)	2	PCLDSPL2	Number of bytes from the beginning of this command descriptor to a PCL describing this parameter's subparameters.
14(E)	1	SUBDSPL	Number of halfwords from the beginning of this parameter data area to the subparameter appendage.
15(F)	1	REPMAX	Maximum times this parameter's subparameters may be repeated in the command.

Constant Appendage

This appendage follows the basic parameter data area if the parameter has constants. In other words, if SCLRDATA=1, this appendage immediately follows the FLAGS field described above.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
12(C)	4	HIVALUE	The greatest value a number constant may have.
16(10)	4	LOWVALUE	The least value a number constant may have.
20(14)	1	MAXLNGTH	The maximum length of the constant after any conversion.
21(15)	1	LISTMAX	Maximum number of times this constant may be repeated in a list of subparameters.
22(16)	1	*	Reserved.
23(17)	1	CFLAG	Flags:
	1...	NUMBER	Indicates the constant is a number.
	.1..	ANYSTRNG	Indicates the constant is a character string.
	..1.	DSNAM	Indicates the constant is a data set name.
	...1	GENERIC	Indicates the constant is a generic data set name.
	.... 1...	VOLID	Indicates a volume serial number may replace a data set name.
	.... .1..	USERID	Indicates a prefix of the TSO user's identification must be added to the data set name if access method services is invoked interactively with TSO.
	.... ..1.	PWORDOPT	Indicates the character string or data set name may be followed by a password.
	x	*	Reserved.

Default Data Appendage

This appendage follows the constant appendage if the parameter data has a default constant. In other words, if DEFAULT=1, this appendage immediately follows the CFLAGS field described above.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
24(18)	1	DEFLTLEN	Length of following field.
25(19)	*v*	DEFLTVAL	Default constant as it would appear in the command.

ID Appendage

This appendage contains the offset from the beginning of the primary parameter data list, PDL, to the parameter data entry, PDE, for each parameter this parameter data area describes. This appendage may follow any other parameter data appendage.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	IDCOUNT	Number of sets of two fields that follow. There is a set of fields for each parameter.
Each set contains:			
	2	IDNUM	Parameter ID number.
	2	PDEOFST1	Offset from the beginning of the primary PDL to the PDE for this parameter.

Keyword Appendage

This appendage contains every keyword for each parameter this parameter data area describes. This appendage may follow any other parameter data appendage.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	KWDCOUNT	Number of sets of fields that follow. There is a set of two fields for each keyword.
Each set contains:			
0(0)	1	KWDLEN	Length of the following keyword.
1(1)	*v*	KWDITEM	Keyword.

Conflicting Parameters Appendage

This appendage contains the parameter ID of each parameter that may not appear with the parameters this parameter data area describes. This appendage may follow any parameter data appendage.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	NOTCOUNT	Number *n* of parameter IDs that follow.
2(2)	2x*n*	NOTID*n*	List of IDs of conflicting parameters.

Necessary Parameters Appendage

This appendage contains the parameter IDs of parameters that must appear with the parameters this parameter data area describes. The parameters are grouped into arrays. One parameter in each array must appear. This appendage may follow any other parameter data appendage.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	NGRPTOT	Number of arrays that follow:
Each array contains:			
0(0)	2	NGRPCNT	Number, *n*, of ID numbers that follow.
	2x*n*	NGRPID*n*	List of parameter ID numbers for necessary parameters.

Prompt Appendage

This appendage contains an offset from the beginning of the prompt PDL to the PDE for prompting information needed by parameters this parameter data area describes. This appendage may follow any other parameter data appendage.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	PDECNT	Number of sets of fields that follow. There is a set of three fields for each PDL offset for each parameter.
Each set contains:			
	2	PDEPRMID	Contains the parameter ID of the parameter in the prompt PDL.
	2	PDEPCLID	Contains the parameter ID of the parameter whose subparameters have been prompted for.
	2	PDEOFST2	Number of bytes from the beginning of the prompt PDL to the PDE for this parameter.

Subparameter Appendage

This appendage contains all the subparameter IDs. This appendage may follow any other parameter data appendage.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	SUBCOUNT	Number of sets of fields that follow. There is a set of two fields for each subparameter.
Each set contains:			
	2	PARMTYPE	Identifies this subparameter as positional, 'P', or keyword, 'K'.
	2	SUBID	Subparameter ID.

IDCRILT

Command Name Table

IDCRILT contains a table of all verbs and verb abbreviations and their command descriptor load module names.

Created by	Modified by	Used by	Size
IBM-supplied	None	IDCRI02 IDCRI04	290

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	LNAMECNT	Number, *n*, of table entries.
2(2)	16x*n*	TABLE*n*	*n* table entries.
Each entry contains the following:			
	8	TBLVERB	Verb or verb abbreviation.
	8	TBLLNAME	Corresponding command descriptor load module name for this verb.

CRA Access Parameter List

The CRA access parameter list provides VSAM catalog management with information necessary to access the CRA as a catalog. It is pointed to by the ACB when the UCRA bit in the ACB is on for the OPEN of a CRA by EXPORTRA. The CRA access parameter list consists of three control blocks. The ACB points directly to the ACC (access method services catalog communication table) which in turn points to the CTT (CRA access translate table) and the VTT (CRA volume timestamp table).

Created by	Modified by	Used by	Size
IDCRC01	None	VSAM catalog management	Variable

Access Method Services/Catalog Communication Table (ACC) Description

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	ACCTRANT	Address of the CRA access translate table (CTT).
4(4)	1	*	Reserved.
5(5)	.3	ACCDSNCI	Control interval number used when LOCATEs are performed via true names.
8(8)	4	ACCVOLTT	Address of the volume timestamp table.

CRA Access Translate Table (CTT) Description

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	CTTENTNO	Number of entries in the table.
4(4)	4x*n*	CTTENTRY	Variable number (*n*) of 4-byte entries.
	1	CTTENTYP	Type of CRA record.
	.3	CTTCATCI	Catalog control interval number of the CRA control interval for this entry.

CRA Volume Timestamp Table (VTT) Description

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	VTTENTNO	Number of entries in the table.
4(4)	14x*n*	VTTENTRY	Variable number (*n*) of 14-byte entries.
	6	VTTVOLSR	Volume serial number for the timestamp of this entry.
	8	VTTTMSTP	The timestamp that is in the format 4 DSCB on this volume.

Dump List

The dump list tells the UDUMP macro which areas to dump. The dump list consists of entries that describe the individual fields. If one or more fields are to be repeated, they can be described as an array where each group of fields is an element in the array. In such cases, the array is preceded by a dump list entry called an array header. The array header causes the fields to be repeated. The end of the dump list is indicated by an entry called the dump list terminator.

Individual entries are printed as *name=data*. Each field in an array is printed as *name(n)=data*. The array name is printed before the array elements. All arrays start on a new line.

Created by	Modified by	Used by	Size
All routines	All routines	IDCDB02	Variable

Individual Field Entry

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	DMPITMNM	Name to be printed with the field. The name is aligned left and padded with blanks.
8(8)	4	DMPITMPT	Address of field to be dumped.
12(C)	2	DMPITMLN	Number of bytes to dump. For hexadecimal, bit, or character strings the number is from 1 to 256. For fixed binary, the number is from 1 to 4.
14(E)	. . 1	DMPITMTP	One character indicating the type of data in field: H – Hexadecimal printed as 2 characters per byte. B – Bit string printed as 8 characters per byte. C – Character printed as 1 character per byte. F – Fixed binary printed as a signed number for halfwords or fullwords or as an unsigned number for 1 or 3 bytes. Leading zeros are suppressed.
15(F)	. . . 1	*	Reserved.

Array Header Entry

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	DMPARYNM	Name to be printed at the start of the array. The name is aligned left and padded with blanks.
8(8)	2	DMPARYSZ	Number of bytes in each input element of the array. The number can be from 1 to 32767.
10(A)	. . 2	DMPARYIC	Number of following individual items that are in the array. The number can be from 1 to 32767.
12(C)	2	DMPARYEX	Number of times to repeat the individual fields. The number can be from 1 to 99.
14(E)	. . 1	DMPARTYTP	Array header type—contains 'A'.
15(F)	. . . 1	*	Reserved.

Dump List Terminator Entry

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	*	End of dump list indicator—contains X'FF'.

DARGLIST

Dynamic Data Argument List

The dynamic data argument list describes variable data to be printed. It is always an argument for a print request (UPRINT macro).

Created by	Modified by	Used by	Size
Calling routine	None	IDCTP01	Variable

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	DARGDBP	Contains the address of block of data referred to by format list or zero.
4(4)	4	DARGRETP	Zero if printing is to occur; nonzero if no printing is to occur. If nonzero, contains the address of the area in which the formatted print lines are to be returned from the text processor and not printed. Spacing control characters are not returned. The data is truncated to the length (DARGRETL) of the provided area if necessary. Data will be returned to the specified location.
8(8)	4	DARGSTID	Zero if a format list is also passed as a parameter. If nonzero, contains the text structure identification (STID) for static text element to be used as the format list.

Each DARGSTID contains:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	3	DARGSMOD	Last 3 characters of the text structure module name.
	. . . 1	DARGSENT	Static text entry.
12(C)	2	DARGILP	Length of block whose address is in DARGDBP.
14(E)	. . 2	DARGCNT	Number of insert and replication elements contained in DARGARY.
16(10)	2	DARGRETL	Length of the return-data area—that is, DARGRETP.
18(12)	. 1	DARGIND	Offset to add to the print column in the format list (FMTOCOL).
19(13)	. . 1	DARGFLGS	Flag byte.
	1...	DARGFULL	Length is greater than 32K.
	.xxx xxxx	*	Reserved.

20(14)	8x*n*	DARGARY*n*	Group array in one of two formats. The following fields are repeated *n* times, where *n* = DARGCNT.

For insert data each array contains:

	2	DARGINS	Insert reference number.
	. . 2	DARGINL	Input data length of the field pointed to by DARGDTM.
	4	DARGDTM	Dynamic data pointer, address of field to use for this insert.

For replication data each array contains:

	2	DARGREP	Replication reference number.
	2	DARGPCT	Replication count, number of times to replicate a series of format substructures (FMTLIST). This field is not used for replication structures.

DLST

Domain List

The DLST contains names of all catalogs or the names of all VVDS referenced by data set being DIAGNOSED.

Created by	Modified by	Used by	Size
IDCDA02	IDCDA01 IDCDA02 IDCDA03	IDCDA01 IDCDA02 IDCDA03	Variable

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	DLSTID	ID='DLST'.
4(4)	1	*	Reserved.
5(5)	3	DLSTLEN	Length for DLST.
8(8)	4	DLSTAVAL	Address of next DLST entry.
12(C)	4	DLSTLAST	Address of last DLST entry.
16(10)	4	DLSTNEXT	Address of next DLST block.
20(14)	var.	DLSTNAMS	Entries of DNTRY.

Each DLSTNAMS entry contains the following:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	DNHDR	Size of fixed header.
2(2)	1	DNFLAG	Flag bits.
	1...	DNFOUND	Entry found on scan.
	.1..	DNADDED	Entry added on scan.
	..xx xxxx	*	Reserved.
3(3)	1	DNNMLL	Length of name.
4(4)	var.	DNNAME	Name 1 through 44 bytes.

ELST

Entry Names List

The ELST contains variable length entry names during DIAGNOSE.

Created by	Modified by	Used by	Size
IDCDA02	IDCDA01 IDCDA02 IDCDA03	IDCDA01 IDCDA02 IDCDA03	Variable

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	ELSTID	ID='ELST'.
4(4)	1	*	Reserved, unused subpool for FREEMAIN.
5(5)	3	ELSTLEN	Length for FREEMAIN.
8(8)	4	ELSTAVAL	Address of next ELST entry.
12(C)	4	ELSTLAST	Address of last ELST entry.
16(10)	4	ELSTNEXT	Address of next ELST block.
20(14)	var.	ELSTNAMS	Entries of ENTRY.

Each ELSTNAMS entry contains the following:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	1	ENNAMLL	Length of the name.
	var	ENAME	Name 1 through 44 bytes. Need a UCLOSE.

ERCNVTAB

Error Conversion Table

The error conversion table is passed whenever a UERROR macro is issued. It contains the information necessary to convert numeric error codes into prose messages.

Created by	Modified by	Used by	Size
All routines	None	IDCTP06	32

	Bytes and		Description:
Offset	Bit Pattern	Field Name	Content, Meaning, Use
0(0)	1	ERTYPE	Type of error code to be converted.
	1...	ERCATLG	VSAM catalog management error.
	.1..	EROSCAT	OS/VS catalog error—used in OS/VS1 only.
	..1.	EROYNAL	Dynamic allocation error.
1(1)	.1	EROPER	VSAM catalog operation being performed when error occurred. Only one operation type allowed per UERROR invocation.
	1...	ERCATLC	CMS locate.
	.1..	ERCATDE	CMS define.
	..1.	ERCATDL	CMS delete.
	...1	ERCATAL	CMS alter.
2(2)	..1	EROSOPER	OS/VS catalog operation being performed—used in VS1 only.
3(3)	...1	*	Reserved.
4(4)	4	*	Reserved.
8(8)	4	*	Reserved.
12(C)	4	ERDSNM	Address of data set name or volume serial number associated with the catalog management request. The data set name is contained in a 44-byte field padded with blanks; the volume serial number is contained in a 44-byte field padded with binary zeros.
16(10)	4	ERCATRC	Catalog management return code.
16(10)	4	ERDYNRC	Dynamic allocation return code.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
20(14)	4	ERCPLPT	Address of catalog parameter list (CTGPL) issued that resulted in error condition.
20(14)	4	ERDARBPT	Address of SVC 99 request block issued which resulted in a dynamic allocation error condition.
24(18)	4	*	Reserved.
28(1C)	4	*	Reserved.

STAEPARM

ESTAE Argument List

The STAEPARM is passed to the ESTAE exit routine in IDCSA10 when an ABEND occurs. It defines whether retry is to be attempted and what recovery must be performed.

Created by	Modified by	Used by	Size
IDCSA10	IDCIO05 IDCSA10 IDCSA06	IDCSA06 IDCIO05 IDCSA10	

	Bytes and		Description:
Offset	Bit Pattern	Field Name	Content, Meaning, Use
0(0)	8	STAHEAD	Set with 'STAEPARM' by IDCSA10, when an ESTAE environment is established.
8(8)	4	STAVUCB	Address of the UCB where the mass storage volume is mounted or was demounted. If STAVMNT or STAVDMNT flag is set, this field must be initialized.
12(C)	6	STAVVOL	Volume serial number of the mass storage volume if STAVMNT or STAVENQ is set.
18(12)	..2	*	Reserved.
20(14)	4	STARUCB	Address of the UCB for the real 3330 unit which is cleared if STACLEAR flag is set.
24(18)	6	STARVOL	Volume serial number cleared from the UCB if STACLEAR is set.
30(1E)	..2	*	Reserved.
32(20)	4	STARTTR	TTR of the VTOC cleared from the UCB if STACLEAR is set.
36(24)	4	STARTSAV	Address of the save area for the retry routine's registers.
40(28)	4	STAEXSAV	Address of the registers at the time the ESTAE environment was established. Used by ESTAE routine to restore registers.
44(2C)	4	STARTADD	Address of the retry routine entry point.

	Bytes and		Description:
Offset	Bit Pattern	Field Name	Content, Meaning, Use
48(30)	4	STATCB	Contains the initiator's TCB. Used to dequeue a volume.
52(34)	1	STARCVY	Recovery flags:
	1...	STAVMNT	Indicates that a mass storage volume is mounted.
	.1..	STAVDMNT	Indicates that a mass storage volume was demounted from a unit, but the UCB has not been marked "NOT READY" yet.
	..1.	STAVENQ	Indicates that an enqueue was done on a volume serial number in STAVVOL. This bit is set off when a dequeue is done.
	...1	STAVPOST	Indicates that the mass storage volume's serial number and VTOC TTR were posted in the UCB following the mount.
	.... 1...	STAVCLEAR	Indicates that the VTOC TTR and volume serial number were removed from a real 3330 UCB, but the volume is still mounted.
53(35)	.1	STARETRY	Retry flags:
	1...	STAIO05	Indicates that the retry routine for IDCIO05 is to be invoked at ABEND.
	.1..	STAOFF	Indicates that the ESTAE environment was canceled as a result of an ABEND.

EXCLAGL

Exclusive Control Argument List

The EXCLAGL is passed when a UMSSUNIT macro is issued with a select or change request. It defines a request to either (1) select a DD statement and UCB that can be used for the volume if the volume is already allocated exclusively to the job step, or, (2) change the allocation of a unit and volume to exclusive.

Created by	Modified by	Used by	Size
Calling routine	IDCSA06	IDCSA06	23

Select Request

	Bytes and		Description:
Offset	Bit Pattern	Field Name	Content, Meaning, Use
0(0)	8	EXCLHEAD	Set by IDCSA06 with 'SELECTXƀ'.
8(8)	4	EXCLUCBP	Address of a 4-byte area in which IDCSA06 returns the address of the UCB that can be used to mount the volume.
12(C)	4	EXCLDDP	Address of an 8-byte area in which IDCSA06 returns the ddname that can be used to open the VTOC on the volume.
16(10)	6	EXCLVOL	Volume serial number of the volume that is being processed by the caller.
22(16)	1	EXCLFLAG	Ignored.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	EXCLHEAD	Set by IDCSA06 with 'CHANGEXb̸'.
8(8)	4	EXCLUCBP	Address of an area that contains the UCB address, or zeros if the UCB address is unknown. If EXCLUCBP contains zeros, IDCSA06 puts the UCB address in the area upon return.
12(C)	4	EXCLDDP	Address of the ddname for the volume being processed.
16(10)	6	EXCLVOL	The volume serial number of the volume whose allocation is to be changed to exclusive, or zeros. If the EXCLMNT bit is set, this field must be specified.
22(16)	1	EXCLFLAG	Option bits:
	1...	EXCLMNT	Indicates the allocation of the volume is to be changed to exclusive only if the volume is mounted on the unit. When this bit is zero, the allocation of any volume mounted on the unit is changed to exclusive.

EXGARG

EXCP GET Argument List

The EXGARG is passed when UEXCP is issued with a GET request. It defines a request to read a data set.

Created by	Modified by	Used by	Size
Calling routine	IDCIO05	IDCIO05	32

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	EXGHEAD	Set with 'GETb̸b̸b̸b̸b̸' by IDCIO05.
8(8)	4	EXGCTLBK	Address of the IOXCTLBK created during the open.
12(C)	4	EXGCCHH1	Address of an area that contains the physical CCHHR (search) of the first records to be read. This field is required only when the EXOTAB open option is selected.
16(10)	4	EXGCCHH2	Address of an area that contains the CCHHR (seek) in the count field of the first record to be read.
20(14)	4	EXGRECNO	The number of records to be read. The value cannot exceed the number of records on 1 track.
24(1B)	4	EXGDATAP	Address of the area into which the records are read. Each record is read and stored contiguously.
28(1C)	1	EXGKEYLN	Key length of the records being read.
29(1D)	.1	*	Reserved.
30(1E)	..2	EXGDATAL	Data length of the records being read.

EXOARG

EXCP OPEN Argument List

The EXOARG is passed when a UEXCP is issued with an open request. It defines a request to open a data set.

Created by	Modified by	Used by	Size
Calling routine	IDCIO05	IDCIO05	34

	Bytes and		Description:
Offset	**Bit Pattern**	**Field Name**	**Content, Meaning, Use**
0(0)	8	EXOHEAD	Set with 'OPENƀ ƀ ƀ ƀ' by IDCIO05.
8(8)	4	EXODDN	Address of 8-byte ddname.
12(C)	4	EXODSN	Address of 44-byte data set name.
16(10)	4	EXOVSN	Address of 8-byte volume serial number.
20(14)	4	EXOUCB	Address of the UCB.
24(18)	4	EXOEXT	Address of 10-byte extent return area. Upon return, the area contains the beginning and ending CCHHR of the first extent.
28(1C)	4	EXOCTLBK	Address of 4-byte area for return of IOXCTLBK address built by I/O adapter.
32(20)	1	EXOPT	EXCP open options:
	1...	EXOTAB	Open for MSC tables. This option is mutually exclusive of the other open options.
	.1..	EXOLAB	Open for the VTOC to read or write the volume label. This option is mutually exclusive of the other open options.
	..1.	EXOPASS	Open to check for password protected data sets. This option is mutually exclusive of other open options.
	...1	EXONEW	Open to initialize the volume with IPL records, label record, and VTOC. This option is mutually exclusive of the other open options.
	.... 1...	EXOVTOC	Open to read the VTOC DSCBs. This option is mutually exclusive of other open options.
	.... .1..	EXOREP	OPENR is specified: open is being done for a REPAIRV function. This option is mutually exclusive of the other open options.
	xx	*	Reserved.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
33(21)	.1	EXFLG	Open flags.
	1...	EXOREAD	Read-only access.
	...1	EXORVS	The user wants to open a VSAM data set for REPAIRV processing. This bit is an option of OPENR.
	.... 1...	EXOSPK	Open is for an entire staging pack. The user wants to be able to read an entire staging pack taken offline from the 3850. This is an option of OPENR.
	.... .1..	EXORDS	The user wants to open a data set for a repair function. This is an option of OPENR.
	.... ..1.	EXORVT	The user wants to open a VTOC for a repair function. This is an option of OPENR.
	1	EXOVTH	The user wants to open a VTOC header for a repair function. This is an option of OPENR.
	.xx.	*	Reserved.

EXPARG

EXCP PUT Argument List

The EXPARG is passed when UEXCP is issued with a PUT request. It defines a request to write a data set.

Created by	Modified by	Used by	Size
Calling routine	IDCIO05	IDCIO05	20

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	EXPHEAD	Set with ‘PUT ƀƀƀƀƀ’ by IDCIO05.
8(8)	4	EXPCTLBK	Address of the IOXCTLBK built during the open.
12(C)	4	EXPDATA	Address of the data-to-write block. EXPDATAB structure is used to initialize this block.
16(10)	4	EXPCCHHR	Address of an area that contains the CCHHR where the data will be written.

EXPDATAB

PUT Data Block

The EXPDATAB is passed when UEXCP is issued with a PUT request. It defines a request to write a data set. The EXPARG points to EXPDATAB.

Created by	Modified by	Used by	Size
Calling routine	IDCIO05	IDCIO05	Variable

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	EXPRECNO	Number of records to be written.
4(4)	8x*n*	EXPARRAY*n*	Data-to-write array. There is an array entry for each record to be written.

Each entry contains:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	4	EXPDATAP	Address of the record to be written.
	1	EXKEYLN	Key length of the record being written. Required for each record.
	.1	*	Reserved.
	..2	EXPDATAL	Data length of the record being written. Required for each record.

FMPL

Field Management Parameter List

The field management parameter list is passed whenever module IDCRC04 is called within EXPORTRA and LISTCRA. It contains information and pointers which enable IDCRC04 to extract data from records within the catalog or CRA.

Created by	Modified by	Used by	Size
IDCRC01 IDCLR01	IDCRC04	IDCRC04	Variable

Field Management Parameter List Description

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	FMPLFLNO	Number of FMFL pointers.
1(1)	.3	FMPLBCIN	Control interval number of the base record.
4(4)	4	FMPLGRTN	Address of the GET routine.
8(8)	4	FMPLWKAR	Address of the field management work area.
12(C)	4	FMPLUPTR	Value passed to user GET routine at input/output processing time.
16(10)	1	FMPLRTCD	Return code from a call to IDCRC04.
17(11)	.1	*	Reserved.
18(12)	..2	FMPLENTH	Length of the output area provided by caller.
20(14)	4	FMPLOAR	Address of the output area.
24(18)	4xn	FMPLFMFL	Array of variable number (*n*) of 4-byte FMFL pointers.

Field Management Field List (FMFL) Description

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	FMFLDLNO	Number of length/data pairs passed by caller.
1(1)	.1	FMFLTSTC	Compare test condition code.
2(1)	..1	FMFLGRPC	Field group code supplied by caller.
3(1)	...1	FMFLINDS	FMFL indicator flags.
	xxxx xxx.	*	Reserved.
	1	FMFLSUCC	Bit indicating success of test. 0=test is successful. 1=test is unsuccessful.
4(4)	4	FMFLWKAR	Work area for field management.
8(8)	4	FMFLDNAM	Pointer to 8-byte field name.
12(C)	4	FMFLTCHN	Address of next test FMFL.
16(10)	8x*n*	FMFLDATA	Variable number (*n*) length/data pointer pairs.
	4.	FMFLENTH	4-byte length of supplied data.
	.4	FMFLADDR	4-byte address of supplied data.

FMTLIST

Format List

The format list defines the format of printed output. This list consists of several substructures, each identified by its flag byte. Format lists exist in the text structures, where they are referenced by STID numbers (static text identifiers). Optionally, they may be passed as an argument of the UPRINT macro, in which case the DARGLIST argument does not furnish a STID.

Created by	Modified by	Used by	Size
Calling routine	None	IDCTP01	Variable

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	FMTFLGS	Flags:
	1...	FMTEOLF	End of structure.
	.1..	FMTSCF	Space control.
	..1.	FMTIDF	Insert data.
	...1	FMTBDF	Block data.
	.... 1...	FMTREPF	Replication.
	.... .1..	FMTSTF	Static text.
	.... ..1.	FMTDFF	Default data.
	1	FMTHDF	Header line.

Interpretation of each substructure of the format list depends on the value of FMTFLGS. Each of the possible substructures is shown below.

Spacing

The spacing substructure of the format list specifies the line spacing or carriage control to use while printing. The default spacing is used only when a line is not immediately preceded by a spacing substructure. A spacing substructure embedded in an entry causes the previously formatted data to be printed and signals the start of a new line.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	FMTFLGS	Flag byte: X‘40’ or X‘41’.
1(1)	. 1	*	Reserved.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
2(2)	. . 2	FMTSPF	Space factor: if FMTSPT is equal to 'A', this is the absolute line number to use for printing this line. If FMTSPT is equal to 'R', this is the number of spaces to take before printing. Page overflow results in printing on the first line of the next page.
4(4)	1	FMTSPT	Spacing type: 'A' signifies absolute line number in FMTSPF, and 'R' signifies relative line number. 'E' signifies page eject.
5(5)	. 1	*	Reserved.

Insert Data

The insert data substructure refers to data defined in the dynamic data argument structure, and identified by reference number. This represents variable data to be inserted into the printed line.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	FMTFLGS	Flag byte: X'20' or X'A0'. (X'A0' also denotes end-of-structure.)
1(1)	. 1	*	Reserved.
2(2)	. . 2	FMTRFNO	Insert reference number: identification number for dynamic data insert that defines the input data to be used for formatting.
4(4)	2	*	Reserved.
6(6)	. . 2	FMTOCOL	Print line column for beginning of this field, or if FMTBS is equal to one, the offset from the column indicated by field PCTAPC.
8(8)	2	FMTOLEN	Output field length. If FMTOLEN is equal to zero or 32767, then the full, converted input length is used.
10(A)	. . 1	FMTCNVF	Flags to define conversion and formatting to be done:
	1...	FMTBH	Byte to printable, hexadecimal representation.
	.1..	FMTBHA	Byte to hexadecimal, preceded by "X" and followed by a single quote.
	..1.	FMTBHD	Standard dump format. FMTOCOL and FMTOLEN are ignored.
	...1	FMTBD	Binary to unpacked decimal characters.
	.... 1...	FMTPU	Packed to unpacked decimal characters.
11(B)	. . . 1	FMTCNVF (cont.)	Conversion flags (continued).
	1...	FMTZS	Suppress leading zeros by replacing with blanks.
	.1..	FMTAL	Aligned left; the high-order nonzero digit is put in first print column as specified by FMTOCOL.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	..1.	FMTSS	Suppress signs.
	...1	FMTBS	Suppress all trailing blanks but one of the preceding field; add the offset in FMTOCOL to the value in PCTAPC for the print column.
	.... 1...	FMTAR	Align EBCDIC character strings to the right. The print column is added to the print field length to determine the last printable position.

Default Text

The default text substructure is only used when it immediately follows an insert data substructure. When examining the insert structure, the value in DARGINS is compared to the value in FMTRFNO. If no match is found, the next format structure is examined to determine whether it is a default structure. If the flag FMTDFF is on in this next structure, the structure is used. In all other cases, it is skipped over.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	FMTFLGS	Flag byte: X'02' or X'82'. (X'82' also denotes end-of-structure.)
1(1)	. 1	*	Reserved.
2(2)	. . 2	FMTILEN	Length of the default text.
4(4)	2	FMTIOFF	Offset from the beginning of the format structures to the default text (which follows the format structures).
6(6)	. . 2	FMTOCOL	Print line column, same as for insert substructure.
8(8)	2	FMTOLEN	Output field length, same as for insert substructure.
10(A)	. . 2	FMTCNVF	Conversion flags, same as for insert substructure.

Block Format

The block format substructure of the format list defines a block of variable data from which fields are extracted for printing.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	FMTFLGS	Flag byte: X'10' or X'90'. (X'90' also denotes end-of-structure.)
1(1)	. 1	*	Reserved.
2(2)	. . 2	FMTILEN	Length of the input field. If FMTILEN is zero or if FMTILEN is greater than DARGILP minus FMTIOFF, then the input length in DARGILP is used.
4(4)	2	FMTIOFF	Offset from the beginning of the input data block at which this field begins. The beginning of the data block is in DARGDBP.
6(6)	. . 2	FMTOCOL	Print line column, same as for insert substructure.
8(8)	2	FMTOLEN	Output field length, same as for insert substructure.
10(A)	. . 2	FMTCNVF	Conversion flags, same as for insert substructure.

Replication

The replication substructure defines substructures of the format list that are to be repeated. The replication substructure always precedes the first substructure to be repeated.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	FMTFLGS	Flag byte: X'08'. (May not have end-of-list flag on.)
1(1)	. 1	*	Reserved.
2(2)	. . 2	FMTRFNO	Reference number to identify the dynamic argument that contains the replication count.
4(4)	2	FMTRBC	Number of substructures that follow that are to be replicated.
6(6)	. . 2	FMTRIO	Offset to add to all offsets contained in block format substructures being replicated to access the input fields.

Static Text

The static text substructure defines data from the text structures to be placed in the printed line.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	FMTFLGS	Flag byte: X‘04’ or X‘84’. (X‘84’ also denotes end-of-structure.)
1(1)	. 1	*	Reserved.
2(2)	. . 2	FMTSTL	Length of static text field.
4(4)	2	FMTSTO	Offset to static text which follows format structures.
6(6)	. . 2	FMTOCOL	Print line column, same as for insert substructure.
8(8)	2	FMTOLEN	Output field length, same as for insert substructure.
10(A)	..2	FMTCNVF	Conversion flags, same as for insert substructure.

FDT

Function Data Table

The function data table is an encoded representation of a command. The reader/interpreter constructs the FDT from information found in the command. All defaults are resolved; no conflicts are allowed among the values of an FDT.

The FDT is not one structure, but rather several small structures that are pointed to by a primary vector of addresses called the FDTTBL. For a parameter that appears in a repeated subparameter list, a secondary vector of addresses results. Figure 13 illustrates this vector and the various small structures to which it points.

The FDT primary vector, FDTTBL, is variable in length. It consists of the command's verb as an 8-byte EBCDIC string, followed by a variable number of fullword pointers. The number of pointers depends on the specific command. There is one pointer per parameter defined in the command descriptor. If a pointer is reserved or is not used because the respective parameter has not been specified, the pointer contains zero.

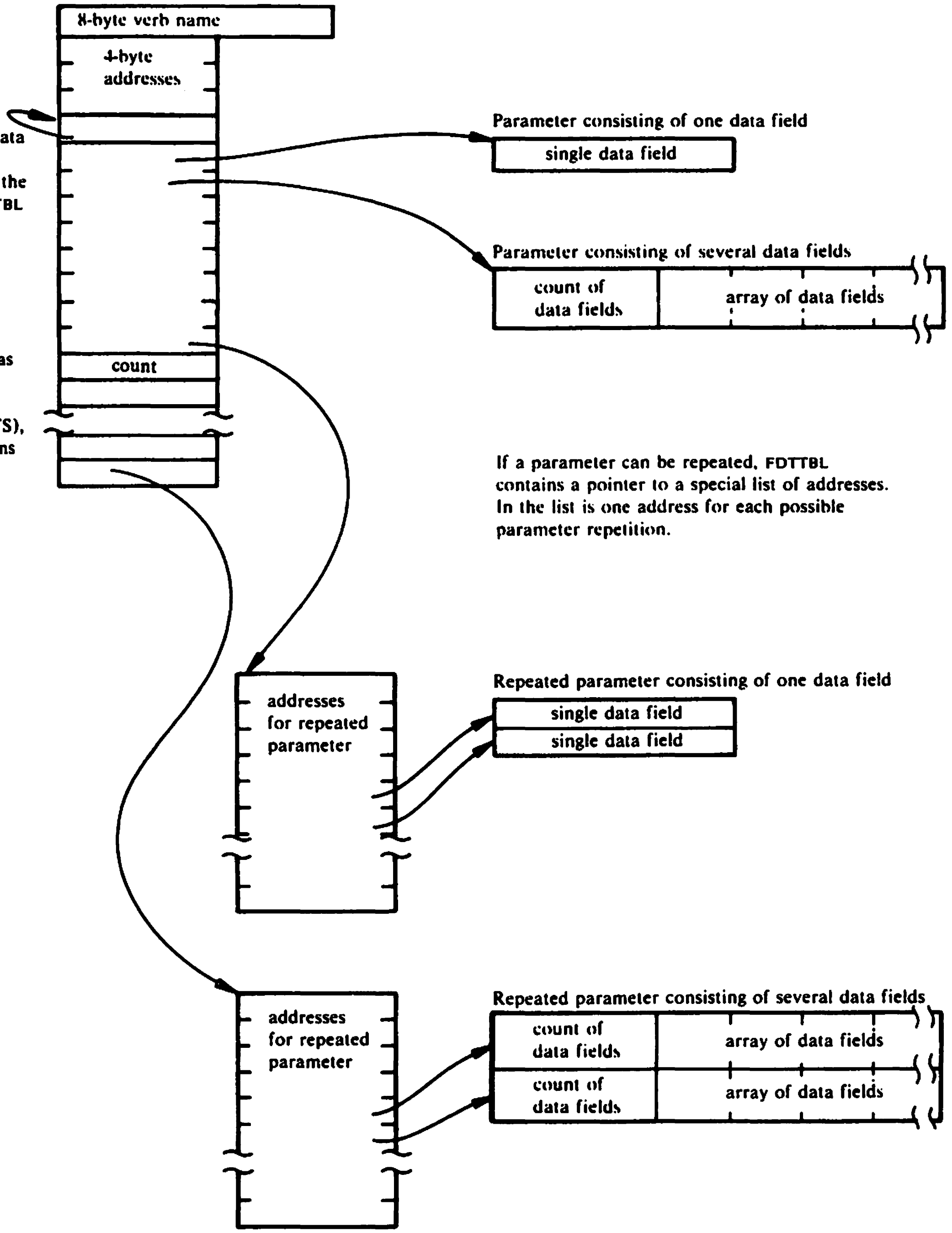
The FDTTBL points to data areas in one of three formats depending upon the input provided by the parameter. If there is more than one data field, an array of data fields is generated. The array is preceded by a count of the array elements. The count is in a fullword for an array of number data areas and in a halfword for an array of any other data areas. The array consists of one element for each data field supplied as input to the parameter. Every element in the array has the same format—one of the three formats shown below. In the following formats, the last 3 characters of the field name are as shown. The first characters may vary and are indicated by *.

Created by	Modified by	Used by	Size
IDCRI01 IDCRI04	None	FSR	Variable

Number Data Area

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	*VAL	Contains the input number in fixed-point binary.

FDTTBL contains one address for each parameter in the command. Each parameter consists of zero, one, or more data fields.


8-byte verb name
4-byte addresses
count
Parameter consisting of one data field
single data field
Parameter consisting of several data fields
count of data fields
array of data fields
addresses for repeated parameter
Repeated parameter consisting of one data field
single data field
single data field
addresses for repeated parameter
Repeated parameter consisting of several data fields
count of data fields
array of data fields
count of data fields
array of data fields

If a parameter consists of no data fields (such as WRITECHECK), the address in FDTTBL points to itself.

If a parameter has repeated subparameters such as OBJECTS), FDTTBL contains the number of subparameters.

If a parameter can be repeated, FDTTBL contains a pointer to a special list of addresses. In the list is one address for each possible parameter repetition.

Figure 13. FDT (Function Data Table)

String Data Area

For a character string or hexadecimal string with or without a password the format is:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	*PLN	Number of characters in the following password. This field does not exist if a password is not allowed with the string.
1(1)	8	*PAS	Password, if supplied, left-justified and padded with blanks. This field does not exist if a password is not allowed with the string.
9(9)	1	*LEN	Number of characters in the following field.
10(A)	*v*	*VAL	Character string left-justified and padded with blanks. The string does not contain delimiters. Double apostrophes are converted to single apostrophes and hexadecimal input is converted to EBCDIC.

Data Set Name or Data Area

For a data set name or generic data set name all with or without a password and member name, the format is:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	*PLN	Number of characters in the following password.
1(1)	8	*PAS	Password, if supplied, left-justified and padded with blanks.
9(9)	1	*POS	Position of any * in the data set name.
10(A)	1	*FLG	Flag byte:
	1...	*FUQ	This flag is used only if access method services is invoked interactively with TSO. A '1' means the data set name is to be qualified according to TSO naming conventions. Refer to *TSO Guide to Writing a Terminal Monitor Program or Command Processor*. '0' means the data set name is to be used without qualification—that is, without any additions.
11(B)	1	*MLN	Number of characters in the following member name.
12(C)	8	*MEM	Member name, if supplied, left-justified and padded with blanks.
20(14)	1	*LEN	Number of characters in the following field.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
21(15)	*v*	*VAL	Data set name or generic data set name in EBCDIC.

FDTs for Specific Commands

The FDT for each command is shown in two different ways in the following sections. First, there is a table relating the pointers to the parameters in the command. Any omitted fields in this table contain zero. Second there is the FDT description as it is used by the FSR for the command.

ALTER FDT

Offset	Content	
0 (0)	ALTE	Rbbb
8 (8)	↑entryname/password	↑CATALOG
16 (10)	↑catname/password	↑dname
24 (18)	↑NEWNAME	↑FILE
32 (20)	0	↑MASTERPW
40 (28)	↑CONTROLPW	↑UPDATEPW
48 (30)	↑READPW	↑CODE
56 (38)	↑ATTEMPTS	↑AUTHORIZATION
64 (40)	↑entrypoint	↑string
72 (48)	0	↑TO
80 (50)	↑FOR	↑OWNER
88 (58)	↑ERASE	↑NOERASE
96 (60)	↑SHAREOPTIONS	0
104 (68)	↑NULLIFY	↑MASTERPW
112 (70)	↑CONTROLPW	↑UPDATEPW
120 (78)	↑READPW	0
128 (80)	↑FREESPACE	↑cipercent
136 (88)	↑capercent	↑WRITECHECK
144 (90)	↑NOWRITECHECK	↑BUFFERSPACE
152 (98)	↑ADDVOLUMES	↑REMOVEVOLUMES
160 (A0)	0	↑INHIBIT
168 (A8)	↑UNINHIBIT	↑OWNER
176 (B0)	↑CODE	↑RETENTION
184 (B8)	↑AUTHORIZATION	↑MODULE
192 (C0)	↑STRING	↑crossregion

Offset	Content	
200 (C8)	↑crosssystem	↑EMPTY
208 (D0)	↑NOEMPTY	↑SCRATCH
216 (D8)	↑NOSCRATCH	0
224 (E0)	↑EXCEPTIONEXIT	↑KEYS
232 (E8)	↑length	↑offset
240 (F0)	↑RECORDSIZE	↑average
248 (F8)	↑maximum	↑UNIQUEKEY
256 (100)	↑NONUNIQUEKEY	↑UPGRADE
264 (108)	↑NOUPGRADE	↑UPDATE
272 (110)	↑NOUPDATE	↑EXCEPTIONEXIT
280 (118)	↑STAGE	↑BIND
288 (120)	↑CYLINDERFAULT	↑NODESTAGEWAIT
296 (128)	↑DESTAGEWAIT	↑BUFND
304 (130)	↑BUFNI	↑STRNO

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb aligned left and padded with blanks—ALTERƀ ƀ ƀ.
8(8)	4	NTRY	Address of information supplied through the *entryname/password* parameter.
12(C)	4	CAT	Address of this pointer itself if the CATALOG parameter is supplied.
16(10)	4	CATLG	Address of information supplied through the *catname/password* subparameter of the CATALOG parameter.
20(14)	4	CATDN	Address of information supplied through the *dname* subparameter of the CATALOG parameter.
24(18)	4	NEWNM	Address of information supplied through the NEWNAME parameter.
28(1C)	4	INDD	Address of information supplied through the FILE parameter.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
32(20)	4	*	Reserved—contains zeros.
36(24)	4	MASTR	Address of information supplied through the MASTERPW parameter.
40(28)	4	CNTVL	Address of information supplied through the CONTROLPW parameter.
44(2C)	4	UPDAT	Address of information supplied through the UPDATEPW parameter.
48(30)	4	READ	Address of information supplied through the READPW parameter.
52(34)	4	CODNM	Address of information supplied through the CODE parameter.
56(38)	4	ATTP	Address of information supplied through the ATTEMPTS parameter.
60(3C)	4	AUTH	Address of this pointer itself if the AUTHORIZATION parameter is supplied.
64(40)	4	USVR	Address of information supplied through the *entrypoint* subparameter of the AUTHORIZATION parameter.
68(44)	4	USAR	Address of information supplied through the *string* subparameter of the AUTHORIZATION parameter.
72(48)	4	*	Reserved—contains zeros.
76(4C)	4	TO	Address of information supplied through the TO parameter.
80(50)	4	FOR	Address of information supplied through the FOR parameter.
84(54)	4	OWNER	Address of information supplied through the OWNER parameter.
88(58)	4	ERASE	Address of this pointer itself if the ERASE parameter is supplied.
92(5C)	4	NERAS	Address of this pointer itself if the NOERASE parameter is supplied.
96(60)	4	SHARE	Address of this pointer itself if the SHAREOPTIONS parameter is supplied.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
100(64)	4	*	Reserved—contains zeros.
104(68)	4	NULLF	Address of this pointer itself if the NULLIFY parameter is supplied.
108(6C)	4	NMSTR	Address of this pointer itself if the MASTERPW subparameter of the NULLIFY parameter is supplied.
112(70)	4	NCNTV	Address of this pointer itself if the CONTROLPW subparameter of the NULLIFY parameter is supplied.
116(74)	4	NUPDT	Address of this pointer itself if the UPDATEPW subparameter of the NULLIFY parameter is supplied.
120(78)	4	NREAD	Address of this pointer itself if the READPW subparameter of the NULLIFY parameter is supplied.
124(7C)	4	*	Reserved—contains zeros.
128(80)	4	FSPAC	Address of this pointer itself if the FREESPACE parameter is supplied.
132(84)	4	FSPCI	Address of information supplied through the *cipercent* subparameter of the FREESPACE parameter.
136(88)	4	FSPCA	Address of information supplied through the *capercent* subparameter of the FREESPACE parameter.
140(8C)	4	WRTCK	Address of this pointer itself if the WRITECHECK parameter is supplied.
144(90)	4	NWTCK	Address of this pointer itself if the NOWRITECHECK parameter is supplied.
148(94)	4	BUFSZ	Address of information supplied through the BUFFERSPACE parameter.
152(98)	4	ADDVL	Address of information supplied through the ADDVOLUMES parameter.
156(9C)	4	REMVL	Address of information supplied through the REMOVEVOLUMES parameter.
160(A0)	4	*	Reserved—contains zeros.
164(A4)	4	INHIB	Address of this pointer itself if the INHIBIT parameter is supplied.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
168(A8)	4	UNHIB	Address of this pointer itself if the UNINHIBIT parameter is supplied.
172(AC)	4	NOWNR	Address of this pointer itself if the OWNER subparameter of the NULLIFY parameter is supplied.
176(B0)	4	NCDNM	Address of this pointer itself if the CODE subparameter of the NULLIFY parameter is supplied.
180(B4)	4	NRETN	Address of this pointer itself if the RETENTION subparameter of the NULLIFY parameter is supplied.
184(B8)	4	NAUTH	Address of this pointer itself if the AUTHORIZATION subparameter of the NULLIFY parameter is supplied.
188(BC)	4	NMDNM	Address of this pointer itself if the MODULE subparameter of the AUTHORIZATION parameter is supplied.
192(C0)	4	NSTRG	Address of this pointer itself if the STRING subparameter of the AUTHORIZATION parameter is supplied.
196(C4)	4	SHAR1	Address of information supplied through the *crossregion* subparameter of the SHAREOPTIONS parameter.
200(C8)	4	SHAR2	Address of information supplied through the *crosssystem* subparameter of the SHAREOPTIONS parameter.
204(CC)	4	GDGEM	Address of this pointer itself if the EMPTY parameter is supplied.
208(D0)	4	GDGNE	Address of this pointer itself if the NOEMPTY parameter is supplied.
212(D4)	4	GDGSC	Address of this pointer itself if the SCRATCH parameter is supplied.
216(D8)	4	GDGNS	Address of this pointer itself if the NOSCRATCH parameter is supplied.
220(DC)	4	*	Reserved—contains zeros.
224(E0)	4	EEXT	Address of information supplied through the *mname* subparameter of the EXCEPTIONEXIT parameter.
228(E4)	4	KEY	Address of this pointer itself if the KEYS parameter has been supplied.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
232(E8)	4	KEYLN	Address of information supplied through the *length* subparameter of the KEYS parameter.
236(EC)	4	KEYPS	Address of information supplied through the *offset* subparameter of the KEYS parameter.
240(F0)	4	RECSZ	Address of this pointer itself if the RECORDSIZE parameter has been supplied.
244(F4)	4	AREC	Address of information supplied through the *average* subparameter of the RECORDSIZE parameter.
248(F8)	4	MREC	Address of information supplied through the *maximum* subparameter of the RECORDSIZE parameter.
252(FC)	4	UNQK	Address of this pointer itself if the UNIQUEKEY parameter has been supplied.
256(100)	4	NUNQK	Address of this pointer itself if the NONUNIQUEKEY parameter has been supplied.
260(104)	4	UPG	Address of this pointer itself if the UPGRADE parameter has been supplied.
264(108)	4	NUPG	Address of this pointer itself if the NOUPGRADE parameter has been supplied.
268(10C)	4	UPD	Address of this pointer itself if the UPDATE parameter has been supplied.
272(110)	4	NUPD	Address of this pointer itself if the NOUPDATE parameter has been supplied.
276(114)	4	NEEXT	Address of this pointer itself if the EXCEPTIONEXIT subparameter of the NULLIFY parameter has been supplied.
280(118)	4	STAGE	Address of this pointer itself if the STAGE parameter is supplied.
284(11C)	4	BIND	Address of this pointer itself if the BIND parameter is supplied.
288(120)	4	CYLF	Address of this pointer itself if the CYLINDERFAULT parameter is supplied.
292(124)	4	NSTGW	Address of this pointer itself if the NODESTAGEWAIT parameter is supplied.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
296(128)	4	STGW	Address of this pointer itself if the DESTAGEWAIT parameter is supplied.
300(12C)	4	BUFND	Address of information supplied through the BUFND parameter.
304(130)	4	BUFNI	Address of information supplied through the BUFNI parameter.
308(134)	4	STRNO	Address of information supplied through the STRNO parameter.

BINDDATA FDT

Offset	Content	
0 (0)	BIND	DATA
8 (8)	↑ ESTABLISH	↑ TERMINATE
16 (10)	↑ FILE	↑ VOLUME
24 (18)	↑ UNIT	↑ LOWCCHH
32 (20)	↑ HIGHCCHH	↑ DEVICE
40 (28)	↑ SUBSYSTEM	

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb—BINDDATA.
8(8)	4	EST	Address of this pointer itself if the ESTABLISH parameter has been supplied.
12(C)	4	TERM	Address of this pointer itself if the TERMINATE parameter has been supplied.
16(10)	4	FILE	Address of information supplied through the FILE parameter.
20(14)	4	VOL	Address of the information supplied through the VOLUME parameter.
24(18)	4	UNT	Address of information supplied through the UNIT parameter.
28(1C)	4	LCCHH	Address of information supplied through the LOWCCHH parameter.
32(20)	4	HCCHH	Address of information supplied through the HIGHCHH parameter.
36(24)	4	DEV	Address of this pointer itself if the DEVICE parameter has been supplied.
40(28)	4	SSYS	Address of this pointer itself if the SUBSYSTEM parameter has been supplied.

BLDINDEX FDT

Offset	Content	
0 (0)	B L D I	N D E X
8 (8)	↑ INFILE	0
16 (10)	↑ OUTFILE	0
24 (18)	↑ CATALOG	↑ WORKFILES
32 (20)	↑ dname1	↑ dname2
40 (28)	↑ EXTERNALSORT	↑ INTERNALSORT

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb—BLDINDEX.
8(8)	4	IFILE	Address of information supplied through the INFILE parameter.
12(C)	4	IDS	Address of the information supplied through the INDATASET parameter.
16(10)	4	OFILE	Address of information supplied through the OUTFILE parameter.
20(14)	4	ODS	Address of the information supplied through the OUTDATASET parameter.
24(18)	4	CAT	Address of information supplied through the *catname/password* subparameter of the CATALOG parameter.
28(1C)	4	WFILE	Address of this pointer itself if the WORKFILES parameter has been supplied.
32(20)	4	WFLE1	Address of information supplied through the *dname1* subparameter of the WORKFILES parameter.
36(24)	4	WFLE2	Address of information supplied through the *dname2* subparameter of the WORKFILES parameter.
40(28)	4	ESORT	Address of this pointer itself if the EXTERNALSORT parameter has been supplied.
44(2C)	4	ISORT	Address of this pointer itself if the INTERNALSORT parameter has been supplied.

CHKLIST FDT

Offset	Content	
0 (0)	CHKL	ISTB
8 (8)	↑ INFILE	↑ OUTFILE
16 (10)	↑ CHECKID	

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb aligned left and padded with blanks—'CHKLISTƀ'.
8(8)	4	IFILE	Address of information supplied through the INFILE parameter.
12(C)	4	OFILE	Address of information supplied through the OUTFILE parameter.
16(10)	4	CHKID	Address of information supplied through the CHECKID parameter.

CNVTCAT FDT

Offset	Content	
0 (0)	C N V T	C A T B
8 (8)	↑ INFILE/*password*	↑ INDATASET/password
16 (10)	↑ CATALOG	↑ *catname/password*
24 (18)	↑ *dname*	↑ CVOLEQUATES
32 (20)	↑ *catname*	↑ *volser*
40 (28)	↑ *catname/password*	↑ LIST
48 (30)	↑ NOLIST	↑ OUTFILE
56 (38)	↑ OUTDATASET	↑ FILE

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb aligned left and padded with blanks—CNVTCATb̸.
8(8)	4	IFILE	Address of information supplied through the INFILE/*password* parameter.
12(C)	4	IDS	Address of information supplied through the INDATASET/*password* parameter.
16(10)	4	CAT	Address of this pointer itself if the CATALOG parameter is supplied.
20(14)	4	CATNM	Address of information supplied through the *catname/password* subparameter of the CATALOG parameter.
24(18)	4	CATDN	Address of information supplied through the *dname* subparameter of the CATALOG parameter.
28(1C)	4	CVEQU	Address of a count of subparameters supplied through the CVOLEQUATES parameter.
32(20)	4	CVECNPTR	Address of information supplied through the *catname* subparameter of the CVOLEQUATES parameter.
36(24)	4	CVEVSPTR	Address of information supplied through the *volser* subparameter of the CVOLEQUATES parameter.

Offset	**Bytes and Bit Pattern**	**Field Name**	**Description: Content, Meaning, Use**
40(28)	4	MRCAT	Address of information supplied through the MASTERCATALOG parameter.
44(2C)	4	LIST	Address of this pointer itself if the LIST parameter is supplied or defaulted.
48(30)	4	NLIST	Address of this pointer itself if the NOLIST parameter is supplied.
52(34)	4	OFILE	Address of information supplied through the OUTFILE/*password* parameter.
56(38)	4	ODS	Address of information supplied through the OUTDATASET/*password* parameter.
60(3C)	4	FILE	Address of information supplied through the FILE parameter.

Define FDT

There are eight illustrations relating the pointers of the FDT to the FSR parameters in the following order: alias, cluster, generation data group, master catalog, non-VSAM data set page spaces, spaces for VSAM data sets, and user catalog.

Define ALIAS

Offset	Content	
0 (0)	DEFI	NEBB
8 (8)	↑ CATALOG	↑ catname/password
16 (10)	↑ dname	0
48 (30)	0	↑ ALIAS
80 (50)	0	↑ NAME
768 (300)	0	↑ RELATE
1352 (548)	0	

Define ALTERNATEINDEX

Offset	Content	
0 (0)	DEFI	NEBB
8 (8)	↑ CATALOG	↑ catname/password
16 (10)	↑ dname	0
32 (20)	0	↑ DATA
40 (28)	↑ INDEX	0
56 (38)	0	↑ ALTERNATEINDEX
88 (58)	0	↑ NAME
96 (60)	↑ NAME	0
128(80)	↑ MODEL	↑ entryname/password
136(88)	↑ catname/password	↑ dname
144(90)	↑ MODEL	↑ entryname/password
152(98)	↑ catname/password	↑ dname
160(A0)	0	0
168(A8)	↑ MASTERPW	↑ MASTERPW
176(B0)	0	0
184(B8)	↑ CONTROLPW	↑ CONTROLPW
192(C0)	0	0
200(C8)	↑ UPDATEPW	↑ UPDATEPW
208(D0)	0	0
216(D8)	↑ READPW	↑ READPW
224(E0)	0	0
232(E8)	↑ CODE	↑ CODE
240(F0)	0	0
248(F8)	↑ ATTEMPTS	↑ ATTEMPTS
256(100)	0	0
272(110)	↑ AUTHORIZATION	↑ entrypoint
280 (118)	↑ string	↑ AUTHORIZATION

Offset	Content	
288 (120)	↑ entrypoint	↑ string
296 (128)	0	0
304 (130)	0	0
312 (138)	0	0
320 (140)	↑ OWNER	↑ OWNER
328 (148)	0	0
336 (150)	0	↑ SHAREOPTIONS
344 (158)	↑ crossregion	↑ crosssystem
352 (160)	↑ SHAREOPTIONS	↑ crossregion
360 (168)	↑ crosssystem	0
368 (170)	0	↑ ERASE
376 (178)	↑ NOERASE	0
384 (180)	0	0
392 (188)	↑ KEYS	↑ length
400 (190)	↑ position	0
408 (198)	0	↑ REPLICATE
416 (1A0)	↑ NOREPLICATE	0
424 (1A8)	0	↑ IMBED
432 (1B0)	↑ NOIMBED	0
440 (1B8)	0	0
448 (1C0)	↑ FILE	↑ FILE
456 (1C8)	0	0
472 (1D8)	↑ VOLUMES	↑ VOLUMES
480 (1E0)	0	0
488 (1E8)	0	↑ KEYRANGES
496 (1F0)	↑ lowkey	↑ highkey
512 (200)	0	0

Offset	Content	
520 (208)	↑ORDERED	↑UNORDERED
528 (210)	↑ORDERED	↑UNORDERED
536 (218)	0	↑SUBALLOCATION
544 (220)	↑SUBALLOCATION	0
552 (228)	↑UNIQUE	↑UNIQUE
560 (230)	0	0
576 (240)	0	0
584 (248)	0	0
600 (258)	0	↑TRACKS
608 (260)	↑CYLINDERS	↑RECORDS
616 (268)	↑primary	↑secondary
624 (270)	↑TRACKS	↑CYLINDERS
632 (278)	↑RECORDS	↑primary
640 (280)	↑secondary	0
664 (298)	↑RECORDSIZE	↑average
672 (2A0)	↑maximum	0
680 (2A8)	0	0
688 (2B0)	0	↑WRITECHECK
696 (2B8)	↑NOWRITECHECK	↑WRITECHECK
704 (2C0)	↑NOWRITECHECK	0
712 (2C8)	0	↑SPEED
720 (2D0)	↑RECOVERY	0
728 (2D8)	0	0
736 (2E0)	↑FREESPACE	↑cipercent
744 (2E8)	↑capercent	0
752 (2F0)	0	↑BUFFERSPACE
760 (2F8)	0	↑CONTROLINTERVALSIZE
768 (300)	↑CONTROLINTERVALSIZE	0

Offset	Content	
1024 (400)	0	0
1032 (408)	0	0
1048 (418)	0	0
1056 (420)	↑ EXCEPTIONEXIT	↑ EXCEPTIONEXIT
1064 (428)	↑ NUMBERED	↑ REUSE
1072 (430)	↑ REUSE	↑ REUSE
1080 (438)	↑ NOREUSE	↑ NOREUSE
1088 (440)	↑ NOREUSE	↑ SPANNED
1096 (448)	↑ SPANNED	↑ NONSPANNED
1104 (450)	↑ NONSPANNED	0
1208 (4B8)	0	0
1216 (4C0)	0	0
1248 (4E0)	0	0
1256 (4E8)	0	0
1288 (508)	0	0
1296 (510)	0	0
1320 (528)	0	↑ primary
1328 (530)	↑ secondary	↑ primary
1336 (538)	↑ secondary	↑ primary
1344 (540)	↑ secondary	↑ primary
1352 (548)	↑ secondary	↑ primary
1360 (550)	↑ MODEL	↑ entryname/password
1368 (558)	↑ catname/password	↑ dname
1376 (560)	↑ MASTERPW	↑ CONTROLPW
1384 (568)	↑ UPDATEPW	↑ READPW
1392 (570)	↑ CODE	↑ ATTEMPTS
1400 (578)	↑ AUTHORIZATION	↑ entrypoint

Offset	Content	
1408 (580)	↑string	↑TO
1416 (588)	↑FOR	↑OWNER
1424 (590)	↑SHAREOPTIONS	↑crosspartition
1432 (598)	↑crosssystem	↑ERASE
1440 (5A0)	↑NOERASE	↑KEYS
1448 (5A8)	↑length	↑offset
1456 (5B0)	↑REPLICATE	↑NOREPLICATE
1464 (5B8)	↑IMBED	↑NOIMBED
1472 (5C0)	↑FILE	↑VOLUMES
1480 (5C8)	↑KEYRANGES	↑lowkey
1488 (5D0)	↑highkey	↑ORDERED
1496 (5D8)	↑UNORDERED	↑SUBALLOCATION
1504 (5E0)	↑UNIQUE	↑TRACKS
1512 (5E8)	↑primary	↑secondary
1520 (5F0)	↑CYLINDERS	↑primary
1528 (5F8)	↑secondary	↑RECORDS
1536 (600)	↑primary	↑secondary
1544 (608)	↑RECORDSIZE	↑average
1552 (610)	↑maximum	↑WRITECHECK
1560 (618)	↑NOWRITECHECK	↑SPEED
1568 (620)	↑RECOVERY	↑FREESPACE
1576 (628)	↑cipercent	↑capercent
1584 (630)	↑BUFFERSPACE	↑CONTROLINTERVALSIZE
1592 (638)	↑RELATE	↑EXCEPTIONEXIT
1600 (640)	↑REUSE	↑NOREUSE
1608 (648)	↑UNIQUEKEY	↑NOUNIQUEKEY
1616 (650)	↑UNIQUEKEY	↑NOUNIQUEKEY
1624 (658)	↑UPGRADE	↑NOUPGRADE

Offset	Content	
1728 (6C0)	0	↑ STAGE
1736 (6C8)	↑ STAGE	↑ STAGE
1744 (6D0)	0	↑ BIND
1752 (6D8)	↑ BIND	↑ BIND
1760 (6E0)	0	↑ CYLINDERFAULT
1768 (6E8)	↑ CYLINDERFAULT	↑ CYLINDERFAULT
1784 (6F8)	0	↑ NODESTAGEWAIT
1792 (700)	↑ NODESTAGEWAIT	↑ NODESTAGEWAIT
1808 (710)	0	↑ DESTAGEWAIT
1816 (718)	↑ DESTAGEWAIT	↑ DESTAGEWAIT
1912 (778)	↑ RECATALOG	↑ NORECATALOG

Define CLUSTER

Offset	Content	
0 (0)	D E F I	N E B B
8 (8)	↑ CATALOG	↑ catname/password
16 (10)	↑ dname	0
24 (18)	0	↑ CLUSTER
32 (20)	0	↑ DATA
40 (28)	↑ INDEX	0
72 (48)	↑ NAME	0
88 (58)	0	↑ NAME
96 (60)	↑ NAME	↑ INDEXED
104 (68)	↑ NONINDEXED	0
112 (70)	↑ MODEL	↑ entryname/password
120 (78)	↑ catname/password	↑ dname
128 (80)	↑ MODEL	↑ entryname/password
136 (88)	↑ catname/password	↑ dname
144 (90)	↑ MODEL	↑ entryname/password
152 (98)	↑ catname/password	↑ dname
160 (A0)	0	↑ MASTERPW
168 (A8)	↑ MASTERPW	↑ MASTERPW
176 (B0)	0	↑ CONTROLPW
184 (B8)	↑ CONTROLPW	↑ CONTROLPW
192 (C0)	0	↑ UPDATEPW
200 (C8)	↑ UPDATEPW	↑ UPDATEPW
208 (D0)	0	↑ READPW
216 (D8)	↑ READPW	↑ READPW
224 (E0)	0	↑ CODE
232 (E8)	↑ CODE	↑ CODE
240 (F0)	0	↑ ATTEMPTS
248 (F8)	↑ ATTEMPTS	↑ ATTEMPTS

Offset	Content	
256 (100)	0	↑ AUTHORIZATION
272 (110)	↑ AUTHORIZATION	↑ entrypoint
280 (118)	↑ string	↑ AUTHORIZATION
288 (120)	↑ entrypoint	↑ string
296 (128)	0	↑ TO
304 (130)	0	↑ FOR
312 (138)	0	↑ OWNER
320 (140)	↑ OWNER	↑ OWNER
328 (148)	↑ SHAREOPTIONS	↑ crossregion
336 (150)	↑ crosssystem	↑ SHAREOPTIONS
344 (158)	↑ crossregion	↑ crosssystem
352 (160)	↑ SHAREOPTIONS	↑ crossregion
360 (168)	↑ crosssystem	↑ ERASE
368 (170)	↑ NOERASE	↑ ERASE
376 (178)	↑ NOERASE	↑ KEYS
384 (180)	↑ length	↑ position
392 (188)	↑ KEYS	↑ length
400 (190)	↑ position	↑ REPLICATE
408 (198)	↑ NOREPLICATE	↑ REPLICATE
416 (1A0)	↑ NOREPLICATE	↑ IMBED
424 (1A8)	↑ NOIMBED	↑ IMBED
432 (1B0)	↑ NOIMBED	0
440 (1B8)	↑ FILE	0
448 (1C0)	↑ FILE	↑ FILE
456 (1C8)	0	↑ VOLUMES
472 (1D8)	↑ VOLUMES	↑ VOLUMES
480 (1E0)	↑ KEYRANGES	↑ lowkey
488 (1E8)	↑ highkey	↑ KEYRANGES

Offset	Content	
496 (1F0)	↑ lowkey	↑ highkey
512 (200)	↑ ORDERED	↑ UNORDERED
520 (208)	↑ ORDERED	↑ UNORDERED
528 (210)	↑ ORDERED	↑ UNORDERED
536 (218)	↑ SUBALLOCATION	↑ SUBALLOCATION
544 (220)	↑ SUBALLOCATION	↑ UNIQUE
552 (228)	↑ UNIQUE	↑ UNIQUE
560 (230)	0	↑ TRACKS
576 (240)	↑ CYLINDERS	0
584 (248)	0	↑ RECORDS
600 (258)	0	↑ TRACKS
608 (260)	↑ CYLINDERS	↑ RECORDS
616 (268)	↑ primary	↑ secondary
624 (270)	↑ TRACKS	↑ CYLINDERS
632 (278)	↑ RECORDS	↑ primary
640 (280)	↑ secondary	0
648 (288)	↑ RECORDSIZE	0
656 (290)	↑ average	↑ maximum
664 (298)	↑ RECORDSIZE	↑ average
672 (2A0)	↑ maximum	0
680 (2A8)	↑ WRITECHECK	0
688 (2B0)	↑ NOWRITECHECK	↑ WRITECHECK
696 (2B8)	↑ NOWRITECHECK	↑ WRITECHECK
704 (2C0)	↑ NOWRITECHECK	↑ SPEED
712 (2C8)	↑ RECOVERY	↑ SPEED
720 (2D0)	↑ RECOVERY	↑ FREESPACE
728 (2D8)	↑ cipercent	↑ capercent
736 (2E0)	↑ FREESPACE	↑ cipercent

Offset	Content	
744 (2E8)	↑capercent	0
752 (2F0)	↑BUFFERSPACE	↑BUFFERSPACE
760 (2F8)	↑CONTROLINTERVALSIZE	↑CONTROLINTERVALSIZE
768 (300)	↑CONTROLINTERVALSIZE	0
1024 (400)	0	↑entrypoint
1032 (408)	↑string	0
1048 (418)	0	↑EXCEPTIONEXIT
1056 (420)	↑EXCEPTIONEXIT	↑EXCEPTIONEXIT
1064 (428)	↑NUMBERED	↑REUSE
1072 (430)	↑REUSE	↑REUSE
1080 (438)	↑NOREUSE	↑NOREUSE
1088 (440)	↑NOREUSE	↑SPANNED
1096 (448)	↑SPANNED	↑NONSPANNED
1104 (450)	↑NONSPANNED	0
1208 (4B8)	0	↑primary
1216 (4C0)	↑secondary	0
1248 (4E0)	0	↑primary
1256 (4E8)	↑secondary	0
1288 (508)	0	↑primary
1296 (510)	↑secondary	0
1320 (528)	0	↑primary
1328 (530)	↑secondary	↑primary
1336 (538)	↑secondary	↑primary
1344 (540)	↑secondary	↑primary
1352 (548)	↑secondary	0
1728 (6C0)	↑STAGE	0
1736 (6C8)	↑STAGE	↑STAGE
1744 (6D0)	↑BIND	0

Offset	Content	
1752 (6D8)	↑BIND	↑BIND
1760 (6E0)	↑CYLINDERFAULT	0
1768 (6E8)	↑CYLINDERFAULT	↑CYLINDERFAULT
1784 (6F8)	↑NODESTAGEWAIT	0
1784 (6F8)	↑NODESTAGEWAIT	0
1792 (700)	↑NODESTAGEWAIT	↑NODESTAGEWAIT
1808 (710)	↑DESTAGEWAIT	0
1816 (718)	↑DESTAGEWAIT	↑DESTAGEWAIT
1880 (758)	↑RECATALOG	↑NORECATALOG

Define GENERATIONDATAGROUP

Offset	Content	
0 (0)	DEFI	NEBB
8 (8)	↑CATALOG	↑catname/password
16 (10)	↑dname	0
56 (38)	↑GENERATIONDATAGROUP	0
88 (58)	↑NAME	0
776 (308)	↑EMPTY	↑NOEMPTY
784 (310)	↑LIMIT	↑SCRATCH
792 (318)	↑NOSCRATCH	0
1352 (548)	0	0
1824 (720)	0	↑TO
1832 (728)	0	↑FOR
1840 (730)	0	↑OWNER

Define MASTERCATALOG

Offset	Content	
0 (0)	D E F I	N E B B
8 (8)	↑ CATALOG	↑ catname/password
16 (10)	↑ dname	↑ MASTERCATALOG
32 (20)	0	↑ DATA
40 (28)	↑ INDEX	0
64 (40)	0	↑ NAME
88 (58)	0	↑ NAME
96 (60)	↑ NAME	0
160 (A0)	↑ MASTERPW	0
176 (B0)	↑ CONTROLPW	0
192 (C0)	↑ UPDATEPW	0
208 (D0)	↑ READPW	0
224 (E0)	↑ CODE	0
240 (F0)	↑ ATTEMPTS	0
256 (100)	↑ AUTHORIZATION	0
264 (108)	↑ entrypoint	↑ string
296 (128)	↑ TO	0
304 (130)	↑ FOR	0
312 (138)	↑ OWNER	0
408 (198)	0	↑ REPLICATE
416 (1A0)	↑ NOREPLICATE	0
424 (1A8)	0	↑ IMBED
432 (1B0)	↑ NOIMBED	0
448 (1C0)	0	↑ FILE
456 (1C8)	↑ VOLUMES	0
560 (230)	↑ TRACKS	0
568 (238)	0	↑ CYLINDERS
584 (248)	↑ RECORDS	0

Offset	Content	
600 (258)	0	↑ TRACKS
608 (260)	↑ CYLINDERS	↑ RECORDS
616 (268)	↑ primary	↑ secondary
624 (270)	↑ TRACKS	↑ CYLINDERS
632 (278)	↑ RECORDS	↑ primary
640 (280)	↑ secondary	0
664 (298)	↑ RECORDSIZE	↑ average
672 (2A0)	↑ maximum	↑ WRITECHECK
680 (2A8)	0	↑ NOWRITECHECK
688 (2B0)	0	↑ WRITECHECK
696 (2B8)	↑ NOWRITECHECK	↑ WRITECHECK
704 (2C0)	↑ NOWRITECHECK	0
736 (2E0)	↑ FREESPACE	↑ cipercent
744 (2E8)	↑ capercent	↑ BUFFERSPACE
752 (2F0)	0	↑ BUFFERSPACE
760 (2F8)	0	↑ CONTROLINTERVALSIZE
768 (300)	↑ CONTROLINTERVALSIZE	0
1104 (450)	0	↑ RECOVERABLE
1112 (458)	0	↑ RECOVERABLE
1120 (460)	↑ NOTRECOVERABLE	0
1128 (468)	↑ NOTRECOVERABLE	0
1200 (4B0)	0	↑ primary
1208 (4B8)	↑ secondary	0
1240 (4D8)	0	↑ primary
1248 (4E0)	↑ secondary	0
1280 (500)	0	↑ primary
1288 (508)	↑ secondary	0
1320 (528)	0	↑ primary

Offset	Content	
1328 (530)	↑ secondary	↑ primary
1336 (538)	↑ secondary	↑ primary
1344 (540)	↑ secondary	↑ primary
1352 (548)	↑ secondary	0
1776 (6F0)	↑ NODESTAGEWAIT	0
1800 (708)	↑ DESTAGEWAIT	0
1848 (738)	↑ BUFND	0
1856 (740)	↑ BUFND	↑ BUFNI
1864 (748)	0	↑ BUFNI
1872 (750)	↑ CONTROLINTERVALSIZE	0
1888 (760)	↑ FREESPACE	0
1896 (768)	↑ VSAMCATALOG	0
1904 (770)	↑ IMBED	0
1928 (788)	↑ NOIMBED	0
1936 (790)	↑ NOREPLICATE	0
1944 (798)	↑ RECORDSIZE	0
1952 (7A0)	↑ REPLICATE	0
1960 (7A8)	↑ SHAREOPTIONS	0
1968 (7B0)	↑ STRNO	0
1976 (7B8)	↑ ICFCATALOG	0
1984 (7C0)	↑ cipercent	0
1992 (7C8)	↑ capercent	0
2000 (7D0)	↑ crossregion	0
2008 (708)	↑ crosssystem	0
2016 (7E0)	↑ average	0
2024 (7E8)	↑ maximum	0

Define NON-VSAM

Offset	Content	
0 (0)	DEFI	NEBB
8 (8)	↑ CATALOG	↑ catname/password
16 (10)	↑ dname	0
48 (30)	↑ NONVSAM	0
80 (50)	↑ NAME	0
464 (1D0)	0	↑ VOLUMES
504 (1F8)	↑ DEVICETYPES	↑ FILESEQUENCENUMBER
1352 (548)	0	0
1824 (720)	↑ TO	0
1832 (728)	↑ FOR	0
1840 (730)	↑ OWNER	0

Define PAGESPACE

Offset	Content	
0 (0)	DEFI	NEBB
8 (8)	↑ CATALOG	↑ catname/password
16 (10)	↑ dname	0
32 (20)	↑ PAGESPACE	0
72 (48)	0	↑ NAME
888 (378)	↑ MODEL	↑ entryname/password
896 (380)	↑ catname/password	↑ dname
904 (388)	↑ MASTERPW	↑ CONTROLPW
912 (390)	↑ UPDATEPW	↑ READPW
920 (398)	↑ CODE	↑ ATTEMPTS
928 (3A0)	↑ AUTHORIZATION	↑ entrypoint
936 (3A8)	↑ string	↑ TO
944 (3B0)	↑ FOR	↑ OWNER
960 (3B8)	0	↑ ERASE
968 (3C8)	↑ NOERASE	↑ FILE
976 (3D0)	↑ VOLUMES	↑ SUBALLOCATION
984 (3D8)	↑ UNIQUE	↑ TRACKS
992 (3E0)	↑ CYLINDERS	↑ RECORDS
1000 (3E8)	↑ SWAP	↑ NOSWAP
1232 (4D0)	0	↑ primary
1240 (4D8)	↑ secondary	0
1264 (4F0)	0	↑ primary
1272 (4F8)	↑ secondary	0
1312 (520)	0	↑ primary
1320 (528)	↑ secondary	0
1352 (548)	0	

Define PATH

Offset	Content	
0 (0)	DEFI	NEBB
8 (8)	↑ CATALOG	↑ catname/password
16 (10)	↑ dname	0
64 (40)	↑ PATH	0
1640 (668)	0	↑ NAME
1648 (570)	↑ MODEL	↑ entryname/password
1656 (578)	↑ catname/password	↑ dname
1664 (680)	↑ MASTERPW	↑ CONTROLPW
1672 (688)	↑ UPDATEPW	↑ READPW
1680 (690)	↑ CODE	↑ ATTEMPTS
1688 (698)	↑ AUTHORIZATION	↑ entrypoint
1696 (6A0)	↑ string	↑ TO
1704 (6A8)	↑ FOR	↑ OWNER
1712 (6B0)	↑ FILE	↑ UPDATE
1720 (6B8)	↑ NOUPDATE	↑ PATHENTRY
1920 (780)	↑ RECATALOG	↑ NORECATALOG

Define SPACE

Offset	Content	
0 (0)	D E F I	N E B B
8 (8)	↑ CATALOG	↑ catname/password
16 (10)	↑ dname	0
40 (28)	0	↑ SPACE
440 (1B8)	0	↑ FILE
464 (1BE)	↑ VOLUMES	0
568 (238)	↑ TRACKS	0
576 (240)	0	↑ CYLINDERS
592 (250)	↑ RECORDS	0
640 (280)	0	↑ CANDIDATE
648 (288)	0	↑ RECORDSIZE
1008 (3F0)	↑ average	↑ maximum
1216 (4C0)	0	↑ primary
1224 (4C8)	↑ secondary	0
1272 (4F8)	0	↑ primary
1280 (500)	↑ secondary	0
1296 (510)	0	↑ primary
1304 (518)	↑ secondary	0
1352 (548)	0	0

Define USERCATALOG

Offset	Content	
0 (0)	DEFI	NEBB
8 (8)	↑ CATALOG	↑ catname/password
16 (10)	↑ dname	0
24 (18)	↑ USERCATALOG	0
32 (20)	0	↑ DATA
40 (28)	↑ INDEX	0
104 (68)	0	↑ MODEL
408 (198)	0	↑ REPLICATE
416 (1A0)	↑ NOREPLICATE	0
424 (1A8)	0	↑ IMBED
432 (1B0)	↑ NOIMBED	0
600 (258)	0	↑ TRACKS
608 (260)	↑ CYLINDERS	↑ RECORDS
616 (268)	↑ primary	↑ secondary
624 (270)	↑ TRACKS	↑ CYLINDERS
632 (278)	↑ RECORDS	↑ primary
640 (280)	↑ secondary	0
664 (298)	↑ RECORDSIZE	↑ average
672 (2A0)	↑ maximum	0
688 (2B0)	0	↑ WRITECHECK
696 (2B8)	↑ NOWRITECHECK	↑ WRITECHECK
704 (2C0)	↑ NOWRITECHECK	0
736 (2E0)	↑ FREESPACE	↑ cipercent
744 (2E8)	↑ capercent	0
752 (2F0)	0	↑ BUFFERSPACE
760 (2F8)	0	↑ CONTROLINTERVALSIZE
768 (300)	↑ CONTROLINTERVALSIZE	0
792 (318)	0	↑ NAME

Offset	Content	
800 (320)	↑ MASTERPW	↑ CONTROLPW
808 (328)	↑ UPDATEPW	↑ READPW
816 (330)	↑ CODE	↑ ATTEMPTS
824 (338)	↑ AUTHORIZATION	↑ entrypoint
832 (340)	↑ string	↑ TO
840 (348)	↑ FOR	↑ OWNER
848 (350)	↑ FILE	↑ VOLUMES
856 (358)	↑ TRACKS	↑ CYLINDERS
864 (360)	↑ RECORDS	0
872 (368)	0	↑ WRITECHECK
880 (370)	↑ NOWRITECHECK	↑ BUFFERSPACE
1016 (3F8)	↑ entrypoint	↑ catname/password
1024 (400)	↑ dname	0
1112 (458)	↑ RECOVERABLE	↑ RECOVERABLE
1120 (460)	0	↑ NOTRECOVERABLE
1128 (468)	↑ NOTRECOVERABLE	0
1256 (4E8)	0	↑ primary
1264 (4F0)	↑ secondary	0
1304 (518)	0	↑ primary
1312 (520)	↑ secondary	0
1320 (528)	0	↑ primary
1328 (530)	↑ secondary	↑ primary
1336 (538)	↑ secondary	↑ primary
1344 (540)	↑ secondary	↑ primary
1352 (548)	↑ secondary	0
1776 (6F0)	0	↑ NODESTAGEWAIT
1792 (700)	↑ NODESTAGEWAIT	↑ NODESTAGEWAIT
1800 (708)	0	↑ DESTAGEWAIT

Offset	Content	
1816 (718)	↑DESTAGEWAIT	↑ DESTAGEWAIT
1848 (738)	0	↑ BUFND
1856 (740)	↑BUFND	0
1864 (748)	↑BUFNI	↑ BUFNI
1872 (750)	0	↑ CONTROLINTERVALSIZE
1888 (760)	0	↑ FREESPACE
1896 (768)	0	↑ VSAMCATALOG
1904 (770)	0	↑ IMBED
1928 (778)	0	↑ NOIMBED
1936 (790)	0	↑ NOREPLICATE
1944 (798)	0	↑ RECORDSIZE
1952 (7A0)	0	↑ REPLICATE
1960 (7A8)	0	↑ SHAREOPTIONS
1968 (7B0)	0	↑ STRNO
1976 (7B8)	0	↑ ICFCATALOG
1984 (7C0)	0	↑ cipercent
1992 (7C8)	0	↑ capercent
2000 (7D0)	0	↑ crossregion
2008 (7D8)	0	↑ crosssystem
2016 (7E0)	0	↑ average
2024 (7E8)	0	↑ maximum

DEFINE FDT Description

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb aligned left and padded with blanks—DEFINEƀ ƀ.
8(8)	4	CAT	Address of this pointer itself if the CATALOG parameter has been supplied.
12(C)	4	CATLG	Address of information supplied through the *catname/password* subparameter of the CATALOG parameter.
16(10)	4	CATDN	Address of information supplied through the *dname* subparameter of the CATALOG parameter.
20(14)	4	MCAT	Address of this pointer itself if the MASTERCATALOG parameter has been supplied—that is, if you are defining a master catalog.
24(18)	4	UCAT	Address of this pointer itself if the USERCATALOG parameter is supplied—that is, if you are defining a USERCATALOG.
28(1C)	4	CLST	Address of this pointer itself if the CLUSTER parameter is supplied—that is, if you are defining a CLUSTER.
32(20)	4	PGSP	Address of this pointer itself if the PAGESPACE parameter is supplied—that is, if you are defining a CLUSTER.
36(24)	4	DATAA	Address of this pointer itself if the DATA parameter is supplied.
40(28)	4	INDEX	Address of this pointer itself if the INDEX parameter is supplied.
44(2C)	4	SPACE	Address of this pointer itself if the SPACE parameter is supplied—that is, if you are defining a VSAM data space.
48(30)	4	ALIEN	Address of this pointer itself if the NONVSAM parameter is supplied—that is, if you are defining a non-VSAM data set.
52(34)	4	ALIAS	Address of this pointer itself if the ALIAS parameter is supplied—that is, if you are defining an alias.
56(38)	4	GENDG	Address of this pointer itself if the GENERATIONDATA-GROUP parameter is supplied—that is, if you are defining a generation data group.
60(3C)	4	AIX	Address of this pointer itself if the ALTERNATEINDEX parameter is supplied—that is, if you are defining an alternate index.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
64(40)	4	PATH	Address of this pointer itself if the PATH parameter is supplied—that is, if you are defining a path.
68(44)	4	METRY	Address of information supplied through the NAME parameter if NAME is supplied under MASTERCATALOG.
72(48)	4	CETRY	Address of information supplied through the NAME parameter if NAME is supplied under CLUSTER.
76(4C)	4	PETRY	Address of information supplied through the NAME parameter if NAME is supplied under PAGESPACE.
80(50)	4	AETRY	Address of information supplied through the NAME parameter if NAME is supplied under NONVSAM.
84(54)	4	XETRY	Address of information supplied through the NAME parameter if NAME is supplied under ALIAS.
88(58)	4	GETRY	Address of information supplied through the NAME parameter if NAME is supplied under GENERATIONDATA GROUP.
92(5C)	4	DETRY	Address of information supplied through the NAME parameter if NAME is supplied under DATA.
96(60)	4	IETRY	Address of information supplied through the NAME parameter if NAME is supplied under INDEX.
100(64)	4	CINDX	Address of this pointer itself if INDEXED is supplied under CLUSTER.
104(68)	4	CNIDX	Address of this pointer itself if NONINDEXED is supplied under CLUSTER.
108(6C)	4	UMODL	Address of this pointer itself if MODEL is supplied under USERCATALOG.
112(70)	4	CMODL	Address of this pointer itself if the MODEL parameter is supplied under CLUSTER.
116(74)	4	CENAM	Address of information supplied through the *entryname/password* subparameter of MODEL if MODEL is supplied under CLUSTER.
120(78)	4	CMDCT	Address of information supplied through the *catname/password* subparameter of MODEL if MODEL is supplied under CLUSTER.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
124(7C)	4	CMDNM	Address of information supplied through the *dname* subparameter of MODEL if MODEL is supplied under CLUSTER.
128(80)	4	DMODL	Address of this pointer itself if the MODEL parameter is supplied under DATA.
132(84)	4	DENAM	Address of information supplied through the *entryname/password* subparameter of MODEL if MODEL is supplied under DATA.
136(88)	4	DMDCT	Address of information supplied through the *catname/password* subparameter of MODEL if MODEL is supplied under DATA.
140(8C)	4	DMDNM	Address of information supplied through the *dname* subparameter of MODEL if MODEL is supplied under DATA.
144(90)	4	IMODL	Address of this pointer itself if MODEL is supplied under INDEX.
148(94)	4	IENAM	Address of information supplied through the *entryname/password* subparameter of MODEL if MODEL is supplied under INDEX.
152(98)	4	IMDCT	Address of information supplied through the *catname/password* subparameter of MODEL if MODEL is supplied under INDEX.
156(9C)	4	IMDNM	Address of information supplied through the *dname* subparameter of MODEL if MODEL is supplied under INDEX.
160(A0)	4	MMSTR	Address of information supplied through the MASTERPW parameter if MASTERPW is supplied under MASTERCATALOG.
164(A4)	4	CMSTR	Address of information supplied through the MASTERPW parameter if MASTERPW is supplied under CLUSTER.
168(A8)	4	DMSTR	Address of information supplied through the MASTERPW parameter if MASTERPW is supplied under DATA.
172(AC)	4	IMSTR	Address of information supplied through the MASTERPW parameter if MASTERPW is supplied under INDEX.
176(B0)	4	MCINT	Address of information supplied through the CONTROLPW parameter if CONTROLPW is supplied under MASTERCATALOG.
180(B4)	4	CCINT	Address of information supplied through the CONTROLPW parameter if CONTROLPW is supplied under CLUSTER.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
184(B8)	4	DCINT	Address of information supplied through the CONTROLPW parameter if CONTROLPW is supplied under DATA.
188(BC)	4	ICINT	Address of information supplied through the CONTROLPW parameter if CONTROLPW is supplied under INDEX.
192(C0)	4	MUPDT	Address of information supplied through the UPDATEPW if UPDATEPW is supplied under MASTERCATALOG.
196(C4)	4	CUPDT	Address of information supplied through the UPDATEPW if UPDATEPW is supplied under CLUSTER.
200(C8)	4	DUPDT	Address of information supplied through the UPDATEPW if UPDATEPW is supplied under DATA.
204(CC)	4	IUPDT	Address of information supplied through the UPDATEPW if UPDATEPW is supplied under INDEX.
208(D0)	4	MREAD	Address of information supplied through the READPW parameter if READPW is supplied under MASTERCATALOG.
212(D4)	4	CREAD	Address of information supplied through the READPW parameter if READPW is supplied under CLUSTER.
216(D8)	4	DREAD	Address of information supplied through the READPW parameter if READPW is supplied under DATA.
220(DC)	4	IREAD	Address of information supplied through the READPW parameter if READPW is supplied under INDEX.
224(E0)	4	MCODE	Address of information supplied through the CODE parameter if CODE is supplied under MASTERCATALOG.
228(E4)	4	CCODE	Address of information supplied through the CODE parameter if CODE is supplied under CLUSTER.
232(E8)	4	DCODE	Address of information supplied through the CODE parameter if CODE is supplied under DATA.
236(EC)	4	ICODE	Address of information supplied through the CODE parameter if CODE is supplied under INDEX.
240(F0)	4	MATTP	Address of information supplied through the ATTEMPTS parameter if ATTEMPTS is supplied under MASTERCATALOG.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
244(F4)	4	CATTP	Address of information supplied through the ATTEMPTS parameter if ATTEMPTS is supplied under CLUSTER.
248(F8)	4	DATTP	Address of information supplied through the ATTEMPTS parameter if ATTEMPTS is supplied under DATA.
252(FC)	4	IATTP	Address of information supplied through the ATTEMPTS parameter if ATTEMPTS is supplied under INDEX.
256(100)	4	MAUTH	Address of this pointer itself if the AUTHORIZATION parameter is supplied under MASTERCATALOG.
260(104)	4	CAUTH	Address of this pointer itself if the AUTHORIZATION parameter is supplied under CLUSTER.
264(108)	4	MEPNM	Address of information supplied through the *entrypoint* subparameter of AUTHORIZATION if AUTHORIZATION is supplied under MASTERCATALOG.
268(10C)	4	MSTRG	Address of information supplied through the *string* subparameter of AUTHORIZATION if AUTHORIZATION is supplied under MASTERCATALOG.
272(110)	4	DAUTH	Address of this pointer itself if the AUTHORIZATION parameter is supplied under DATA.
276(114)	4	DEPNM	Address of information supplied through the *entrypoint* subparameter of AUTHORIZATION if AUTHORIZATION is supplied under DATA.
280(118)	4	DSTRG	Address of information supplied through the *string* subparameter of AUTHORIZATION if AUTHORIZATION is supplied under DATA.
284(11C)	4	IAUTH	Address of this pointer itself if the AUTHORIZATION parameter is supplied under INDEX.
288(120)	4	IEPNM	Address of information supplied through the *entrypoint* subparameter of AUTHORIZATION if the AUTHORIZATION parameter is supplied under INDEX.
292(124)	4	ISTRG	Address of information supplied through the *string* subparameter of AUTHORIZATION if the AUTHORIZATION parameter is supplied under INDEX.
296(128)	4	MTO	Address of information supplied through the TO parameter if TO is supplied under MASTERCATALOG.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
300(12C)	4	CTO	Address of information supplied through the TO parameter if TO is supplied under CLUSTER.
304(130)	4	MFOR	Address of information supplied through the FOR parameter if FOR is supplied under MASTERCATALOG.
308(134)	4	CFOR	Address of information supplied through the FOR parameter if FOR is supplied under CLUSTER.
312(138)	4	MOWNR	Address of information supplied through the OWNER parameter if OWNER is supplied under MASTERCATALOG.
316(13C)	4	COWNR	Address of information supplied through the OWNER parameter if OWNER is supplied under CLUSTER.
320(140)	4	DOWNR	Address of information supplied through the OWNER parameter if OWNER is supplied under DATA.
324(144)	4	IOWNR	Address of information supplied through the OWNER parameter if OWNER is supplied under INDEX.
328(148)	4	CSHAR	Address of this pointer itself if the SHAREOPTIONS parameter is supplied under CLUSTER.
332(14C)	4	CSHR1	Address of information supplied through the *crossregion* subparameter of SHAREOPTIONS if SHAREOPTIONS is supplied under CLUSTER.
336(150)	4	CSHR2	Address of information supplied through the *crosssystem* subparameter of SHAREOPTIONS if SHAREOPTIONS is supplied under CLUSTER.
340(154)	4	DSHAR	Address of this pointer itself if the SHAREOPTIONS parameter is supplied under DATA.
344(158)	4	DSHR1	Address of information supplied through the *crossregion* subparameter of SHAREOPTIONS if SHAREOPTIONS is supplied under DATA.
348(15C)	4	DSHR2	Address of information supplied through the *crossregion* subparameter of SHAREOPTIONS if SHAREOPTIONS is supplied under DATA.
352(160)	4	ISHAR	Address of this pointer itself if the SHAREOPTIONS parameter is supplied under INDEX.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
356(164)	4	ISHR1	Address of information supplied through the *crossregion* subparameter of SHAREOPTIONS if SHAREOPTIONS is supplied under INDEX.
360(168)	4	ISHR2	Address of information supplied through the *crosssystem* subparameter of SHAREOPTIONS if SHAREOPTIONS is supplied under INDEX.
364(16C)	4	CERAS	Address of this pointer itself if the ERASE parameter is supplied under CLUSTER.
368(170)	4	CNERS	Address of this pointer itself if the NOERASE parameter is supplied under CLUSTER.
372(174)	4	DERAS	Address of this pointer itself if the ERASE parameter is supplied under DATA.
376(178)	4	DNERS	Address of this pointer itself if the NOERASE parameter is supplied under DATA.
380(17C)	4	CKEY	Address of this pointer itself if the KEYS parameter is supplied under CLUSTER.
384(180)	4	CKYLN	Address of information supplied through the *length* subparameter of KEYS if KEYS is supplied under CLUSTER.
388(184)	4	CKYPS	Address of information supplied through the *position* subparameter of KEYS if KEYS is supplied under CLUSTER.
392(188)	4	DKEY	Address of this pointer itself if the KEYS parameter is supplied under DATA.
396(18C)	4	DKYLN	Address of information supplied through the *length* subparameter of KEYS if KEYS is supplied under DATA.
400(190)	4	DKYPS	Address of information supplied through the *position* subparameter of KEYS if KEYS is supplied under DATA.
404(194)	4	CREPL	Address of this pointer itself if the REPLICATE parameter is supplied under CLUSTER.
408(198)	4	CNREP	Address of this pointer itself if the NOREPLICATE parameter is supplied under CLUSTER.
412(19C)	4	IREPL	Address of this pointer itself if the REPLICATE parameter is supplied under INDEX.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
416(1A0)	4	INREP	Address of this pointer itself if the NOREPLICATE parameter is supplied under INDEX.
420(1A4)	4	CIMBD	Address of this pointer itself if the IMBED parameter is supplied under CLUSTER.
424(1A8)	4	CNIBD	Address of this pointer itself if the NOIMBED parameter is supplied under CLUSTER.
428(1AC)	4	IIMBD	Address of this pointer itself if the IMBED parameter is supplied under INDEX.
432(1B0)	4	INIBD	Address of this pointer itself if the NOIMBED parameter is supplied under INDEX.
436(1B4)	4	MINDD	Address of information supplied through the FILE parameter if FILE is supplied under MASTERCATALOG.
440(1B8)	4	CINDD	Address of information supplied through the FILE parameter if FILE is supplied under CLUSTER.
444(1BC)	4	SINDD	Address of information supplied through the FILE parameter if FILE is supplied under SPACE.
448(1C0)	4	DINDD	Address of information supplied through the FILE parameter if FILE is supplied under DATA.
452(1C4)	4	IINDD	Address of information supplied through the FILE parameter if FILE is supplied under INDEX.
456(1C8)	4	MVSER	Address of information supplied through the VOLUMES parameter if VOLUMES is supplied under MASTERCATALOG.
460(1CC)	4	CVSER	Address of information supplied through the VOLUMES parameter if VOLUMES is supplied under CLUSTER.
464(1D0)	4	SVSER	Address of information supplied through the VOLUMES parameter if VOLUMES is supplied under SPACE.
468(1D4)	4	AVSER	Address of information supplied through the VOLUMES parameter if VOLUMES is supplied under NONVSAM.
472(1D8)	4	DVSER	Address of information supplied through the VOLUMES parameter if VOLUMES is supplied under DATA.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
476(1DC)	4	IVSER	Address of information supplied through the VOLUMES parameter if VOLUMES is supplied under INDEX.
480(1E0)	4	CRANG	Address of a count of subparameters supplied through the KEYRANGES parameter if KEYRANGES is supplied under CLUSTER.
484(1E4)	4	CRGLOPTR	Address of information supplied through the *lowkey* subparameter of KEYRANGES if KEYRANGES is supplied under CLUSTER.
488(1E8)	4	CRGHIPTR	Address of information supplied through the *highkey* subparameter of KEYRANGES if KEYRANGES is supplied under CLUSTER.
492(1EC)	4	DRANG	Address of a count of subparameters supplied through the KEYRANGES parameter if KEYRANGES is supplied under DATA.
496(1F0)	4	DRGLOPTR	Address of information supplied through the *lowkey* subparameter of KEYRANGES if KEYRANGES is supplied under DATA.
500(1F4)	4	DRGHIPTR	Address of information supplied through the *highkey* subparameter of KEYRANGES if KEYRANGES is supplied under DATA.
504(1F8)	4	ADEVT	Address of information supplied through the DEVICETYPES parameter if DEVICETYPES is supplied under NONVSAM.
508(1FC)	4	AFSNO	Address of information supplied through the FILESEQUENCENUMBER parameter if FILESEQUENCENUMBER is supplied under NONVSAM.
512(200)	4	CORDR	Address of this pointer itself if the ORDERED parameter is supplied under CLUSTER.
516(204)	4	CUORD	Address of this pointer itself if the UNORDERED parameter is supplied under CLUSTER.
520(208)	4	DORDR	Address of this pointer itself if the ORDERED parameter is supplied under DATA.
524(20C)	4	DUORD	Address of this pointer itself if the UNORDERED parameter is supplied under DATA.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
528(210)	4	IORDR	Address of this pointer itself if the ORDERED parameter is supplied under INDEX.
532(214)	4	IUORD	Address of this pointer itself if the UNORDERED parameter is supplied under INDEX.
536(218)	4	CSUBA	Address of this pointer itself if the SUBALLOCATION parameter is supplied under CLUSTER.
540(21C)	4	DSUBA	Address of this pointer itself if the SUBALLOCATION parameter is supplied under DATA.
544(220)	4	ISUBA	Address of this pointer itself if the SUBALLOCATION parameter is supplied under INDEX.
548(224)	4	CUNIQ	Address of this pointer itself if the UNIQUE parameter is supplied under CLUSTER.
552(228)	4	DUNIQ	Address of this pointer itself if the UNIQUE parameter is supplied under DATA.
556(22C)	4	IUNIQ	Address of this pointer itself if the UNIQUE parameter is supplied under INDEX.
560(230)	4	MTRKS	Address of this pointer itself if the TRACKS parameter is supplied under MASTERCATALOG.
564(234)	4	CTRKS	Address of this pointer itself if the TRACKS parameter is supplied under CLUSTER.
568(238)	4	STRKS	Address of this pointer itself if the TRACKS parameter is supplied under SPACE.
572(23C)	4	MCYLD	Address of this pointer itself if the CYLINDERS parameter is supplied under MASTERCATALOG.
576(240)	4	CCYLD	Address of this pointer itself if the CYLINDERS parameter is supplied under CLUSTER.
580(244)	4	SCYLD	Address of this pointer itself if the CYLINDERS parameter is supplied under SPACE.
584(248)	4	MRCDS	Address of this pointer itself if the RECORDS parameter is supplied under MASTERCATALOG.
588(24C)	4	CRCDS	Address of this pointer itself if the RECORDS parameter is supplied under CLUSTER.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
592(250)	4	SRCDS	Address of this pointer itself if the RECORDS parameter is supplied under SPACE.
596(254)	8	*	Reserved—contains zeros.
604(25C)	4	DTRKS	Address of this pointer itself if the TRACKS parameter is supplied under DATA.
608(260)	4	DCYLD	Address of this pointer itself if the CYLINDERS parameter is supplied under DATA.
612(264)	4	DRCDS	Address of this pointer itself if the RECORDS parameter is supplied under DATA.
616(268)	4	DTKPR	Address of information supplied through the *primary* subparameter of TRACKS if TRACKS is supplied under DATA.
620(26C)	4	DTKSC	Address of information supplied through the *secondary* subparameter of TRACKS if TRACKS is supplied under DATA.
624(270)	4	ITRKS	Address of this pointer itself if the TRACKS parameter is supplied under INDEX.
628(274)	4	ICYLD	Address of this pointer itself if the CYLINDERS parameter is supplied under INDEX.
632(278)	4	IRCDS	Address of this pointer itself if the RECORDS parameter is supplied under INDEX.
636(27C)	4	ITKPR	Address of information supplied through the *primary* subparameter of TRACKS if TRACKS is supplied under INDEX.
640(280)	4	ITKSC	Address of information supplied through the *secondary* subparameter of TRACKS if TRACKS is supplied under INDEX.
644(284)	4	SCAND	Address of this pointer itself if the CANDIDATE parameter is supplied under SPACE.
648(288)	4	CRSIZ	Address of this pointer itself if the RECORDSIZE parameter is supplied under CLUSTER.
652(28C)	4	SRSIZ	Address of this pointer itself if the RECORDSIZE parameter is supplied under SPACE.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
656(290)	4	CARSZ	Address of information supplied through the *average* subparameter of RECORDSIZE if RECORDSIZE is supplied under CLUSTER.
660(294)	4	CMRSZ	Address of information supplied through the *maximum* subparameter of RECORDSIZE if RECORDSIZE is supplied under CLUSTER.
664(298)	4	DRSIZ	Address of this pointer itself if the RECORDSIZE parameter is supplied under DATA.
668(29C)	4	DARSZ	Address of information supplied through the *average* subparameter of RECORDSIZE if RECORDSIZE is supplied under DATA.
672(2A0)	4	DMRSZ	Address of information supplied through the *maximum* subparameter of RECORDSIZE if RECORDSIZE is supplied under DATA.
676(2A4)	4	MWCK	Address of this pointer itself if the WRITECHECK parameter is supplied under MASTERCATALOG.
680(2A8)	4	CWCK	Address of this pointer itself if the WRITECHECK parameter is supplied under CLUSTER.
684(2AC)	4	MNWCK	Address of this pointer itself if the NOWRITECHECK parameter is supplied under MASTERCATALOG.
688(2B0)	4	CNWCK	Address of this pointer itself if the NOWRITECHECK parameter is supplied under CLUSTER.
692(2B4)	4	DWCK	Address of this pointer itself if the WRITECHECK parameter is supplied under DATA.
696(2B8)	4	DNWCK	Address of this pointer itself if the NOWRITECHECK parameter is supplied under DATA.
700(2BC)	4	IWCK	Address of this pointer itself if the WRITECHECK parameter is supplied under INDEX.
704(2C0)	4	INWCK	Address of this pointer itself if the NOWRITECHECK parameter is supplied under INDEX.
708(2C4)	4	CSPED	Address of this pointer itself if the SPEED parameter is supplied under CLUSTER.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
712(2C8)	4	CRECV	Address of this pointer itself if the RECOVERY parameter is supplied under CLUSTER.
716(2CC)	4	DSPED	Address of this pointer itself if the SPEED parameter is supplied under DATA.
720(2D0)	4	DRECV	Address of this pointer itself if the RECOVERY parameter is supplied under DATA.
724(2D4)	4	CFSPC	Address of this pointer itself if the FREESPACE parameter is supplied under CLUSTER.
728(2D8)	4	CCIFS	Address of information supplied through the *cipercent* subparameter of FREESPACE if FREESPACE is supplied under CLUSTER.
732(2DC)	4	CCAFS	Address of information supplied through the *capercent* subparameter of FREESPACE if FREESPACE is supplied under CLUSTER.
736(2E0)	4	DFSPC	Address of this pointer itself if the FREESPACE parameter is supplied under DATA.
740(2E4)	4	DCIFS	Address of information supplied through the *cipercent* subparameter of FREESPACE if FREESPACE is supplied under DATA.
744(2E8)	4	DCAFS	Address of information supplied through the *capercent* subparameter of FREESPACE if FREESPACE is supplied under DATA.
748(2EC)	4	MBFSZ	Address of information supplied through the BUFFERSPACE parameter if BUFFERSPACE is supplied under MASTERCATALOG.
752(2F0)	4	CBFSZ	Address of information supplied through the BUFFERSPACE parameter if BUFFERSPACE is supplied under CLUSTER.
756(2F4)	4	DBFSZ	Address of information supplied through the BUFFERSPACE parameter if BUFFERSPACE is supplied under DATA.
760(2F8)	4	CCINV	Address of information supplied through the CONTROLINTERVALSIZE parameter if CONTROLINTERVALSIZE is supplied under CLUSTER.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
764(2FC)	4	DCINV	Address of information supplied through the CONTROLINTERVALSIZE parameter if CONTROLINTERVALSIZE is supplied under DATA.
768(300)	4	ICINV	Address of information supplied through the CONTROLINTERVALSIZE parameter if CONTROLINTERVALSIZE is supplied under INDEX.
772(304)	4	ALREL	Address of information supplied through the RELATE parameter if RELATE is supplied under ALIAS.
776(308)	4	GENEM	Address of this pointer itself if the EMPTY parameter is supplied under GENERATIONDATA GROUP.
780(30C)	4	GENNE	Address of this pointer itself if the NOEMPTY parameter is supplied under GENERATIONDATA GROUP.
784(310)	4	GENLM	Address of information supplied through the LIMIT parameter if LIMIT is supplied under GENERATIONDATA GROUP.
788(314)	4	GENSC	Address of this pointer itself if the SCRATCH parameter is supplied under GENERATIONDATA GROUP.
792(318)	4	GENNS	Address of this pointer itself if the NOSCRATCH parameter is supplied under GENERATIONDATA GROUP.
796(31C)	4	UETRY	Address of information supplied through the NAME parameter if NAME is supplied under USERCATALOG.
800(320)	4	UMSTR	Address of information supplied through the MASTERPW parameter if MASTERPW is supplied under USERCATALOG.
804(324)	4	UCINT	Address of information supplied through the CONTROLPW parameter if CONTROLPW is supplied under USERCATALOG.
808(328)	4	UUPDT	Address of information supplied through the UPDATEPW parameter if UPDATEPW is supplied under USERCATALOG.
812(32C)	4	UREAD	Address of information supplied through the READPW parameter if READPW is supplied under USERCATALOG.
816(330)	4	UCODE	Address of information supplied through the CODE parameter if CODE is supplied under USERCATALOG.
820(334)	4	UATTP	Address of information supplied through the ATTEMPTS parameter if ATTEMPTS is supplied under USERCATALOG.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
824(338)	4	UAUTH	Address of this pointer itself if the AUTHORIZATION parameter is supplied under USERCATALOG.
828(33C)	4	UEPNM	Address of information supplied through the *entrypoint* subparameter of AUTHORIZATION if AUTHORIZATION is supplied under USERCATALOG.
832(340)	4	USTRG	Address of information supplied through the *string* subparameter of AUTHORIZATION if AUTHORIZATION is supplied under USERCATALOG.
836(344)	4	UTO	Address of information supplied through the TO parameter if TO is supplied under USERCATALOG.
840(348)	4	UFOR	Address of information supplied through the FOR parameter if FOR is supplied under USERCATALOG.
844(34C)	4	UOWNR	Address of information supplied through the OWNER parameter if OWNER is supplied under USERCATALOG.
848(350)	4	UINDD	Address of information supplied through the FILE parameter if FILE is supplied under USERCATALOG.
852(354)	4	UVSER	Address of information supplied through the VOLUMES parameter if VOLUMES is supplied under USERCATALOG.
856(358)	4	UTRKS	Address of this pointer itself if the TRACKS parameter is supplied under USERCATALOG.
860(35C)	4	UCYLD	Address of this pointer itself if the CYLINDERS parameter is supplied under USERCATALOG.
864(360)	4	URCDS	Address of this pointer itself if the RECORDS parameter is supplied under USERCATALOG.
868(364)	8	*	Reserved—contains zeros.
876(36C)	4	UWCK	Address of this pointer itself if the WRITECHECK parameter is supplied under USERCATALOG.
880(370)	4	UNWCK	Address of this pointer itself if the NOWRITECHECK parameter is supplied under USERCATALOG.
884(374)	4	UBFSZ	Address of information supplied through the BUFFERSPACE parameter if BUFFERSPACE is supplied under USERCATALOG.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
888(378)	4	PMODL	Address of this pointer itself if the MODEL parameter is supplied under PAGESPACE.
892(37C)	4	PENAM	Address of information supplied through the *entryname/password* subparameter of MODEL if MODEL is supplied under PAGESPACE.
896(380)	4	PMDCT	Address of information supplied through the *catname/password* subparameter of MODEL if MODEL is supplied under PAGESPACE.
900(384)	4	PMDNM	Address of information supplied through the *dname* subparameter of MODEL if MODEL is supplied under PAGESPACE.
904(388)	4	PMSTR	Address of information supplied through the MASTERPW parameter if MASTERPW is supplied under PAGESPACE.
908(38C)	4	PCINT	Address of information supplied through the CONTROLPW parameter if CONTROLPW is supplied under PAGESPACE.
912(390)	4	PUPDT	Address of information supplied through the UPDATEPW parameter if UPDATEPW is supplied under PAGESPACE.
916(394)	4	PREAD	Address of information supplied through the READPW parameter if READPW is supplied under PAGESPACE.
920(398)	4	PCODE	Address of information supplied through the CODE parameter if CODE is supplied under PAGESPACE.
924(39C)	4	PATTP	Address of information supplied through the ATTEMPTS parameter if ATTEMPTS is supplied under PAGESPACE.
928(3A0)	4	PAUTH	Address of this pointer itself if the AUTHORIZATION parameter is supplied under PAGESPACE.
932(3A4)	4	PEPNM	Address of information supplied through the *entrypoint* subparameter of AUTHORIZATION if AUTHORIZATION is supplied under PAGESPACE.
936(3A8)	4	PSTRG	Address of information supplied through the *string* subparameter of AUTHORIZATION if AUTHORIZATION is supplied under PAGESPACE.
940(3AC)	4	PTO	Address of information supplied through the TO parameter if TO is supplied under PAGESPACE.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
944(3B0)	4	PFOR	Address of information supplied through the FOR parameter if FOR is supplied under PAGESPACE.
948(3B4)	4	POWNR	Address of information supplied through the OWNER parameter if OWNER is supplied under PAGESPACE.
952(3B8)	12	*	Reserved—contains zeros.
964(3C4)	4	PERAS	Address of this pointer itself if ERASE parameter has been supplied under PAGESPACE.
968(3C8)	4	PNERS	Address of this pointer itself if the NOERASE parameter has been supplied under PAGESPACE.
972(3CC)	4	PINDD	Address of information supplied through the FILE parameter if FILE is supplied under PAGESPACE.
976(3D0)	4	PVSER	Address of information supplied through the VOLUMES parameter if VOLUMES is supplied under PAGESPACE.
980(3D4)	4	PSUBA	Address of this pointer itself if the SUBALLOCATION parameter is supplied under PAGESPACE.
984(3D8)	4	PUNIQ	Address of this pointer itself if the UNIQUE parameter is supplied under PAGESPACE.
988(3DC)	4	PTRKS	Address of this pointer itself if the TRACKS parameter is supplied under PAGESPACE.
992(3E0)	4	PCYLD	Address of this pointer itself if the CYLINDERS parameter is supplied under PAGESPACE.
996(3E4)	4	PRCDS	Address of this pointer itself if the RECORDS parameter is supplied under PAGESPACE.
1000(3E8)	8	*	Reserved–contains zeros.
1000(3E8)	4	PSWAP	Address of this pointer itself if the SWAP parameter is supplied under CLUSTER.
1004(3EC)	4	PNSWP	Address of this pointer itself if the NOSWAP parameter is supplied under PAGESPACE.
1008(3F0)	4	SARSZ	Address of information supplied through the *average* subparameter of RECORDSIZE if RECORDSIZE is supplied under SPACE.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
1012(3F4)	4	SMRSZ	Address of information supplied through the *maximum* subparameter of RECORDSIZE if RECORDSIZE is supplied under SPACE.
1016(3F8)	4	UENAM	Address of information supplied through the *entrypoint* subparameter of MODEL if MODEL is supplied under USERCATALOG.
1020(3FC)	4	UMDCT	Address of information supplied through the *catname/password* subparameter of MODEL if MODEL is supplied under USERCATALOG.
1024(400)	4	UMDNM	Address of information supplied through the *dname* subparameter of MODEL if MODEL is supplied under USERCATALOG.
1028(404)	4	CEPNM	Address of information supplied through the *entrypoint* subparameter of AUTHORIZATION if AUTHORIZATION is supplied under CLUSTER.
1032(408)	4	CSTRG	Address of information supplied through the *string* subparameter of AUTHORIZATION if AUTHORIZATION is supplied under CLUSTER.
1036(40C)	16	*	Reserved—contains zeros.
1052(41C)	4	CEEXT	Address of information supplied through the EXCEPTIONEXIT parameter if EXCEPTIONEXIT is supplied under CLUSTER.
1056(420)	4	DEEXT	Address of information supplied through the EXCEPTIONEXIT parameter if EXCEPTIONEXIT is supplied under DATA.
1060(424)	4	IEEXT	Address of information supplied through the EXCEPTIONEXIT parameter if EXCEPTIONEXIT is supplied under INDEX.
1068(42C)	4	CNUMD	Address of this pointer itself if NUMBERED is supplied under CLUSTER.
1072(430)	4	CRUS	Address of this pointer itself if REUSE is supplied under CLUSTER.
1076(434)	4	DRUS	Address of this pointer itself if REUSE is supplied under DATA.
1080(438)	4	IRUS	Address of this pointer itself if REUSE is supplied under INDEX.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
1084(43C)	4	CNRUS	Address of this pointer itself if NOREUSE is supplied under CLUSTER.
1088(440)	4	DNRUS	Address of this pointer itself if NOREUSE is supplied under DATA.
1092(444)	4	INRUS	Address of this pointer itself if NOREUSE is supplied under INDEX.
1096(448)	4	CSPND	Address of this pointer itself if SPANNED is supplied under CLUSTER.
1100(44C)	4	DSPND	Address of this pointer itself if SPANNED is supplied under DATA.
1104(450)	4	CNSPD	Address of this pointer itself if NONSPANNED is supplied under CLUSTER.
1108(454)	4	DNSPD	Address of this pointer itself if NONSPANNED is supplied under DATA.
1112(458)	4	MRVBL	Address of this pointer itself if RECOVERABLE is supplied under MASTERCATALOG.
1116(45C)	4	URVBL	Address of this pointer itself if RECOVERABLE is supplied under USERCATALOG.
1120(460)	4	DRVBL	Address of this pointer itself if RECOVERABLE is supplied under DATA.
1124(464)	4	MNRVL	Address of this pointer itself if NOTRECOVERABLE is supplied under MASTERCATALOG.
1128(468)	4	UNRVL	Address of this pointer itself if NOTRECOVERABLE is supplied under USERCATALOG.
1132(46C)	4	DNRVL	Address of this pointer itself if NOTRECOVERABLE is supplied under DATA.
1136(470)	68	*	Reserved—contains zeros.
1204(4B4)	4	MTKPR	Address of information supplied through the *primary* subparameter of TRACKS if TRACKS is supplied under MASTERCATALOG.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
1208(4B8)	4	MTKSC	Address of information supplied through the *secondary* subparameter of TRACKS if TRACKS is supplied under MASTERCATALOG.
1212(4BC)	4	CTKPR	Address of information supplied through the *primary* subparameter of TRACKS if TRACKS is supplied under CLUSTER.
1216(4C0)	4	CTKSC	Address of information supplied through the *secondary* subparameter of TRACKS if TRACKS is supplied under CLUSTER.
1220(4C4)	4	STKPR	Address of information supplied through the *primary* subparameter of TRACKS if TRACKS is supplied under SPACE.
1224(4C8)	4	STKSC	Address of information supplied through the *secondary* subparameter of TRACKS if TRACKS is supplied under SPACE.
1228(4CC)	4	UTKPR	Address of information supplied through the *primary* subparameter of TRACKS if TRACKS is supplied under USERCATALOG.
1232(4D0)	4	UTKSC	Address of information supplied through the *secondary* subparameter of TRACKS if TRACKS is supplied under USERCATALOG.
1236(4D4)	4	PTKPR	Address of information supplied through the *primary* subparameter of TRACKS if TRACKS is supplied under PAGESPACE.
1240(4D8)	4	PTKSC	Address of information supplied through the *secondary* subparameter of TRACKS if TRACKS is supplied under PAGESPACE.
1244(4DC)	4	MCLPR	Address of information supplied through the *primary* subparameter of CYLINDERS if CYLINDERS is supplied under MASTERCATALOG.
1248(4E0)	4	MCLSC	Address of information supplied through the *secondary* subparameter of CYLINDERS if CYLINDERS is supplied under MASTERCATALOG.
1252(4E4)	4	CCLPR	Address of information supplied through the *primary* subparameter of CYLINDERS if CYLINDERS is supplied under CLUSTER.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
1256(4E8)	4	CCLSC	Address of information supplied through the *secondary* subparameter of CYLINDERS if CYLINDERS is supplied under CLUSTER.
1260(4EC)	4	UCLPR	Address of information supplied through the *primary* subparameter of CYLINDERS if CYLINDERS is supplied under USERCATALOG.
1264(4F0)	4	UCLSC	Address of information supplied through the *secondary* subparameter of CYLINDERS if CYLINDERS is supplied under USERCATALOG.
1268(4F4)	4	PCLPR	Address of information supplied through the *primary* subparameter of CYLINDERS if CYLINDERS is supplied under PAGESPACE.
1272(4F8)	4	PCLSC	Address of information supplied through the *secondary* subparameter of CYLINDERS if CYLINDERS is supplied under PAGESPACE.
1276(4FC)	4	SCLPR	Address of information supplied through the *primary* subparameter of CYLINDERS if CYLINDERS is supplied under SPACE.
1280(500)	4	SCLSC	Address of information supplied through the *secondary* subparameter of CYLINDERS if CYLINDERS is supplied under SPACE.
1284(504)	4	MRCPR	Address of information supplied through the *primary* subparameter of RECORDS if RECORDS is supplied under MASTERCATALOG.
1288(508)	4	MRCSC	Address of information supplied through the *secondary* subparameter of RECORDS if RECORDS is supplied under MASTERCATALOG.
1292(50C)	4	CRCPR	Address of information supplied through the *primary* subparameter of RECORDS if RECORDS is supplied under CLUSTER.
1296(510)	4	CRCSC	Address of information supplied through the *secondary* subparameter of RECORDS if RECORDS is supplied under CLUSTER.
1300(514)	4	SRCPR	Address of information supplied through the *primary* subparameter of RECORDS if RECORDS is supplied under SPACE.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
1304(518)	4	SRCSC	Address of information supplied through the *secondary* subparameter of RECORDS if RECORDS is supplied under SPACE.
1308(51C)	4	URCPR	Address of information supplied through the *primary* subparameter of RECORDS if RECORDS is supplied under USERCATALOG.
1312(520)	4	URCSC	Address of information supplied through the *secondary* subparameter of RECORDS if RECORDS is supplied under USERCATALOG.
1316(524)	4	PCRPR	Address of information supplied through the *primary* subparameter of RECORDS if RECORDS is supplied under PAGESPACE.
1320(528)	4	PRCSC	Address of information supplied through the *secondary* subparameter of RECORDS if RECORDS is supplied under PAGESPACE.
1324(52C)	4	DCLPR	Address of information supplied through the *primary* subparameter of CYLINDERS if CYLINDERS is supplied under DATA.
1328(530)	4	DCLSC	Address of information supplied through the *secondary* subparameter of CYLINDERS if CYLINDERS is supplied under DATA.
1332(534)	4	DRCPR	Address of information supplied through the *primary* subparameter of RECORDS if RECORDS is supplied under DATA.
1336(538)	4	DRCSC	Address of information supplied through the *secondary* subparameter of RECORDS if RECORDS is supplied under DATA.
1340(53C)	4	ICLPR	Address of information supplied through the *primary* subparameter of CYLINDERS if CYLINDERS is supplied under INDEX.
1344(540)	4	ICLSC	Address of information supplied through the *secondary* subparameter of CYLINDERS if CYLINDERS is supplied under INDEX.
1348(544)	4	IRCPR	Address of information supplied through the *primary* subparameter of RECORDS if CYLINDERS is supplied under INDEX.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
1352(548)	4	IRCSC	Address of information supplied through the *secondary* subparameter of RECORDS if CYLINDERS is supplied under INDEX.
1356(54C)	4	GETRY	Address of information supplied through the NAME parameter if NAME is supplied under ALTERNATEINDEX.
1360(550)	4	GMODL	Address of this pointer itself if MODEL is supplied under ALTERNATEINDEX.
1364(554)	4	GENAM	Address of information supplied through the *entryname/password* subparameter of MODEL if MODEL is supplied under ALTERNATEINDEX.
1368(558)	4	GMDCT	Address of information supplied through the *CATNAME/password* subparameter of MODEL if MODEL is supplied under ALTERNATEINDEX.
1372(55C)	4	GMDNM	Address of information supplied through the *dname* subparameter of MODEL if MODEL is supplied under ALTERNATEINDEX.
1376(560)	4	GMSTR	Address of information supplied through the MASTERPW parameter if MASTERPW is supplied under ALTERNATEINDEX.
1380(564)	4	GCINT	Address of information supplied through the CONTROLPW parameter if CONTROLPW is supplied under ALTERNATEINDEX.
1384(568)	4	GUPDT	Address of information supplied through the UPDATEPW parameter if UPDATEPW is supplied under ALTERNATEINDEX.
1388(56C)	4	GREAD	Address of information supplied through the READPW parameter if READPW is supplied under ALTERNATEINDEX.
1392(570)	4	GCODE	Address of information supplied through the CODE parameter if CODE is supplied under ALTERNATEINDEX.
1396(574)	4	GATTP	Address of information supplied through the ATTEMPTS parameter if ATTEMPTS is supplied under ALTERNATEINDEX.
1400(578)	4	GAUTH	Address of this pointer itself if the AUTHORIZATION parameter is supplied under ALTERNATEINDEX.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
1404(57C)	4	GEPNM	Address of information supplied through the *entrypoint* subparameter of AUTHORIZATION if the AUTHORIZATION parameter is supplied under ALTERNATEINDEX.
1408(580)	4	GSTRG	Address of information supplied through the *string* subparameter of AUTHORIZATION if AUTHORIZATION is supplied under ALTERNATEINDEX.
1412(584)	4	GTO	Address of information supplied through the TO parameter if TO is supplied under ALTERNATEINDEX.
1416(588)	4	GFOR	Address of information supplied through the FOR parameter if FOR is supplied under ALTERNATEINDEX.
1420(58C)	4	GOWNR	Address of information supplied through the OWNER parameter if OWNER is supplied under ALTERNATEINDEX.
1424(590)	4	GSHAR	Address of this pointer itself if the SHAREOPTIONS parameter is supplied under ALTERNATEINDEX.
1428(594)	4	GSHR1	Address of information supplied through the *crosspartition* subparameter of SHAREOPTIONS if SHAREOPTIONS is supplied under ALTERNATEINDEX.
1432(598)	4	GSHR2	Address of information supplied through the *crosssystem* subparameter of SHAREOPTIONS if SHAREOPTIONS is supplied under ALTERNATEINDEX.
1436(59C)	4	GERAS	Address of this pointer itself if the ERASE parameter is supplied under ALTERNATEINDEX.
1440(5A0)	4	GNERS	Address of this pointer itself if the NOERASE parameter is supplied under ALTERNATEINDEX.
1444(5A4)	4	GKEY	Address of this pointer itself if the KEYS parameter is supplied under ALTERNATEINDEX.
1448(5A8)	4	GKYLN	Address of information supplied through the *length* subparameter of KEYS if KEYS is supplied under ALTERNATEINDEX.
1452(5AC)	4	GKYPS	Address of information supplied through the *offset* subparameter of KEYS if KEYS is supplied under ALTERNATEINDEX.
1456(5B0)	4	GREPL	Address of this pointer itself if the REPLICATE parameter is supplied under ALTERNATEINDEX.

Offset	**Bytes and Bit Pattern**	**Field Name**	**Description: Content, Meaning, Use**
1460(5B4)	4	GNREP	Address of this pointer itself if the NOREPLICATE parameter is supplied under ALTERNATEINDEX.
1464(5B8)	4	GIMBD	Address of this pointer itself if the IMBED parameter is supplied under ALTERNATEINDEX.
1468(5BC)	4	GNIBD	Address of this pointer itself if the NOIMBED parameter is supplied under ALTERNATEINDEX.
1472(5C0)	4	GINDD	Address of information supplied through the FILE parameter if FILE is supplied under ALTERNATEINDEX.
1476(5C4)	4	GVSER	Address of information supplied through the VOLUMES parameter if VOLUMES is supplied under ALTERNATEINDEX.
1480(5C8)	4	GRANG	Address of a count of subparameters supplied through the KEYRANGES parameter if KEYRANGES is supplied under ALTERNATEINDEX.
1484(5CC)	4	GRGLO	Address of information supplied through the *lowkey* subparameter of KEYRANGES if KEYRANGES is supplied under ALTERNATEINDEX.
1488(5D0)	4	GRGHI	Address of information supplied through the *highkey* subparameter of KEYRANGES if KEYRANGES is supplied under ALTERNATEINDEX.
1492(5D4)	4	GORDR	Address of this pointer itself if the ORDERED parameter is supplied under ALTERNATEINDEX.
1496(5D8)	4	GUORD	Address of this pointer itself if the UNORDERED parameter is supplied under ALTERNATEINDEX.
1500(5DC)	4	GSUBA	Address of this pointer itself if the SUBALLOCATION parameter is supplied under ALTERNATEINDEX.
1504(5E0)	4	GUNIQ	Address of this pointer itself if the UNIQUE parameter is supplied under ALTERNATEINDEX.
1508(5E4)	4	GTRKS	Address of this pointer itself if the TRACKS parameter is supplied under ALTERNATEINDEX.
1512(5E8)	4	GTKPR	Address of information supplied through the *primary* subparameter of TRACKS if TRACKS is supplied under ALTERNATEINDEX.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
1516(5EC)	4	GTKSC	Address of information supplied through the *secondary* subparameter of TRACKS if TRACKS is supplied under ALTERNATEINDEX.
1520(5F0)	4	GCYLD	Address of this pointer itself if the CYLINDERS parameter is supplied under ALTERNATEINDEX.
1524(5F4)	4	GCLPR	Address of information supplied through the *primary* subparameter of CYLINDERS if CYLINDERS is supplied under ALTERNATEINDEX.
1528(5F8)	4	GCLSC	Address of information supplied through the *secondary* subparameter of CYLINDERS if CYLINDERS is supplied under ALTERNATEINDEX.
1532(5FC)	4	GRCDS	Address of this pointer itself if the RECORDS parameter is supplied under ALTERNATEINDEX.
1536(600)	4	GRCPR	Address of information supplied through the *primary* subparameter of RECORDS if RECORDS is supplied under ALTERNATEINDEX.
1540(604)	4	GRCSC	Address of information supplied through the *secondary* subparameter of RECORDS if RECORDS is supplied under ALTERNATEINDEX.
1544(608)	4	GRSIZ	Address of this pointer itself if the RECORDSIZE parameter is supplied under ALTERNATEINDEX.
1548(60C)	4	GARSZ	Address of information supplied through the *average* subparameter of RECORDSIZE if RECORDSIZE is supplied under ALTERNATEINDEX.
1552(610)	4	GMRSZ	Address of information supplied through the *maximum* subparameter of RECORDSIZE if RECORDSIZE is supplied under ALTERNATEINDEX.
1556(614)	4	GWCK	Address of this pointer itself if the WRITECHECK parameter is supplied under ALTERNATEINDEX.
1560(618)	4	GNWCK	Address of this pointer itself if the NOWRITECHECK parameter is supplied under ALTERNATEINDEX.
1564(61C)	4	GSPED	Address of this pointer itself if the SPEED parameter is supplied under ALTERNATEINDEX.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
1568(620)	4	GRECV	Address of this pointer itself if the RECOVERY parameter is supplied under ALTERNATEINDEX.
1572(624)	4	GFSPC	Address of this pointer itself if the FREESPACE parameter is supplied under ALTERNATEINDEX.
1576(628)	4	GCIFS	Address of information supplied through the *cipercent* subparameter of FREESPACE if FREESPACE is supplied under ALTERNATEINDEX.
1580(62C)	4	GCAFS	Address of information supplied through the *capercent* subparameter of FREESPACE if FREESPACE is supplied under ALTERNATEINDEX.
1584(630)	4	GBFSZ	Address of information supplied through the BUFFERSPACE parameter if BUFFERSPACE is supplied under ALTERNATEINDEX.
1588(634)	4	GCINV	Address of information supplied through the CONTROLINTERVALSIZE parameter if CONTROLINTERVALSIZE is supplied under ALTERNATEINDEX.
1592(638)	4	GREL	Address of information supplied through the RELATE parameter if RELATE is supplied under ALTERNATEINDEX.
1596(63C)	4	GEEXT	Address of information supplied through the EXCEPTIONEXIT parameter if EXCEPTIONEXIT is supplied under ALTERNATEINDEX.
1600(640)	4	GRUS	Address of information supplied through the REUSE parameter if REUSE is supplied under ALTERNATEINDEX.
1604(644)	4	GNRUS	Address of information supplied through the NOREUSE parameter if NOREUSE is supplied under ALTERNATEINDEX.
1608(648)	4	GUNQK	Address of information supplied through the UNIQUEKEY parameter if UNIQUEKEY is supplied under ALTERNATEINDEX.
1612(64C)	4	GNUQK	Address of information supplied through the NONUNIQUEKEY parameter if NONUNIQUEKEY is supplied under ALTERNATEINDEX.
1616(650)	4	DUNQK	Address of information supplied through the UUNIQUEKEY parameter if UNIQUEKEY is supplied under DATA.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
1620(654)	4	DNUQK	Address of information supplied through the NONUNIQUEKEY parameter if NONUNIQUEKEY is supplied under DATA.
1624(658)	4	GUPG	Address of information supplied through the UPGRADE parameter if UPGRADE is supplied under ALTERNATEINDEX.
1628(65C)	4	GNUPG	Address of information supplied through the NOUPGRADE parameter if NOUPGRADE is supplied under ALTERNATEINDEX.
1632(660)	12	*	Reserved—contains zeros.
1644(66C)	4	RETRY	Address of information supplied through the NAME parameter if NAME is supplied under PATH.
1648(670)	4	RMODL	Address of this pointer itself if the MODEL parameter is supplied under PATH.
1652(674)	4	RENAM	Address of information supplied through the *entryname/password* subparameter of MODEL if MODEL is supplied under PATH.
1656(678)	4	RMDCT	Address of information supplied through the *catname/password* subparameter of MODEL if MODEL is supplied under PATH.
1660(67C)	4	RMDNM	Address of information supplied through the *dname* subparameter of MODEL if MODEL is supplied under PATH.
1664(680)	4	RMSTR	Address of information supplied through the MASTERPW parameter if MASTERPW is supplied under PATH.
1668(684)	4	RCINT	Address of information supplied through the CONTROLPW parameter if CONTROLPW is supplied under PATH.
1672(688)	4	RUPDT	Address of information supplied through the UPDATEPW parameter if UPDATEPW is supplied under PATH.
1676(68C)	4	RREAD	Address of information supplied through the READPW parameter if READPW is supplied under PATH.
1680(690)	4	RCODE	Address of information supplied through the CODE parameter if CODE is supplied under PATH.
1684(694)	4	RATTP	Address of information supplied through the ATTEMPTS parameter if ATTEMPTS is supplied under PATH.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
1688(698)	4	RAUTH	Address of information supplied through the AUTHORIZATION parameter if AUTHORIZATION is supplied under PATH.
1692(69C)	4	REPNM	Address of information supplied through the *entrypoint* subparameter of AUTHORIZATION if AUTHORIZATION is supplied under PATH.
1696(6A0)	4	RSTRG	Address of information supplied through the *string* subparameter of AUTHORIZATION if AUTHORIZATION is supplied under PATH.
1700(6A4)	4	RTO	Address of information supplied through the TO parameter if TO is supplied under PATH.
1704(6A8)	4	RFOR	Address of information supplied through the FOR parameter if FOR is supplied under PATH.
1708(6AC)	4	ROWNR	Address of information supplied through the OWNER parameter if OWNER is supplied under PATH.
1712(6B0)	4	RINDD	Address of information supplied through the FILE parameter if FILE is supplied under PATH.
1716(6B4)	4	RUPD	Address of information supplied through the UPDATE parameter if UPDATE is supplied under PATH.
1720(6B8)	4	RNUPD	Address of information supplied through the NOUPDATE parameter if NOUPDATE is supplied under PATH.
1724(6BC)	4	RPENT	Address of information supplied through the PATHENTRY parameter if PATHENTRY is supplied under PATH.
1728(6C0)	4	CSTAG	Address of information supplied through the STAGE parameter if STAGE is supplied under CLUSTER.
1732(6C8)	4	GSTAG	Address of information supplied through the STAGE parameter if STAGE is supplied under ALTERNATEINDEX.
1736(6C8)	4	DSTAG	Address of information supplied through the STAGE parameter if STAGE is supplied under DATA.
1740(6CC)	4	ISTAG	Address of information supplied through the STAGE parameter if STAGE is supplied under INDEX.
1744(6D0)	4	CBIND	Address of information supplied through the BIND parameter if BIND is supplied under CLUSTER.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
1748(6D4)	4	GBIND	Address of information supplied through the BIND parameter if BIND is supplied under ALTERNATEINDEX.
1752(6D8)	4	DBIND	Address of information supplied through the BIND parameter if BIND is supplied under DATA.
1756(6DC)	4	IBIND	Address of information supplied through the BIND parameter if BIND is supplied under INDEX.
1760(6E0)	4	CCYLF	Address of information supplied through the CYLINDERFAULT parameter if CYLINDERFAULT is supplied under CLUSTER.
1764(6E4)	4	GCYLF	Address of information supplied through the CYLINDERFAULT parameter if CYLINDERFAULT is supplied under ALTERNATEINDEX.
1768(6E8)	4	DCYLF	Address of information supplied through the CYLINDERFAULT parameter if CYLINDERFAULT is supplied under DATA.
1772(6EC)	4	ICYLF	Address of information supplied through the CYLINDERFAULT parameter if CYLINDERFAULT is supplied under INDEX.
1776(6F0)	4	MNSTW	Address of information supplied through the NODESTAGEWAIT parameter if NODESTAGEWAIT is supplied under MASTERCATALOG.
1780(6F4)	4	UNSTW	Address of information supplied through the NODESTAGEWAIT parameter if NODESTAGEWAIT is supplied under USERCATALOG.
1784(6F8)	4	CNSTW	Address of information supplied through the NODESTAGEWAIT if NODESTAGEWAIT is supplied under CLUSTER.
1788(6FC)	4	GNSTW	Address of information supplied through the NODESTAGEWAIT parameter if NODESTAGEWAIT is supplied under ALTERNATEINDEX.
1792(700)	4	DNSTW	Address of information supplied through the NODESTAGEWAIT parameter if NODESTAGEWAIT is supplied under DATA.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
1796(704)	4	INSTW	Address of information supplied through the NODESTAGEWAIT parameter if NODESTAGEWAIT is supplied under INDEX.
1800(708)	4	MSTGW	Address of information supplied through the DESTAGEWAIT parameter if DESTAGEWAIT is supplied under MASTERCATALOG.
1804(70C)	4	USTGW	Address of information supplied through the DESTAGEWAIT parameter if DESTAGEWAIT is supplied under USERCATALOG.
1808(710)	4	CSTGW	Address of information supplied through the DESTAGEWAIT parameter if DESTAGEWAIT is supplied under CLUSTER.
1812(714)	4	GSTGW	Address of information supplied through the DESTAGEWAIT parameter if DESTAGEWAIT is supplied under ALTERNATEINDEX.
1816(718)	4	DSTGW	Address of information supplied through the DESTAGEWAIT parameter if DESTAGEWAIT is supplied under DATA.
1820(71C)	4	ISTGW	Address of information supplied through the DESTAGEWAIT parameter if DESTAGEWAIT is supplied under INDEX.
1824(720)	4	ATO	Address of information supplied through the TO parameter if TO is supplied under NONVSAM.
1828(724)	4	BTO	Address of information supplied through the TO parameter if TO is supplied under GENERATIONDATA GROUP.
1832(728)	4	AFOR	Address of information supplied through the FOR parameter if FOR is supplied under NONVSAM.
1836(72C)	4	BFOR	Address of information supplied through the FOR parameter if FOR is supplied under GENERATION- DATAGROUP.
1840(730)	4	AOWNR	Address of information supplied through the OWNER parameter if OWNER is supplied under NONVSAM.
1844(734)	4	BOWNR	Address of information supplied through the OWNER parameter if OWNER is supplied under GENERATION- DATA GROUP.
1848(738)	4	MBFND	Address of information supplied through the BUFND parameter if BUFND is supplied under MASTERCATALOG.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
1852(73C)	4	UBFND	Address of information supplied through the BUFND parameter if BUFND is supplied under USERCATALOG.
1856(740)	4	DBFND	Address of information supplied through the BUFND parameter if BUFND is supplied under DATA.
1860(744)	4	MBUFNI	Address of information supplied through the BUFNI parameter if BUFNI is supplied under MASTERCATALOG.
1864(748)	4	UBUFNI	Address of information supplied through the BUFNI parameter if BUFNI is supplied under USERCATALOG.
1868(74C)	4	IBUFNI	Address of information supplied through the BUFNI parameter if BUFNI is supplied under INDEX.
1872(750)	4	MCINV	Address of information supplied through the CONTROLINTERVALSIZE parameter if CONTROLINTERVALSIZE is supplied under MASTERCATALOG.
1876(754)	4	UCINV	Address of information supplied through the CONTROLINTERVALSIZE parameter if CONTROLINTERVALSIZE is supplied under USERCATALOG.
1880(758)	4	CRCIL	Address of information supplied through the RECATALOG parameter if RECATALOG is supplied under CLUSTER.
1884(75C)	4	CNRCT	Address of information supplied through the NORECATALOG parameter if NORECATALOG is supplied under CLUSTER.
1888(760)	4	MFSPC	Address of information supplied through the FREESPACE parameter if FREESPACE is supplied under MASTERCATALOG.
1892(764)	4	UFSPC	Address of information supplied through the FREESPACE parameter if FREESPACE is supplied under USERCATALOG.
1896(768)	4	MVCTP	Address of information supplied through the VSAMCATALOG parameter if VSAMCATALOG is supplied under MASTERCATALOG.
1900(76C)	4	UVCTP	Address of information supplied through the VSAMCATALOG parameter if VSAMCATALOG is supplied under USERCATALOG.

	Bytes and		Description:
Offset	Bit Pattern	Field Name	Content, Meaning, Use
1904(770)	4	MIMBD	Address of information supplied through the IMBED parameter if IMBED is supplied under MASTERCATALOG.
1908(774)	4	UIMBD	Address of information supplied through the IMBED parameter if IMBED is supplied under USERCATALOG.
1972(778)	4	GRCTL	Address of information supplied through the RECATALOG parameter if RECATALOG is supplied under ALTERNATEINDEX.
1916(77C)	4	GNRCT	Address of information supplied through the NORECATALOG parameter if NORECATALOG is supplied under ALTERNATEINDEX.
1920(780)	4	RRCTL	Address of information supplied through the RECATALOG parameter if RECATALOG is supplied under PATH.
1924(784)	4	RNRCT	Address of information supplied through the NORECATALOG parameter if NORECATALOG is supplied under PATH.
1928(788)	4	MNIBD	Address of information supplied through the NOIMBED parameter if NOIMBED is supplied under MASTERCATALOG.
1932(78C)	4	UNIBD	Address of information supplied through the NOIMBED parameter if NOIMBED is supplied under USERCATALOG.
1936(790)	4	MNREP	Address of information supplied through the NOREPLICATE parameter if NOREPLICATE is supplied under MASTERCATALOG.
1940(794)	4	UNREP	Address of information supplied through the NOREPLICATE parameter if NOREPLICATE is supplied under USERCATALOG.
1944(798)	4	MRSIZ	Address of information supplied through the RECORDSIZE parameter if RECORDSIZE is supplied under MASTERCATALOG.
1948(79C)	4	URSIZ	Address of information supplied through the RECORDSIZE parameter if RECORDSIZE is supplied under USERCATALOG.
1952(7A0)	4	MREPL	Address of information supplied through the REPLICATE parameter if REPLICATE is supplied under MASTERCATALOG.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
1956(7A4)	4	UREPL	Address of information supplied through the REPLICATE parameter if REPLICATE is supplied under USERCATALOG.
1960(7A8)	4	MSHAR	Address of information supplied through the SHAREOPTIONS parameter if SHAREOPTIONS is supplied under MASTERCATALOG.
1964(7AC)	4	USHAR	Address of information supplied through the SHAREOPTIONS parameter if SHAREOPTIONS is supplied under USERCATALOG.
1968(7B0)	4	MSTRN	Address of information supplied through the STRNO parameter if STRNO is supplied under MASTERCATALOG.
1972(7B4)	4	USTRN	Address of information supplied through the STRNO parameter if STRNO is supplied under USERCATALOG.
1976(7B8)	4	MICTP	Address of information supplied through the ICFCATALOG parameter if ICFCATALOG is supplied under MASTERCATALOG.
1980(7BC)	4	UICTP	Address of information supplied through the ICFCATALOG parameter if ICFCATALOG is supplied under USERCATALOG.
1984(7C0)	4	MCIFS	Address of information supplied through the *cipercent* subparameter of FREESPACE if FREESPACE is supplied under MASTERCATALOG.
1988(7C4)	4	UCIFS	Address of information supplied through the *cipercent* subparameter of FREESPACE if FREESPACE is supplied under USERCATALOG.
1992(7C8)	4	MCAFS	Address of information supplied through the *capercent* subparameter of FREESPACE if FREESPACE is supplied under MASTERCATALOG.
1996(7CC)	4	UCAFS	Address of information supplied through the *capercent* subparameter of FREESPACE if FREESPACE is supplied under USERCATALOG.
2000(7D0)	4	MSHR1	Address of information supplied through the *crossregion* subparameter of SHAREOPTIONS if SHAREOPTIONS is supplied under MASTERCATALOG.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
2004(7D4)	4	USHR1	Address of information supplied through the *crossregion* subparameter of SHAREOPTIONS if SHAREOPTIONS is supplied under USERCATALOG.
2008(7D4)	4	MSHR2	Address of information supplied through the *crosssystem* subparameter of SHAREOPTIONS if SHAREOPTIONS is supplied under MASTERCATALOG.
2012(7DC)	4	USHR2	Address of information supplied through the *crosssystem* subparameter of SHAREOPTIONS if SHAREOPTIONS is supplied under USERCATALOG.
2016(7E0)	4	MARSZ	Address of information supplied through the *average* subparameter of RECORDSIZE if RECORDSIZE is supplied under MASTERCATALOG.
2020(7E4)	4	UARSZ	Address of information supplied through the *average* subparameter of RECORDSIZE if RECORDSIZE is supplied under USERCATALOG.
2024(7E8)	4	MMRSZ	Address of information supplied through the *maximum* subparameter of RECORDSIZE if RECORDSIZE is supplied under MASTERCATALOG.
2028(7EC)	4	UMRSZ	Address of information supplied through the *maximum* subparameter of RECORDSIZE if RECORDSIZE is supplied under USERCATALOG.

DELETE FDT

Offset	Content	
0 (0)	D E L E	T E b b
8 (8)	↑ entryname/password	↑ CATALOG
16 (10)	↑ catname/password	↑ dname
24 (18)	↑ FILE	↑ PURGE
32 (20)	↑ NOPURGE	↑ ERASE
40 (28)	↑ NOERASE	0
48 (30)	↑ CLUSTER	↑ SPACE
56 (38)	↑ USERCATALOG	↑ MASTERCATALOG
64 (40)	↑ NONVSAM	↑ SCRATCH
72 (48)	↑ NOSCRATCH	↑ PAGESPACE
80 (50)	↑ GENERATIONDATAGROUP	↑ ALIAS
88 (58)	↑ AIX	↑ PATH
96 (60)	↑ FRC	↑ NFRC
104(68)	↑ RCVY	↑ NRCVY
112(70)	↑ TNAME	↑ VVR

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb aligned left and padded with blanks—DELETEƀ ƀ.
8(8)	4	NTRY	Address of information supplied through the *entryname/password* parameter.
12(C)	4	CATLG	Address of this pointer itself if the CATALOG parameter is supplied.
16(10)	4	CAT	Address of information supplied through the *catname/password* subparameter of the CATALOG parameter.
20(14)	4	CATDD	Address of information supplied through the *dname* subparameter of the CATALOG parameter.
24(18)	4	INDD	Address of information supplied through the FILE parameter.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
28(1C)	4	PURGE	Address of this pointer itself if the PURGE parameter is supplied or defaulted.
32(20)	4	NOPUR	Address of this pointer itself if the NOPURGE parameter is supplied.
36(24)	4	ERASE	Address of this pointer itself if the ERASE parameter is supplied.
40(28)	4	NOERA	Address of this pointer itself if the NOERASE parameter is supplied.
44(2C)	4	*	Reserved—contains zero.
48(30)	4	CLUST	Address of this pointer itself if the CLUSTER parameter is supplied.
52(34)	4	SPACE	Address of this pointer itself if the SPACE parameter is supplied.
56(38)	4	UCAT	Address of this pointer itself if the USERCATALOG parameter is supplied.
60(3C)	4	MCAT	Address of this pointer itself if the MASTERCATALOG parameter is supplied.
64(40)	4	ALIEN	Address of this pointer itself if the NONVSAM parameter is supplied.
68(44)	4	SCR	Address of this pointer itself if the SCRATCH parameter is supplied.
72(48)	4	NSCR	Address of this pointer itself if the NOSCRATCH parameter is supplied.
76(4C)	4	PGSPC	Address of this pointer itself if the PAGESPACE parameter is supplied.
80(50)	4	GDG	Address of this pointer itself if the GENERATIONDATA GROUP parameter is supplied.
84(54)	4	ALIAS	Address of this pointer itself if the ALIAS parameter is supplied.
88(58)	4	AIX	Address of this pointer itself if the ALTERNATE INDEX parameter has been supplied.
92(5C)	4	PATH	Address of this pointer itself if the PATH parameter has been supplied.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
96(60)	4	FRC	Address of this pointer itself if the FORCE parameter has been supplied.
100(64)	4	NFRC	Address of this pointer itself if the NOFORCE parameter has been supplied.
104(68)	4	RCVY	Address of this pointer itself if the RECOVERY parameter has been supplied.
108(6C)	4	NRCVY	Address of this pointer itself if the NORECOVERY parameter has been supplied.
112(70)	4	TNAME	Address of this pointer itself if the TRUENAME parameter has been supplied.
116(74)	4	VVR	Address of this pointer itself if the VVR parameter has been supplied.

DIAGNOSE FDT

Offset	Content	
0 (0)	DIAG	NOSE
8 (8)	↑ ICFCATALOG	↑ VVDS
16 (10)	↑ NFILE	↑ INDATASET
24 (18)	↑ OUTFILE	↑ COMPAREDD
32 (20)	↑ COMPAREDS	↑ INCLUDE
40 (28)	↑ EXCLUDE	↑ ENTRIES
48 (30)	↑ LEVEL	↑ CATALOG
56 (38)	↑ ENTRIES	↑ LEVEL
64 (40)	↑ CATALOG	↑ LIST
72 (48)	↑ NOLIST	↑ DUMP
80 (50)	↑ NODUMP	↑ ERRORLIMIT

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb name—'DIAGNOSE'.
8(8)	4	ICFC	Address of this pointer if ICFCATALOG parameter is supplied.
12(C)	4	VVDS	Address of this pointer if VVDS parameter is supplied.
16(10)	4	INFL	Address of information supplied through INFILE parameter.
20(14)	4	INDS	Address of information supplied through INDATASET parameter.
24(18)	4	OUTDD	Address of information supplied through OUTFILE parameter.
28(1C)	4	CMDD	Address of information supplied through COMPAREDD parameter.
32(20)	4	CMDS	Address of information supplied through COMPAREDS parameter.
36(24)	4	NCLD	Address of information supplied through INCLUDE parameter.
40(28)	4	XCLD	Address of information supplied through EXCLUDE parameter.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
44(2C)	4	IENT	Address of information supplied through ENTRIES if INCLUDE is specified.
48(30)	4	ILVL	Address of information supplied through LEVEL if INCLUDE is specified.
52(34)	4	ICAT	Address of information supplied through CATALOG if INCLUDE is specified.
56(38)	4	EENT	Address of information supplied through ENTRIES if EXCLUDE is specified.
60(3C)	4	ELVL	Address of information supplied through LEVEL if EXCLUDE is specified.
64(40)	4	ECAT	Address of information supplied through CATALOG if EXCLUDE is specified.
68(44)	4	LIST	Address of this pointer if LIST is specified.
72(48)	4	NLST	Address of this pointer if NOLIST is specified.
76(4C)	4	DUMP	Address of this pointer if DUMP is specified.
80(50)	4	NDMP	Address of this pointer if NODUMP is specified.
84(54)	4	ELMT	Address of information supplied through ERRORLIMIT parameter.

EXAMINE FDT

Offset	Content	
0 (0)	EXAM	INEb
8 (8)	↑NAME	↑INDEXTEST
16 (10)	↑NOINDEXTEST	↑DATATEST
24 (18)	↑NODATATEST	↑ERR(.IMIT

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb name—EXAMINE.
8(8)	4	NAME	Address of information supplied through NAME parameter.
12(C)	4	IXT	Address of this field if INDEXTEST parameter is supplied.
16(10)	4	NIXT	Address of this field if NOINDEXTEST parameter is supplied.
20(14)	4	DAT	Address of this field if DATATEST parameter is supplied.
24(18)	4	NDAT	Address of this field i:' NODATATEST parameter is supplied.
28(1C)	4	ELMT	Address of informations supplied through ERRORLIMIT parameter.

EXPORT FDT

Offset	Content	
0 (0)	E X P O	R T b b
8 (8)	↑entryname/password	↑INFILE
16 (10)	↑OUTFILE	↑dname
24 (18)	0	↑TEMPORARY
32 (20)	↑PERMANENT	↑INHIBITSOURCE
40 (28)	↑INHIBITTARGET	↑ERASE
48 (30)	↑NOERASE	↑PURGE
56 (38)	↑NOPURGE	↑DISCONNECT
64 (40)	↑NOINHIBITSOURCE	↑NOINHIBITTARGET
72 (48)	0	↑OUTDATASET
80 (50)	0	0
88 (58)	0	

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb aligned left and padded with blanks—EXPORTƀ ƀ.
8(8)	4	ENT	Address of information supplied through the *entryname/password* parameter.
12(C)	4	INDD	Address of information supplied through the INFILE parameter.
16(10)	4	OUT	Address of this pointer itself if the OUTFILE parameter is supplied.
20(14)	4	OUTDD	Address of information supplied through the *dname* subparameter of the OUTFILE parameter.
24(18)	4	ENVIR	Reserved—contains zeros.
28(1C)	4	TEMP	Address of this pointer itself if the TEMPORARY parameter is supplied.
32(20)	4	PERM	Address of this pointer itself if the PERMANENT parameter is supplied.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
36(24)	4	INHBS	Address of this pointer itself if the INHIBITSOURCE parameter is supplied.
40(28)	4	INHBT	Address of this pointer itself if the INHIBITTARGET parameter is supplied.
44(2C)	4	ERASE	Address of this pointer itself if the ERASE parameter is supplied.
48(30)	4	NOERS	Address of this pointer itself if the NOERASE parameter is supplied.
52(34)	4	PURGE	Address of this pointer itself if the PURGE parameter is supplied.
56(38)	4	NPRG	Address of this pointer itself if the NOPURGE parameter is supplied.
60(3C)	4	DISCT	Address of this pointer itself if the DISCONNECT parameter is supplied.
64(40)	4	NINHS	Address of this pointer itself if the NOINHIBITSOURCE parameter is supplied.
68(44)	4	NINHT	Address of this pointer itself if the NOINHIBITTARGET parameter is supplied.
72(48)	4	*	Reserved—contains zeros.
76(4C)	4	OUTDS	Address of information supplied through the OUTDATASET parameter.
80(50)	4	*	Reserved—contains zeros.
84(54)	4	PDEV	Reserved—contains zeros.
88(58)	4	BLKSZ	Reserved.

EXPORTRA FDT

Offset	Content	
0 (0)	EXPO	RTRA
8 (8)	↑ FORCE	↑ NOFORCE
16 (10)	↑ OUTFILE	↑ CRA count
24 (18)	↑ dname	↑ ALL
32 (20)	↑ NONE	↑ ENTRIES
40 (28)	↑ INFILE	↑ MASTERPW
48 (30)	0	0
56 (38)	↑ dname	↑ entryname
64 (40)	↑ dname	0

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb—EXPORTRA.
8(8)	4	FRC	Address of this pointer itself if the FORCE parameter has been supplied.
12(C)	4	NFRC	Address of this pointer itself if the NOFORCE parameter has been supplied.
16(10)	4	OUT	Address of this pointer itself if the OUTFILE parameter has been supplied.
20(14)	4	CRACNT	Count of the number of catalog recovery areas (CRAs) that were provided in the EXPORTRA command.
24(18)	4	CRADDPTR	Address of an array of pointers. Each pointer points at the *dname* for the CRA it relates to in the order that they appear in the EXPORTRA command.
28(1C)	4	ALLNTPTR	Address of an array of pointers. Each pointer points to itself if ALL was specified for the related CRA.
32(20)	4	NONEPTR	Address of an array of pointers. Each pointer points to itself if NONE was specified for the related CRA or was defaulted for the CRA.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
36(24)	4	ENTREPTR	Address of an array of counts. Each count indicates the number of entries if ENTRIES was specified for the related CRA. Zero indicates that ENTRIES was not specified.
40(28)	4	IFILEPTR	Address of an array of pointers. Each pointer points to the *dname* subparameter of the INFILE subparameter, i.e., the *dname* to be used for a CRA if ALL was specified for that CRA.
44(2C)	4	MRPW	Address of information supplied through the *password* subparameter of the MASTERPW parameter.
48(30)	4	ENVIR	Reserved—contains zeros.
52(34)	4	PDEV	Reserved—contains zeros.
56(38)	4	OUTDD	Address of information supplied through the *dname* subparameter of the OUTFILE parameter.
60(3C)	4	ENTNMPTR	Address of an array of pointers. Each pointer points to the *entryname* subparameter of the ENTRIES subparameter, i.e., all of the entry names in each CRA specified.
64(40)	4	ENTDNPTR	Address of an array of pointers. Each pointer points to the *dname* subparameter of the ENTRIES subparameter to be used to export the associated entry name in ENTNMPTR.
68(44)	4	BLKSZ	Reserved—contains zeros.

IMPORT FDT

Offset	Content	
0 (0)	I M P O	R T b b
8 (8)	↑INFILE	↑OUTFILE
16 (10)	↑OBJECTS	↑objectname
24 (18)	↑NEWNAME	↑FILE
32 (20)	↑VOLUMES	↑KEYRANGES
40 (28)	↑DEVICETYPES	↑ORDERED
48 (30)	↑UNORDERED	↑lowkey
56 (38)	↑highkey	↑CONNECT
64 (40)	↑dname	0
72 (48)	↑PURGE	↑NOPURGE
80 (50)	↑ERASE	↑NOERASE
96 (60)	0	↑INDATASET
104(68)	↑OUTDATASET	0
136(88)	↑CATALOG	↑IMPTY
144(9C)	↑SRAC	↑NSRAC

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb aligned left and padded with blanks—IMPORTb̸ b̸.
8(8)	4	IN	Address of this pointer itself if the INFILE parameter is supplied.
12(C)	4	OUTDD	Address of information supplied through the OUTFILE parameter.
16(10)	4	OBJTS	Address of a count of subparameters supplied through the OBJECTS parameter.
20(14)	4	OBJNMPTR	Address of information supplied through the *objectname* subparameter of the OBJECTS parameter.
24(18)	4	NEWNMPTR	Address of information supplied through the NEWNAME subparameter of the OBJECTS parameter.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
28(1C)	4	OBJFLPTR	Address of information supplied through the FILE subparameter of the OBJECTS parameter.
32(20)	4	LISTVPTR	Address of information supplied through the VOLUMES subparameter of the OBJECTS parameter.
36(24)	4	RANGEPTR	Address of a count of *lowkey highkey* pairs supplied through the KEYRANGES subparameter of the OBJECTS parameter.
40(28)	4	DEVTPTR	Address of information supplied through the DEVICETYPES subparameter of the OBJECTS parameter.
44(2C)	4	ORDPTR	Address of information supplied through the ORDERED subparameter of the OBJECTS parameter.
48(30)	4	UNORDPTR	Address of information supplied through the UNORDERED subparameter of the OBJECTS parameter.
52(34)	4	LOWKYPTR	Address of information supplied through the *lowkey* subparameter of the KEYRANGES parameter.
56(38)	4	HIKEYPTR	Address of information supplied through the *highkey* subparameter of the KEYRANGES parameter.
60(3C)	4	CON	Address of this pointer itself if the CONNECT parameter is supplied.
64(40)	4	INDD	Address of information supplied through the *dname* subparameter of the INFILE parameter.
68(44)	4	ENV	Reserved—contains zeros.
72(48)	4	PRG	Address of this pointer itself if the PURGE parameter is supplied.
76(4C)	4	NPRG	Address of this pointer itself if the NOPURGE parameter is supplied.
80(50)	4	ERAS	Address of this pointer itself if the ERASE parameter is supplied.
84(54)	4	NERAS	Address of this pointer itself if the NOERASE parameter is supplied.
88(58)	4	BLKSZ	Reserved—contains zeros.
92(5C)	4	PDEV	Reserved—contains zeros.
96(60)	4	RCSZE	Reserved—contains zeros.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
100(64)	4	INDS	Address of information supplied through the INDATASET parameter.
104(68)	4	OUTDS	Address of information supplied through the OUTDATASET parameter.
108(6C)	28	*	Reserved—contains zeros.
136(88)	4	CAT	Address of information supplied through the CATALOG parameter.
140(8C)	4	IMPTY	Address of this pointer itself if the INTOEMPTY parameter is supplied.
144(90)	4	SRAC	Address of this pointer itself if the SAVRAC parameter is specified.
148(94)	4	NSRAC	Address of this pointer itself if the NOSAVRAC parameter is specified.

IMPORTRA FDT

Offset	Content	
0 (0)	IMPO	RTRA
8 (8)	↑ INFILE	↑ OUTFILE
16 (10)	↑ OBJECTS	↑ objectname
24 (18)	0	0
32 (20)	↑ VOLUMES	0
40 (28)	↑ DEVICETYPE	0
48 (30)	0	0
56 (38)	0	0
64 (40)	↑ dname	0
72 (48)	0	0
80 (50)	0	0
88 (58)	0	0
96 (60)	0	↑ INDATASET
104(68)	0	0
112(70)	0	0
120(78)	0	0
128(80)	0	0
136(88)	↑ CATALOG	↑ SRAC
144(9C)	↑ NSRAC	

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb—IMPORTRA.
8(8)	4	IN	Address of this pointer itself if the INFILE parameter has been supplied.
12(C)	4	OUTDD	Address of information supplied through the OUTFILE parameter.
16(10)	4	OBJTS	Address of the count of objects supplied through the OBJECTS parameter.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
20(14)	4	OBJNMPTR	Address of information supplied through the *name* subparameter of the OBJECTS parameter.
24(18)	8	*	Reserved—contains zeros.
32(20)	4	LISTVPTR	Address of information supplied through the VOLUMES subparameter of the OBJECTS parameter.
36(24)	4	*	Reserved—contains zeros.
40(28)	4	DEVTPTR	Address of information supplied through the DEVICETYPE subparameter of the OBJECTS parameter.
44(2C)	20	*	Reserved—contains zeros.
64(40)	4	INDD	Address of information supplied through the *dname* subparameter of the INFILE parameter.
68(44)	4	ENV	Reserved—contains zeros.
72(48)	16	*	Reserved—contains zeros.
88(58)	4	BLKSZ	Reserved—contains zeros.
92(5C)	4	PDEV	Reserved—contains zeros.
96(60)	4	*	Reserved—contains zeros.
100(64)	4	INDS	Address of the input data set name.
104(68)	32	*	Reserved—contains zeros.
136(88)	4	CAT	Address of information supplied through the CATALOG parameter.
140(8C)	4	SRAC	Address of this location itself if the SAVRAC parameter is specified.
144(90)	4	NSRAC	Address of this location itself if the NOSAVRAC parameter is specified.

LISTCAT FDT

Offset	Content	
0 (0)	L I S T	C A T b
8 (8)	↑ CATALOG	↑ OUTFILE
16 (10)	↑ ENTRIES	0
24 (18)	↑ CLUSTER	↑ DATA
32 (20)	↑ INDEX	↑ SPACE
40 (28)	↑ NONVSAM	↑ USERCATALOG
48 (30)	↑ catname/password	↑ dname
56 (38)	0	↑ NAME
64 (40)	↑ ALL	↑ VOLUME
72 (48)	↑ ALLOCATION	↑ ALIAS
80 (50)	↑ GENERATIONDATAGROUP	↑ PAGESPACE
88 (58)	↑ LEVEL	↑ ALTERNATEINDEX
96 (60)	↑ PATH	↑ NOTUSABLE
104(68)	↑ CREATION	↑ EXPIRATION
112(70)	↑ HISTORY	↑ FILE

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb aligned left and padded with blanks—LISTCATƀ.
8(8)	4	CAT	Address of this pointer itself if the CATALOG parameter is supplied.
12(C)	4	OUTDD	Address of information supplied through the OUTFILE parameter.
16(10)	4	ENT	Address of information supplied through the ENTRIES parameter.
20(14)	4	*	Reserved—contains zeros.
24(18)	4	CLUST	Address of this pointer itself if the CLUSTER parameter is supplied.
28(1C)	4	DATUM	Address of this pointer itself if the DATA parameter is supplied.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
32(20)	4	INDEX	Address of this pointer itself if the INDEX parameter is supplied.
36(24)	4	SPACE	Address of this pointer itself if the SPACE parameter is supplied.
40(28)	4	ALIEN	Address of this pointer itself if the NONVSAM parameter is supplied.
44(2C)	4	UCAT	Address of this pointer itself if the USERCATALOG parameter is supplied.
48(30)	4	CATNM	Address of information supplied through the *catname/password* subparameter of the CATALOG parameter.
52(34)	4	CATDD	Address of information supplied through the *dname* subparameter of the CATALOG parameter.
56(38)	4	*	Reserved—contains zeros.
60(3C)	4	NAME	Address of this pointer itself if the NAME parameter is supplied.
64(40)	4	FALL	Address of this pointer itself if the ALL parameter is supplied.
68(44)	4	VOL	Address of this pointer itself if the VOLUME parameter is supplied.
72(48)	4	ALLOC	Address of this pointer itself if the ALLOCATION parameter is supplied.
76(4C)	4	ALIAS	Address of this pointer itself if the ALIAS parameter is supplied.
80(50)	4	GDG	Address of this pointer itself if the GENERATIONDATA GROUP parameter is supplied.
84(54)	4	PGSPC	Address of this pointer itself if the PAGESPACE parameter is supplied.
88(58)	4	LVL	Address of information supplied through the LEVEL parameter.
92(5C)	4	AIX	Address of this pointer itself if the ALTERNATEINDEX parameter has been supplied.
96(60)	4	PATH	Address of this pointer itself if the PATH parameter has been supplied.
100(64)	4	NUSE	Address of this pointer itself if the NOTUSABLE parameter has been supplied.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
104(68)	4	CREAT	Address of information supplied through the CREATION parameter.
108(6C)	4	EXPIR	Address of information supplied through the EXPIRATION parameter.
112(70)	4	HIST	Address of this pointer itself if the HISTORY parameter is supplied.
116(74)	4	INDD	Address of information supplied through the *dname* subparameter of the FILE parameter.

LISTCRA FDT

Offset	Content	
0 (0)	LIST	CRAb
8 (8)	↑ INFILE	↑ COMPARE
16 (10)	↑ NOCOMPARE	↑ DUMP
24 (18)	↑ NAME	↑ CATALOG
32 (20)	↑ catname/password	↑ dname
40 (28)	↑ MASTERPW	↑ SEQUENTIALDUMP

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb aligned left and padded with blanks—LISTCRAb̸.
8(8)	4	IFILE	Address of information supplied through the *dname* subparameter of the INFILE parameter.
12(C)	4	CMPR	Address of this pointer itself if the COMPARE parameter has been supplied.
16(10)	4	NCMPR	Address of this pointer itself if the NOCOMPARE parameter has been supplied.
20(14)	4	DUMP	Address of this pointer itself if the DUMP parameter has been supplied.
24(18)	4	NAME	Address of this pointer itself if the NAME parameter has been supplied.
28(1C)	4	CAT	Address of this pointer itself if the CATALOG parameter has been supplied.
32(20)	4	CATNM	Address of information supplied through the *catname/password* subparameter of the CATALOG parameter.
36(24)	4	CATDN	Address of information supplied through the *dname* subparameter of the CATALOG parameter.
40(28)	4	MRPW	Address of information supplied through the *password* subparameter of the MASTERPW parameter.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
44(32)	4	SDUMP	Address of this pointer itself if the SEQUENTIALDUMP parameter has been supplied.

LISTDATA FDT

Offset	Content	
0 (0)	LIST	DATA
8 (8)	↑ COUNTS	↑ STATUS
16 (10)	↑ FILE	↑ VOLUME
24 (18)	↑ UNIT	↑ DEVICE Q
32 (20)	↑ SUBSYSTEM	↑ ALL
40 (28)	↑ OUTFILE	↑ OUTDATASET
40 (28)	↑ LEGEND	↑ NOLOGEND

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb—LISTDATA.
8(8)	4	CNTS	Address of this pointer itself if the COUNTS parameter has been supplied.
12(C)	4	STAT	Address of this pointer itself if the STATUS parameter has been supplied.
16(10)	4	FILE	Address of information supplied through the FILE parameter.
20(14)	4	VOL	Address of information supplied through the VOLUME parameter.
24(18)	4	UNT	Address of information supplied through the UNIT parameter.
28(1C)	4	DEV	Address of this pointer itself if the DEVICE parameter has been supplied.
32(20)	4	SSYS	Address of this pointer itself if the SUBSYSTEM parameter has been supplied.
36(24)	4	LALL	Address of this pointer itself if the ALL parameter has been supplied.
40(28)	4	OFILE	Address of information supplied through the OUTFILE parameter.
44(2C)	4	ODS	Address of information supplied through the OUTDATASET parameter.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
48(30)	4	LGND	Address of this pointer itself if the LEGEND parameter has been supplied.
52(34)	4	NLGND	Address of this pointer itself if the NOLEGEND parameter has been supplied.

PARM FDT

Offset	Content	
0 (0)	P A R M	b b b b
8 (8)	↑ TEST	↑ OFF
16 (10)	↑ TRACE	↑ AREAS
24 (18)	↑ FULL	↑ dumpid
32 (20)	↑ count1	↑ count2
40 (28)	↑ GRAPHICS	↑ CHAIN
48 (30)	↑ TABLE	↑ MARGINS
56 (38)	↑ leftmargin	↑ rightmargin
64 (40)	↑ AN	↑ HN
72 (48)	↑ PN	↑ QN
80 (50)	↑ RN	↑ SN
88 (58)	↑ TN	

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb aligned left and padded with blanks—PARMb̸ b̸ b̸ b̸.
8(8)	4	TEST	Address of this pointer itself if the TEST parameter is supplied.
12(C)	4	TOFF	Address of this pointer itself if the OFF parameter is supplied.
16(10)	4	TRACE	Address of this pointer itself if the TRACE parameter is supplied.
20(14)	4	AREA	Address of information supplied through the AREAS parameter.
24(18)	4	FULL	Address of a count of subparameters supplied through the FULL parameter.
28(1C)	4	FIDPTR	Address of information supplied through the *dumpid* subparameter of the FULL parameter.
32(20)	4	BEGINPTR	Address of information supplied through the *count1* subparameter of the FULL parameter.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
36(24)	4	COUNTPTR	Address of information supplied through the *count2* subparameter of the FULL parameter.
40(28)	4	GRAPH	Address of this pointer itself if the GRAPHICS parameter is supplied.
44(2C)	4	CHAIN	Address of information supplied through the CHAIN parameter.
48(30)	4	TABLE	Address of information supplied through the TABLE parameter.
52(34)	4	MARG	Address of this pointer itself if the MARGINS parameter is supplied.
56(38)	4	LMARG	Address of information supplied through the *leftmargin* subparameter of the MARGINS parameter.
60(3C)	4	RMARG	Address of information supplied through the *rightmargin* subparameter of the MARGINS parameter.
64(40)	4	CHNAN	Address of this pointer itself if the AN subparameter of the CHAIN parameter is supplied.
68(44)	4	CHNHN	Address of this pointer itself if the HN subparameter of the CHAIN parameter is supplied.
72(48)	4	CHNPN	Address of this pointer itself if the PN subparameter of the CHAIN parameter is supplied.
76(4C)	4	CHNQN	Address of this pointer itself if the QN subparameter of the CHAIN parameter is supplied.
80(50)	4	CHNRN	Address of this pointer itself if the RN subparameter of the CHAIN parameter is supplied.
84(54)	4	CHNSN	Address of this pointer itself if the SN subparameter of the CHAIN parameter is supplied.
88(58)	4	CHNTN	Address of this pointer itself if the TN subparameter of the CHAIN parameter is supplied.

PRINT FDT

Offset	Content	
0 (0)	P R I N	T b b b
8 (8)	↑ INFILE	0
16 (10)	↑ FROMKEY	↑ FROMADDRESS
24 (18)	↑ SKIP	↑ TOKEY
32 (20)	↑ TOADDRESS	↑ COUNT
40 (28)	↑ dname/password	↑ INDATASET
48 (30)	0	↑ HEX
56 (38)	↑ CHARACTER	↑ DUMP
64 (40)	0	↑ ENVIRONMENT
72 (48)	↑ RECORDFORMAT	↑ BLOCKSIZE
80 (50)	↑ RECORDSIZE	0
88 (58)	↑ HINDEXDEVICE	↑ PRIMEDATADEVICE
96 (60)	↑ FIXUNB	↑ FIXBLK
104(68)	↑ VARUNB	↑ VARBLK
112(70)	↑ SPNUNB	↑ SPNBLK
120(78)	↑ UNDEF	↑ FROMNUMBER
128(80)	↑ TONUMBER	

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb aligned left and padded with blanks—PRINTb̸ b̸ b̸.
8(8)	4	INDN	Address of this pointer itself if the INFILE parameter is supplied.
12(C)	4	OUTDD	Address of information supplied through the OUTFILE parameter.
16(10)	4	FMKYC	Address of information supplied through the FROMKEY parameter.
20(14)	4	FMRBA	Address of information supplied through the FROMADDRESS parameter.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
24(18)	4	SKIP	Address of information supplied through the SKIP parameter.
28(1C)	4	TOKYC	Address of information supplied through the TOKEY parameter.
32(20)	4	TORBA	Address of information supplied through the TOADDRESS parameter.
36(24)	4	COUNT	Address of information supplied through the COUNT parameter.
40(28)	4	INPDD	Address of information supplied through the *dname/password* subparameter of the INFILE parameter.
44(2C)	4	INDS	Address of information supplied through the INDATASET parameter.
48(30)	4	*	Reserved—contains zeros.
52(34)	4	FHEX	Address of this pointer itself if the HEX parameter is supplied.
56(38)	4	FCHAR	Address of this pointer itself if the CHARACTER parameter is supplied.
60(3C)	4	FDUMP	Address of this pointer itself if the DUMP parameter is supplied.
64(40)	4	*	Reserved—contains zeros.
68(44)	4	IENV	Reserved—contains zeros.
72(48)	4	IRFMT	Reserved—contains zeros.
76(4C)	4	IBKSZ	Reserved—contains zeros.
80(50)	4	IRCSZ	Reserved—contains zeros.
84(54)	4	*	Reserved—contains zeros.
88(58)	4	IHDEV	Reserved—contains zeros.
92(5C)	4	IPDEV	Reserved—contains zeros.
96(60)	4	IFUNB	Reserved—contains zeros.
100(64)	4	IFBLK	Reserved—contains zeros.
104(68)	4	IVUNB	Reserved—contains zeros.
108(6C)	4	IVBLK	Reserved—contains zeros.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
112(70)	4	ISUNB	Reserved—contains zeros.
116(74)	4	ISBLK	Reserved—contains zeros.
120(78)	4	IUNDF	Reserved—contains zeros.
124(7C)	4	FMNUM	Address of information supplied through the FROMNUMBER parameter.
128(80)	4	TONUM	Address of information supplied through the TONUMBER parameter.

REPRO FDT

Offset	Content	
0 (0)	R E P R	0 b b b
8 (8)	↑ INFILE	↑ OUTFILE
16 (10)	↑ FROMKEY	↑ FROMADDRESS
24 (18)	↑ SKIP	↑ TOKEY
32 (20)	↑ TOADDRESS	↑ COUNT
40 (28)	↑ dname/password	↑ dname/password
48 (30)	↑ INDATASET	↑ OUTDATASET
56 (38)	↑ FROMNUMBER	↑ TONUMBER
64 (40)	0	↑ ENVIRONMENT
72 (48)	↑ RECORDFORMAT	↑ BLOCKSIZE
80 (50)	↑ RECORDSIZE	0
88 (58)	↑ HINDEXDEVICE	↑ PRIMEDATADEVICE
96 (60)	↑ FIXUNB	↑ FIXBLK
104(68)	↑ VARUNB	↑ VARBLK
112(70)	↑ SPNUNB	↑ SPNBLK
120(78)	↑ UNDEF	0
128(80)	0	↑ ENVIRONMENT
136(88)	↑ RECORDFORMAT	↑ BLOCKSIZE
144(90)	↑ RECORDSIZE	0
152(98)	↑ HINDEXDEVICE	↑ PRIMEDATADEVICE
160(A0)	↑ FIXUNB	↑ FIXBLK
168(A8)	↑ VARUNB	↑ VARBLK
176(B0)	↑ SPNUNB	↑ SPNBLK
184(B8)	↑ UNDEF	0
192(C0)	0	↑ DUMMY

Offset	Content	
200(C8)	↑REPLACE	↑NOREPLACE
208(D0)	↑REUSE	↑NOREUSE
216(D8)	↑ENCIPHER	↑DECIPHER
224(E0)	↑INTERNALKEYNAME	↑EXTERNALKEYNAME
232(E8)	↑PRIVATEKEY	↑SHIPKEYNAMES
240(F0)	↑DATAKEYVALUE	↑DATAKEYFILE
248(F8)	↑STOREKEYNAME	↑STOREDATAKEY
256(100)	↑NOSTOREDATAKEY	↑CIPHERUNIT
264(108)	↑USERDATA	↑SYSTEMKEY
272(110)	↑SYSTEMKEYNAME	↑SYSTEMDATAKEY
280(118)	↑DATAKEYVALUE	↑DATAKEYFILE
288(120)	↑dsname/password	↑ENVIRONMENT
296(128)	↑dummy	0
304(130)	0	0
312(138)	0	0
320(140)	0	↑FILE
328(148)	↑MERGECAT	0
336(150)	↑ENTRIES data	↑LEVEL data

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb aligned left and padded with blanks—REPROb̸ b̸ b̸.
8(8)	4	INDN	Address of this pointer itself if the INFILE parameter is supplied.
12(C)	4	OUTDN	Address of this pointer itself if the OUTFILE parameter is supplied.
16(10)	4	FMKYC	Address of information supplied through the FROMKEY parameter.
20(14)	4	FMRBA	Address of information supplied through the FROMADDRESS parameter.
24(18)	4	SKIP	Address of information supplied through the SKIP parameter.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
28(1C)	4	TOKYC	Address of information supplied through the TOKEY parameter.
32(20)	4	TORBA	Address of information supplied through the TOADDRESS parameter.
36(24)	4	COUNT	Address of information supplied through the COUNT parameter.
40(28)	4	INPDD	Address of information supplied through the *dname/password* subparameter of the INFILE parameter.
44(2C)	4	OUTDD	Address of information supplied through the *dname/password* subparameter of the OUTFILE parameter.
48(30)	4	INDS	Address of information supplied through the INDATASET parameter.
52(34)	4	OUTDS	Address of information supplied through the OUTDATASET parameter.
56(38)	4	FMNUM	Address of information supplied through the FROMNUMBER parameter
60(3C)	4	TONUM	Address of information supplied through the TONUMBER parameter
64(40)	4	*	Reserved—contains zeros.
68(44)	4	IENV	Address of this pointer itself if the ENVIRONMENT subparameter of the INFILE parameter is supplied.
72(48)	4	IRFMT	Reserved—contains zeros.
76(4C)	4	IBKSZ	Reserved—contains zeros.
80(50)	4	IRCSZ	Reserved—contains zeros.
84(54)	4	*	Reserved—contains zeros.
88(58)	4	IHDEV	Reserved—contains zeros.
92(5C)	4	IPDEV	Reserved—contains zeros.
96(60)	4	IFUNB	Reserved—contains zeros.
100(64)	4	IFBLK	Reserved—contains zeros.
104(68)	4	IVUNB	Reserved—contains zeros.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
108(6C)	4	IVBLK	Reserved—contains zeros.
112(70)	4	ISUNB	Reserved—contains zeros.
116(74)	4	ISBLK	Reserved—contains zeros.
120(78)	4	IUNDF	Reserved—contains zeros.
124(7C)	8	*	Reserved—contains zeros.
132(84)	4	OENV	Reserved—contains zeros.
136(88)	4	ORFMT	Reserved—contains zeros.
140(8C)	4	OBKSZ	Reserved—contains zeros.
144(90)	4	ORCSZ	Reserved—contains zeros.
148(94)	4	*	Reserved—contains zeros.
152(98)	4	OHDEV	Reserved—contains zeros.
156(9C)	4	OPDEV	Reserved—contains zeros.
160(A0)	4	OFUNB	Reserved—contains zeros.
164(A4)	4	OFBLK	Reserved—contains zeros.
168(A8)	4	OVUNB	Reserved—contains zeros.
172(AC)	4	OVBLK	Reserved—contains zeros.
176(B0)	4	OSUNB	Reserved—contains zeros.
180(B4)	4	OSBLK	Reserved—contains zeros.
184(B8)	4	OUNDF	Reserved—contains zeros.
188(BC)	8	*	Reserved—contains zeros.
196(C4)	4	IDUMY	Address of this pointer itself if the DUMMY subparameter of the ENVIRONMENT parameter is supplied for the input data set.
200(C8)	4	REP	Address of this pointer itself if the REPLACE parameter has been supplied.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
204(CC)	4	NREP	Address of this pointer itself if the NDREPLACE parameter has been supplied.
208(D0)	4	RUS	Address of this pointer itself if the REUSE parameter has been supplied.
212(D4)	4	NRUS	Address of this pointer itself if the NOREUSE parameter has been supplied.
216(D8)	4	ENCPR	Address of this pointer itself if the ENCIPHER parameter is supplied.
220(DC)	4	DECPR	Address of this pointer itself if the DECIPHER parameter is supplied.
224(EO)	4	INTKN	Address of information supplied through the INTERNALKEYNAME subparameter of the ENCIPHER parameter.
228(E4)	4	EXTKN	Address of information supplied through the EXTERNALKEYNAME subparameter of the ENCIPHER parameter.
232(E8)	4	PRIKY	Address of this pointer itself if the PRIVATEKEY parameter is supplied.
236(EC)	4	SHPKN	Address of information supplied through the SHIPKEYNAMES subparameter of the ENCIPHER parameter.
240(FO)	4	EDKV	Address of information supplied through the DATAKEYVALUE subparameter of the ENCIPHER parameter.
244(F4)	4	EDKF	Address of information supplied through the DATAKEYFILE subparameter of the ENCIPHER parameter.
248(F8)	4	STKN	Address of information supplied through the STOREKEYNAME subparameter of the ENCIPHER parameter.
252(FC)	4	STDK	Address of this pointer itself if the STOREKEYDATA parameter is supplied.
256(100)	4	NSTDK	Address of this pointer itself if the NOSTOREDATAKEY parameter is supplied.
260(104)	4	CPBLK	Address of information supplied through the CIPHERUNIT subparameter of the ENCIPHER parameter.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
264(108)	4	UDTA	Address of information supplied through the USERDATA subparameter of the ENCIPHER parameter.
268(10C)	4	SYSKY	Address of this pointer itself if the SYSTEMKEY parameter is supplied.
272(110)	4	SYSKN	Address of information supplied through the SYSTEMKEYNAME subparameter of the DECIPHER parameter.
276(114)	4	SYSDK	Address of information supplied through the SYSTEMDATAKEY subparameter of the DECIPHER parameter.
280(118)	4	DDKV	Address of information supplied through the DATAKEYVALUE subparameter of the DECIPHER parameter.
284(11C)	4	DDKF	Address of information supplied through the DATAKEYFILE subparameter of the DECIPHER parameter.
288(120)	4	INDS	Pointer to information supplied by the INDATASET parameter.
292(124)	4	IDENV	Address of this pointer itself if the ENVIRONMENT subparameter of the INDATASET is supplied.
296(128)	4	IDUM	Address of this pointer itself if the DUMMY subparameter of the ENVIRONMENT parameter is supplied for the INDATASET parameter.
300(12C)	24	*	Reserved—contains zeros.
324(144)	4	INDD	Address of information supplied by the FILE parameter.
328(148)	4	MRGC	Address of this pointer itself if the MERGECAT parameter is supplied.
332(14C)	4	NMRGC	Address of this pointer itself if the NOMERGECAT parameter is supplied.
336(150)	4	ENT	Address of information supplied through the ENTRIES parameter.
340(154)	4	LVL	Address of information supplied through the LEVEL parameter.

RESETCAT FDT

Offset	Content	
0 (0)	RESE	TCAT
8 (8)	↑ CATALOG	↑ catname/password
16 (10)	↑ dname	↑ MASTERPW
24 (18)	↑ WORKFILE	↑ WORKCAT
32 (20)	↑ IGNORE	↑ NOIGNORE
40 (28)	↑ CRAFILES count	↑ dname
48 (30)	↑ ALL	↑ NONE
56 (38)	↑ CRAVOLUMES count	↑ volser
64 (40)	↑ devtype	↑ dname/password

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb—RESETCAT.
8(8)	4	CAT	Address of this parameter itself if the CATALOG parameter has been supplied.
12(C)	4	CATNM	Address of the information supplied through the *catname/password* subparameter of the CATALOG parameter.
16(10)	4	CATDN	Address of information supplied through the *dname* subparameter of the CATALOG parameter.
20(14)	4	MRPW	Address of information supplied through the *password* subparameter of the MASTERPW parameter.
24(18)	4	WFDN	Address of this parameter itself if the WORKFILE parameter is supplied.
28(1C)	4	WCAT	Address of the information supplied through the *catname/password* subparameter of the WORKCAT parameter.
32(20)	4	IGN	Address of this parameter itself if the IGNORE parameter is supplied.
36(24)	4	NIGN	Address of this parameter itself if the NOIGNORE parameter is supplied.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
40(28)	4	CFILE	Count of the number of CRAs that were specified through the CRAFILES parameter.
44(2C)	4	CRADNPTR	Address of an array of pointers. Each pointer points at a *dname* for the CRA it relates to in the order that they appear in the CRAFILES parameter.
48(30)	4	ALLPPTR	Address of an array of pointers. Each pointer points to itself if ALL was specified for the related CRA in the CRAFILES parameter.
52(34)	4	NONEPTR	Address of an array of pointers. Each pointer points to itself if NONE was specified for the related CRA in the CRAFILES parameter.
56(38)	4	CVOL	Address of this parameter itself if the CRAVOLUMES parameter is supplied.
60(3C)	4	CRAVLPTR	Address of an array of pointers. Each pointer points to the information supplied by the *volser* subparameter of the CRAVOLUMES parameter.
64(40)	4	CRADVPTR	Address of an array of pointers. Each pointer points to the information supplied by the *devtype* subparameter of the CRAVOLUMES parameter.
68(44)	4	WFILE	Address of the information supplied by the *dname/password* subparameter of the WORKFILE parameter.

SETCACHE FDT

Offset	Content	
0 (0)	SETC	ACHE
8 (8)	↑ FILE	↑ VOLUME
16 (10)	↑ UNIT	↑ DEVICE
24 (18)	↑ SUBSYSTEM	↑ ON
32 (20)	↑ OFF	

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb—SETCACHE.
8(8)	4	FILE	Address of information supplied through the FILE parameter.
12(C)	4	VOL	Address of information supplied through the VOLUME parameter.
16(10)	4	UNT	Address of information supplied through the UNIT parameter.
20(14)	4	DEV	Address of this pointer itself if the DEVICE parameter has been supplied.
24(18)	4	SSYS	Address of this pointer itself if the SUBSYSTEM parameter has been supplied.
28(1C)	4	SCON	Address of this pointer itself if the ON parameter has been supplied.
32(20)	4	SCOFF	Address of this pointer itself if the OFF parameter has been supplied.

VERIFY FDT

Offset	Content	
0 (0)	V E R I	F Y b b
8 (8)	↑FILE	↑DATASET

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	FDTVERB	Verb aligned left and padded with blanks—VERIFYƀ ƀ.
8(8)	4	IN	Address of information supplied through the FILE parameter.
12(C)	4	INDAT	Address of information supplied through the DATASET parameter.

GDT

Global Data Table

The GDT is the directory for the services and data areas of the processor. It contains a branch vector for the services provided by the system adapter, the I/O adapter, and the text processor. It also points to the invoker's parameter list, trace tables, and historical tables. The GDT is always the first parameter passed to any routine. The GDT is contained in the storage associated with module IDCSA01.

Created by	Modified by	Used by	Size
IDCSA01	All service routines	All routines	292

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	GDTHDR	Global data table header; contains 'GDTb̸'.
4(4)	4	GDTPRM	Address of parameter list from invoker of the processor.
8(8)	4	GDTTR1	Address of inter-module trace table.
12(C)	4	GDTTR2	Address of intra-module trace table.
16(10)	4	GDTDBH	Address of debugging-aids historical area.
20(14)	4	GDTSTH	Reserved—contains zeros.
24(18)	4	GDTRIH	Address of reader/interpreter historical area.
28(1C)	4	GDTTPH	Address of text processor historical area.
32(20)	4	GDTSAH	Address of system adapter historical area.
36(24)	4	GDTIOH	Address of I/O adapter historical area.
40(28)	4	GDTDBG	Address of entry point for dump routine, IDCDB01 (UDUMP macro).
44(2C)	4	GDTSTC	Reserved—contains zeros.
48(30)	4	GDTPRT	Address of entry point to print, IDCIOPR (UPRINT macro).

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
52(34)	4	GDTESS	Address of entry point to establish PCT from text structure, IDCTPES (UESTS macro).
56(38)	4	GDTESA	Address of entry point to establish PCT from storage, IDCTPEA (UESTA macro).
60(3C)	4	GDTRST	Address of entry point to modify PCT, IDCTPRS (UREST macro).
64(40)	4	GDTRES	Address of entry point to reset PCT, IDCTPRE (URESET macro).
68(44)	4	GDTCAL	Address of entry point to call, IDCSACL (UCALL macro).
72(48)	4	GDTGSP	Address of entry point to get space, IDCSAGS (UGSPACE macro).
76(4C)	4	GDTFSP	Address of entry point to free storage, IDCSAFS (UFSPACE macro).
80(50)	4	GDTGPL	Address of entry point to get storage, IDCSAGP (UGPOOL macro).
84(54)	4	GDTFPL	Address of entry point to free storage, IDCSAFP (UFPOOL macro).
88(58)	4	GDTLOD	Address of entry point to load module, IDCSALD (ULOAD macro).
92(5C)	4	GDTDEL	Address of entry point to delete module, IDCSADE (UDELETE macro).
96(60)	4	GDTPRL	Address of entry point for prolog, IDCSAPR.
100(64)	4	GDTEPL	Address of entry point for epilog, IDCSAEP (UEPIL macro).
104(68)	4	GDTTIM	Address of entry point for time, IDCSATI (UTIME macro).
108(6C)	4	GDTIIO	Address of entry point for I/O initialization, IDCIOIT (UIOINIT macro).
112(70)	4	GDTTIO	Address of entry point for I/O termination, IDCIOTM (UIOTERM macro).
116(74)	4	GDTRIP	Address of reader/interpreter name—either 'IDCRI01' or 'IDCRI04'.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
120(78)	4	GDTTOH	Reserved—contains zeros.
124(7C)	4	GDTOPN	Address of entry point to open data sets, IDCIOOP (UOPEN macro).
128(80)	4	GDTCLS	Address of entry point to close data sets, IDCIOCL (UCLOSE macro).
132(84)	4	GDTGET	Address of entry point to get a logical record, IDCIOGT (UGET macro).
136(88)	4	GDTPUT	Address of entry point to put a logical record, IDCIOPT (UPUT macro).
140(8C)	4	GDTPOS	Address of entry point to position to a logical record, IDCIOPO (UPOSIT macro).
144(90)	4	GDTCPY	Address of entry point to copy logical records, IDCIOCO (UCOPY macro).
148(94)	4	GDTCAT	Address of entry point for manipulating VSAM or ICF catalog, IDCSACA (UCATLG macro).
152(98)	4	GDTABT	Address to abort, SAABT in IDCSA02 (UABORT macro).
156(9C)	4	GDTABH	Address of UABORT register save area.
160(A0)	4	*	Reserved—contains zeros.
164(A4)	4	GDTSNP	Address of entry point to snap dump, IDCSASN (USNAP macro).
168(A8)	4	GDTSPR	Address of IDCSA03's storage.
172(AC)	4	GDTVFY	Address of entry point to VERIFY data set, IDCIOVY (UVERIFY macro).
176(B0)	4	GDTCMB	Address of TSO command buffer obtained from TSO.
180(B4)	4	GDTUPT	Address of TSO user profile table obtained from TSO.
184(B8)	4	GDTPSB	Address of TSO protected step block obtained from TSO.
188(BC)	4	GDTECT	Address of TSO environment table obtained from TSO.
192(C0)	4	GDTUID	Address of entry point to obtain TSO user's identification, IDCSAID (UID macro).

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
196(C4)	4	GDTPMT	Address of entry point to obtain information from TSO user IDCSAPT (UPROMPT macro).
200(C8)	4	GDTCIR	Address of entry point to get fully qualified data set names IDCSACR (UCIR macro).
204(CC)	4	GDTLNK	Address of entry point to link to module IDCSALK (ULINK macro).
208(D0)	4	GDTALC	Address of entry point to allocate a data set or mount a volume, IDCSAAL (UALLOC macro).
212(D4)	4	GDTDLC	Address of entry point to deallocate a data set or demount a volume, IDCSADL (UDEALLOC macro).
216(D8)	4	GDTQAL	Address of entry point to construct a fully qualified data set name in TSO, IDCSAQL (UQUAL macro).
220(DC)	4	GDTSTW	Address of entry point to stow a member name of a partitioned data set, IDCIOST (USTOW macro).
224(E0)	4	GDTSSC	Address of entry point to perform mass storage control (MSC) functions, IDCSA06 (USSC macro).
228(E4)	4	GDTENQ	Address of entry point to acquire control of a resource, IDCSANQ (UENQ macro).
232(E8)	4	GDTRSV	Address of entry point to acquire control of an I/O unit, IDCSARV (URESERVE macro).
236(EC)	4	GDTDEQ	Address of entry point to release a resource IDCSADQ (UDEQ macro).
240(F0)	4	GDTSFO	Address of entry point to obtain system control block information, IDCSASI (USYSINFO macro).
244(F4)	4	GDTWTO	Address of entry point to write a message to the console operator, IDCSAWO (UWTO macro).
248(F8)	4	GDTSCR	Address of entry point to delete a direct-access data set, IDCSASC (USCRATCH macro).
252(FC)	4	GDTUNT	Address of entry point to mount, demount, post, or check a volume, IDCSA06 (UMSSUNIT macro).
256(100)	4	GDTRCT	Address of entry point to recatalog a non-VSAM data set, IDCSARC (URECAT macro).

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
260(104)	4	GDTIFO	Address of entry point to obtain data set information, IDCIOSI (UIOINFO macro).
264(108)	4	GDTEXP	Address of entry point to open, close, read, or write a volume with EXCP, IDCIO05 (UEXCP macro).
268(10C)	4	GDTSTA	Address of entry point to perform ABEND recovery, IDCSA10 (USTAE macro).
272(110)	4	GDTERR	Address of entry point to convert numeric return codes (UERROR macros).
276(114)	4	GDTUNC	Address of entry point to uncatalog and scratch a data set, IDCSAUC (UUNCATLG macro).
280(118)	4	GDTLOC	Addres of entry point to locate a VSAM data set, IDCSALC (ULOCATE macro).
284(11C)	4	GDTRCK	Address of entry point to obtain RACF authorization, IDCSA08 (URACHECK macro).
288(120)	4	GDTCRP	Address of entry point to encipher/decipher data (UCRYPT macro).
292(124)	4	GTDOBT	Address of entry point to obtain a DSCB by data set name or CCHHR (UOBTAIN macro).

ILST

INCLUDE/EXCLUDE List

The ILST contains the INCLUDE/EXCLUDE list for DIAGNOSE.

Created by	Modified by	Used by	Size
IDCDA02	IDCDA01 IDCDA02 IDCDA03	IDCDA01 IDCDA02 IDCDA03	Variable

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	ILSTID	ID='CLST'.
4(4)	1	*	Reserved.
5(5)	3	ILSTLEN	Length of ILST.
8(8)	4	ILSTAVAL	Address of next ILST entry.
12(C)	4	ILSTLAST	Address of last ILST entry.
16(10)	4	ILSTNEXT	Address of next ILST block.
20(14)	var.	ILSTNAMS	Entries of INTRY.

Each ILSTNAMS entry contains the following:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	3	INHDR	Header define.
	1	INFLAG	Flag bits.
	1...	INFOUND	Flag bits.
	.1..	INGEN	Flag bits.
	..1.	INVL	Flag bits.
	...x xxxx	*	Reserved.
	1	INPOS	Asterisk position.
	1	INMLL	Length of the name.
	var.	INNAME	Name 1 through 44 bytes.

IPT

Input Parameter Table

The input parameter table is a parameter list passed by IDCRC01 to IDCRC02 within EXPORTRA. It is an array of five pointers. Each object pointed to is described after the IPT pointers.

Created by	Modified by	Used by	Size
IDCRC01	IDCRC02	IDCRC02	20

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4		Address of control block describing the object to be exported.
4(4)	4		Address of control block describing the output (portable) data set.
8(8)	4		Address of the input data set name.
12(C)	4		Address of the output data set name.
16(10)	4		Address of the environment parameter.

Description of control block describing object to be exported.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	OBJTYP	Type of object.
1(1)	.3	OBJVAL	The catalog control interval number of the entry.
4(4)	4	RESINP	Reserved.
8(8)	1	OBJPLN	Password length.
9(9)	8	OBJPAS	Password.

Description of control block describing output (portable) data set.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)		OUTLEN	Maximum record length of data component.
4(4)		SAVOIOCS	Pointer to output IOCS.
8(8)		USBKSZ	User supplied output block size.
12(C)	4	RESOUTP	Reserved.
16(10)	1	OUTFLGS	Status of output data set.
	1...	OPNFLG	This flag is on if output data set is open.
	.1..	ENDFLG	This flag is on if this is the last request.
	..1.	EMPTYDS	This flag is on if the object contains no data records.

The third pointer in the IPT points to an 8-byte input dname.

The fourth pointer in the IPT points to an 8-byte output dname.

The fifth pointer in the IPT points to an 8-byte field describing the prime data device (PDEV subparameters).

IODATA

I/O Adapter Historical Area

The I/O adapter historical area is pointed to by GDTIOH. It is built on the first call to the I/O adapter (UIOINIT Umacro), and contains information that is common to all modules of the I/O adapter.

Created by	Modified by	Used by	Size
IDCIO01	IDCIO01	IDCIO01 IDCIO02	68

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	IODIOC	First IOCSTR in chain.
4(4)	4	IODMSG	Address of an area for VSAM messages.
8(8)	4	IODADD	Address of the alternate DD list.
12(C)	4	IODXTN	Address of the external I/O routine list.
16(10)	4	IODSID	Identifier containing:
	2	IODMID	Module identifier.
	. . 2	IODINC	Pool identifier.
20(14)	4	IODIEV	Address of input end-of-volume exit routine.
24(18)	4	IODOEV	Address of output end-of-volume exit routine.
28(1C)	4	*	Reserved—contains zeros.
32(20)	4	IODEOD	Address of end-of-data routine for non-VSAM data sets.
36(24)	4	IODOSS	Non-VSAM input SYNAD routine address.
40(28)	4	IODOSO	Non-VSAM output SYNAD routine address.
44(2C)	4	IODICS	Address of the access method services input IOCSTR.
48(30)	4	IODOCS	Address of the access method services output IOCSTR.
52(34)	4	IODIOX	Address of the EXCP control block chain.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
56(38)	4	*	Reserved—contains zeros.
60(3C)	4	IODAEI	Address of VSAM EODAD routine.
64(40)	4	*	Reserved—contains zeros.

IOCSTR

Input/Output Communications Structure

An IOCSTR exists for each open data set, or for any on which an open has been attempted. It contains all information about the data set that may be required by the processor. An IOCSTR is built at open time, and a pointer to the IOCSTR is returned to the requester of the open, in the OPNIOC field of the OPNAGL. A UGPOOL area immediately precedes the IOCSTR. The UGPOOL area contains the identifier assigned to the data set by the I/O adapter. All other requests for I/O service include this IOCSTR as one of the parameters for the request.

Created by	Modified by	Used by	Size
IDCIO02	All routines	All routines	68

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
-4(-4)	4	*	Always contains 'IOCS'.
0(0)	4	IOCDAD	Address of data area.
4(4)	4	IOCDLN	Length of data record.
8(8)	4	IOCTRN	Transmission length: LRECL for logical processing or control interval for block processing.
12(C)	1	IOCKYL	Key length in bytes.
13(D)	. 3	IOCRKP	Relative key position, value assumes VSAM or ISAM meaning.
16(10)	1	IOCDSO	Data set organization:
	1...	IOCDSOAM	VSAM data set.
	.1..	IOCDSOPS	Non-VSAM sequential data set (QSAM or BSAM).
	..1.	IOCDSOIS	Indexed sequential (ISAM) data set.
	...1	IOCDSOPO	Partitioned (BPAM) data set.
17(11)	. 1	IOCRFM	Non-VSAM record format:
	1...	IOCRFMFX	Fixed-length records.
	.1..	IOCRFMVR	Variable-length records, not spanned.
	..1.	IOCRFMUN	Undefined-length records.
	...1	IOCRFMSF	Spanned records.
	.... 1...	IOCRFMBK	Blocked records.

	Bytes and		Description:
Offset	Bit Pattern	Field Name	Content, Meaning, Use
18(12)	.. 1	IOCMAC	Macro form used:
	1...	IOCMACIN	Input processing.
	.1..	IOCMACOT	Output processing.
	..1.	IOCMACUP	Update processing.
	...0	IOCMACCR	Keyed sequence (VSAM).
	...1		Entry sequence (VSAM).
	.... 0...	IOCMACBK	Logical records (QSAM or QISAM).
	.... 1...		Blocks or control intervals (BSAM, BPAM, or VSAM).
	.... .0..	IOCMACDR	Sequential data set.
	.... .1..		Direct data set.
	.... ..1.	IOCMACCC	Copy catalog.
	1	IOCMACIX	Use GETIX/PUTIX
19(13)	...1	IOCMAC2	
	1...	IOCMACSK	Skip sequential processing.
	.1..	IOCMACAS	Asynchronous processing.
	..1.	IOCMACRR	Relative record processing.
	...1	IOCMACCP	Change processing.
	.... 1...	IOCMACEN	PUT-ENDREQ processing.
	.... .1..	IOCMACPA	Reprocessing flag.
	.... ..1.	IOCMACER	PUT-ERASE processing.
	1	IOCMACNT	NOTE processing.
20(14)	1	IOCCHP	Change processing mode:
	1...	IOCCHPSQ	Change to sequential.
	.1..	IOCCHPDR	Change to direct.
	..1.	IOCCHPSK	Change to skip sequential.
	...1	IOCCHPKS	Change to keyed.
	.... 1...	IOCCHPCR	Change to addressed.
	.... .1..	IOCCHPBK	Change to control interval.
	.... ..1.	IOCCHPUP	Change to update.
	1	IOCCHPNU	Change to no update.
21(15)	. 1	IOCMSG	Message flags:
	1...	IOCCHPKE	Change to key equal.
	.1..	IOCCHPKG	Change to key greater than or equal.
	..1.	IOCMSGOP	Data set is open for processing.
	...1	IOCMSGOE	VSAM open error.
	.... 1...	IOCMSGCE	VSAM close error.
	.... .1..	IOCMSGAE	VSAM error other than open or close.
	.... ..1.	IOCMSGSM	Suppress logical error messages.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
22(16)	..6	IOCVOLSR	Volume serial number of the opened data set.
28(1C)	4	IOCHURBA	High used RBA.
32(20)	4	IOCDSN	Address of data set name. The data set name usually follows the IOCSTR extension.
36(24)	4	IOCCBP	Address of ACB or DCB if OPNMODRC is set.
40(28)	4	*	Either IOCRBA, IOCTTR, or IOCVRC:
	4	IOCRBA	VSAM record relative byte address (RBA).
	4	IOCTTR	Track and record number (TTR) for UPOSIT macro.
	4	IOCVRC	VSAM error codes.
44(2C)	4	*	Either IOCMEM or IOCKYA:
	4	IOCMEM	Address of member name for BPAM data set.
	4	IOCKYA	Address of key.
48(30)	4	*	Either IOCNWM or IOCPTL with IOCPNM:
	4	IOCNWM	Address of new member name for BPAM data set.
	2	IOCPTL	Length of key supplied for position request.
	. . 2	IOCPNM	Number of stacked puts.
52(34)	4	IOCRRN	Relative record number.
56(38)	4	IOCWORK	Address of input work area.
60(3C)	4	IOCREL	Relative record number.
64(40)	4	IOCEXT	IOCSTR extension address.

IOCSEX

Input/Output Communications Structure Extension

The IOCSTR communications structure extension is built immediately after the IOCSTR. However, for flexibility and to make the IOCSTR easily extensible, field IOCEXT points to the IOCSEX.

Created by	Modified by	Used by	Size
IDCIO02	IDCIO01	IDCIO01	45

	Bytes and		Description:
Offset	Bit Pattern	Field Name	Content, Meaning, Use
0(0)	4	IOCCBA	Address of ACB or DCB.
4(4)	4	*	Either IOCRPL or IOCDEC:
	4	IOCRPL	Address of VSAM RPL.
	4	IOCDEC	Address of update DECB.
8(8)	2	IOCCBL	Length of ACB or DCB.
10(A)	. . 2	IOCLRP	Length of RPL.
12(C)	4	IOCWKA	Address of input work area.
At decimal displacements 16 and 20, one of the two following sets of fields appears:			
Set one:			
16(10)	4	IOCXAD	User routine address.
20(14)	4	IOCXPM	User routine parameter address.
Set two:			
16(10)	4	IOCEXA	VSAM exit list address.
20(14)	2	IOCEXL	VSAM exit list length.
22(16)	. . 2	*	Reserved—contains zeros.

The data area then continues as follows.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
24(18)	4	IOCNIO	Address of next IOCSTR in chain.
28(1C)	4	IOCSID	Storage pool identifier.
32(20)	1	IOCFLG	Extension flags:
	1...	IOCFLGEX	User controlled data set.
	.1..	IOCFLGDF	Data set defined by JCL.
	..1.	IOCFLGEF	End-of-file on user data set.
	...1	IOCFLGIO	DD * or SYSOUT data set.
	.... 1...	IOCFLGOP	Data set is open.
	.... .x..	IOCFLGOE	Reserved—contains zeros.
	.... ..1.	IOCFLGSP	Access method services print data set.
	1	IOCFLGFB	Force block size.
33(21)	. 1	IOCDEV	Device type flags:
	1...	IOCDEVDA	Direct access.
	.1..	IOCDEVMT	Magnetic tape.
	..1.	IOCDEVUR	Unit record.
34(22)	. . 1	IOCINF	Information flags:
	.1...	IOCINFPT	Point has been issued.
	.1..	IOCINFAE	ABEND exit taken.
	..x.	IOCINFND	Reserved—contains zeros.
	...x	IOCINFQX	Reserved—contains zeros.
	.... 1...	IOCINFAC	ANSI control character.
	.... .x..	IOCINFDO	Reserved in OS/VS—contains zeros.
	.... ..1.	IOCINFCT	Opened as a catalog.
	x	IOCINFR1	Reserved—contains zeros.
35(23)	. . . 1	IOCMOD	Additional information flags:
	1...	IOCMODPD	Process dummy (deleted) records.
	.1..	IOCMODRR	Return the RPL address.
	..1.	IOCMODDY	Dynamically allocated data set.
	...1	IOCMODRG	DSORG specified.
	.... 1...	IOCMODUB	User buffering.
	.... .1..	IOCMODXM	Export/import.
	.... ..1.	IOCMODRP	Replace processing.
	1	IOCMODEX	Exclusive control.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
36(24)	4	IDOCDDN	Pointer to ddname.
40(28)	2	IOCDNM	Reserved in OS/VS—contains zeros.
42(2A)	. . 2	*	Reserved—contains zeros.
44(2C)	1	IOCRCV	Recovery flags.
	1...	IOCRCVXM	VSAM recovery flag.
	.1..	IOCRCVRA	Open CRA.

IOXCTLBK

Input/Output Control Block for EXCP

The IOXCTLBK is built when a data set is opened with UEXCP. It is used throughout all processing until the data set is closed with UEXCP.

Created by	Modified by	Used by	Size
IDCIO05	IDCIO05	IDCIO05	Variable

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	IOXCHAIN	Contains either 0 or the address of the previous IOXCTLBK.
4(4)	8	IOXID	Identifies the control block in a dump with the characters 'IOXCTLBK'.
12(C)	4	IOXWRITE	Contains the number of new records written.
16(10)	4	IOXREADS	Contains the number of records read.
20(14)	2	IOXACODE	Contains the ABEND code for OPEN/CLOSE.
22(16)	..2	IOXRCODE	Contains the return code for OPEN/CLOSE.
24(18)	1	IOXTYPE	Contains the type of EXCP open request: X'10' for volume label, X'20' for MSC tables, X'30' for password checking, and X'40' for new volume.
25(19)	.1	IOXFLGS	Flags:
	1...	IOXOPN	DCB opened.
	.1..	IOXABEND	Password abend exit entered.
	..1.	IOXCLOSE	DCB closed.
	...1	IOXSYNAD	SYNAD error occurred.
26(1A)	..2	*	Reserved—contains zeros.
28(1C)	8	IOXDDN	8-byte ddname.
36(24)	4	IOXDCBID	Identifies the DCB in a dump with the characters 'DCBb̸'.
40(28)	*v*	IOXDCB	Contains the DCB for the data set.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	4	IOXECBID	Identifies the control block in a dump with the characters 'ECBƀ'.
	8	IOXECB	Contains the code posted by the I/O supervisor and the address used by the I/O supervisor.
	4	IOXIOBID	Identifies the control block in a dump with the characters 'IOBƀ'.
	v	IOXIOB	Contains the I/O control block used by EXCP.
	4	IOXJFCBI	Identifies the job file control block (JFCB) in a dump with the characters 'JFCB'.
	200	IOXJFCB	Contains the job file control block (JFCB).
40(28)	4	IOXEXID	Identifies the control block in a dump with the characters 'XLST'.
	1	*	Reserved—contains zeros.
	.3	IOXJFCBP	Address of the altered job file control block (JFCB).
	4	*	Reserved—contains zeros.
	4	IOXUCBP	Address of the UCB.
	4	IOXCCWID	Identifies the control block in a dump with the characters 'CCWƀ'.
	1000	IOXCCW5	Array containing the CCWs.
	400	IOXCOUNT	Array containing record count fields for records to be written.

Inter-Module Trace Table

The inter-module trace table contains information on the flow of control between modules. The table is pointed to by GDTTR1. The oldest identifier is at the beginning of the table. The latest identifier is at the end of the table. Each time a UPROL or UEPIL macro is issued the oldest identifier is removed and the new identifier is added at the end. A UPROL adds the identifier of the current module. A UEPIL adds the identifier of the module to which control is being returned. The UDUMP macro prints the table on SYSPRINT.

Created by	Modified by	Used by	Size
IDCSA01	UEPIL UPROL macros	IDCDB01	100

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
-6(-6)	6	*	Table identification INTERɓ.
0(0)	100	*	Inter-module trace table with 20 entries.

Each entry contains the following:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	4	*	Identifier provided by module issuing UEPIL or UPROL macros. The identifier is the last 4 characters of the module name.
	1	*	Blank 'ɓ'.

Intra-Module Trace Table

The intra-module trace table contains information on the flow of control within modules. The table is pointed to by GDTTR2. The oldest identifier is at the beginning of the table. The latest identifier is at the end of the table. Each time a UTRACE is issued the oldest identifier is removed and the new identifier is added at the end. The UDUMP macro prints the table on SYSPRINT.

Created by	Modified by	Used by	Size
IDCSA01	UTRACE macro	IDCDB01	100

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
-6(-6)	6	*	Table identification INTRAb̸.
0(0)	100	*	Intra-module trace table with 20 entries.

Each entry contains the following:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	4	*	Identifier provided by module issuing UTRACE. The first 2 characters are the mnemonic identifier which are characters 4 and 5 of the module name. For example, EX refers to the executive.
	1	*	Blank 'b̸'.

LLBLK

Load List Block

The load list block contains an entry for each module loaded by UCALL, ULOAD, and ULINK. It is used to control the loading and deleting of modules used by access method services.

Created by	Modified by	Used by	Size
IDCSA01 IDCSA02	IDCSA02 IDCSA03	IDCSA02 IDCSA03	392

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	LLFSTSLT	Address of first available slot; zero if no slots available.
4(4)	4	LLNXTBLK	Pointer to next load list block.
8(8)	16	LLSLOT	Load list slot (24 per block).

Each available inactive slot contains the following:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
8(8)	4	LLNXTSLT	Pointer to next available slot; zero if last slot in block.
12(C)	4	LLNOMOD	Contains binary zeros.

Each active slot contains the following:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
8(8)	8	LLNAME	Name of loaded module in EBCDIC.
16(10)	4	LLADDR	Entry point or address of loaded module.
20(14)	1	LLUSECTR	Module use count; if zero, module is not in use.
21(15)	3	LLMODSZ	Size of module.

LCTINFO

Locate Data Set Return Information Area

LCTINFO is an area a caller passes when a ULOCATE macro is issued. Information about a non-VSAM data set is returned to the caller in this area.

Created by	Modified by	Used by	Size
Calling routine	IDCSA07	IDCSA07	33

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	LCTHEAD	Initialized with 'LCTINFO' by ULOCATE.
8(8)	4	LCTMULVC	Number of volumes for this data set entry in the catalog.
12(C)	4	LCTACBP	Pointer to the ACB.
16(10)	4	LCTCREAT	VSAM creation date.
20(14)	4	LCTEXPIR	VSAM expiration date.
24(18)	8	OCTOWNER	Catalog entry owner name.
32(20)	1	LCTFLAGS	Return flags.
	1...	LCTVSCAT	VSAM or ICF catalog entry.

IDCRIKT

Modal Verb and Keyword Symbol Table

Load module IDCRIKT contains the modal verb and keyword symbol table, which acts as the "command descriptor" for the modal commands.

Created by	Modified by	Used by	Size
IBM-supplied	None	IDCRI01	90

	Bytes and		Description:
Offset	**Bit Pattern**	**Field Name**	**Content, Meaning, Use**
0(0)	1	PARMSMLN	Length of PARM character string.
1(1)	.9	PARMSYM	PARM character string.
10(A)	..1	SETSMLN	Length of SET character string.
11(13)	...9	SETSYM	SET character string.
20(14)	1	IFSMLN	Length of IF character string.
21(15)	.9	IFSYM	IF character string.
30(1E)	..1	THENSMLN	Length of THEN character string.
31(1F)	...9	THENSYM	THEN character string.
40(28)	1	ELSESMLN	Length of ELSE character string.
41(29)	.9	ELSESYM	ELSE character string.
50(32)	..1	DOSMLN	Length of DO character string.
51(33)	...9	DOSYM	DO character string.
60(3C)	1	ENDSMLN	Length of END character string.
61(30)	.9	ENDSYM	END character string.
70(46)	..1	LSTCCLN	Length of LASTCC character string.
71(47)	...9	LSTCCSYM	LASTCC character string.

Offset	**Bytes and Bit Pattern**	**Field Name**	**Description: Content, Meaning, Use**
80(50)	1	MAXCCLN	Length of MAXCC character string.
81(51)	. 9	MAXCCSYM	MAXCC character string.

MDAGL

Mount/Demount Argument List

The MDAGL is passed when a UMSSUNIT macro is issued with a mount or demount request. It defines a request to mount or demount a mass storage volume.

Created by	Modified by	Used by	Size
Calling routine	IDCSA06	IDCSA06	31

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	MDHEAD	Set by IDCSA06 with 'MOUNTb̸ b̸ b̸' for a mount request or 'DEMOUNTb̸' for a demount request.
8(8)	4	MDUCBPTR	Address of an area containing the UCB address or zeros if the UCB address is unknown. If the UCB address is zeros, the UCB address will be put in the area upon return.
12(C)	8	MDDDNAME	ddname of the volume.
20(14)	4	MDPUAGL	Address of an initialized PUAGL argument list if bits MDPOST or MDCLEAR are set.
24(18)	6	MDLABEL	Volume serial number recorded in the MSC tables.
30(1E)	..1	MDFLAGS	Mount/demount options:
	1...	MDNEWVOL	Indicates that the volume to be mounted does not have a volume label or a VTOC. This field is ignored on a demount request.
	.1..	MDCLEAR	Indicates that the caller wants to clear the volume serial number from a UCB prior to the mount function. This field is ignored on a demount request.
	..1.	MDPOST	Indicates that the caller wants to post the volume serial number in a UCB after the demount function. This field is ignored on a mount request.
	...1	MDWAIT	Indicates that the caller wants a demount with delayed response to be issued. This field is ignored on a mount request.
	.... 1...	MDENQ	Indicates that an enqueue was done by IDCSA06 as part of a mount request.
	.... .1..	MDDEQ	Indicates that the volume should be dequeued when demounted.

OBTAGL

Obtain Argument List

The OBTAGL is passed when a UOBTAIN macro is issued. It defines an OBTAIN request having a 44-byte data set or a 5-byte CCHHR as its search argument.

Created by	Modified by	Used by	Size
Calling routine	IDCIO03	IDCIO03	16

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	OBTOPT	Obtain options.
	1...	OBTOPTSN	Search by data set name.
	.1..	OBTOPTSA	Seek address.
	..xx xxxx		Reserved.
1(1)	3	*	Reserved.
4(4)	4	OBTDIDPT	Pointer to 44-byte data set name or 5-byte CCHHR.
8(8)	4	OBTVSRPT	Pointer to volume serial number.
12(C)	4	OBTWKAPT	Pointer to work area.

OPNAGL

Open Argument List

The OPNAGL defines a request to open a data set. The address of the OPNAGL is passed as a parameter to the I/O adapter from any routine that requires the open function.

Created by	Modified by	Used by	Size
Routine that requests an open	IDCIO02	IDCIO02	48

	Bytes and		Description:
Offset	Bit Pattern	Field Name	Content, Meaning, Use
0(0)	1	OPNOPT	Open options (determine data set usage):
	1...	OPNOPTIN	Input data set.
	.1..	OPNOPTOT	Output data set.
	..1.	OPNOPTUP	Update mode of processing.
	...1	OPNOPTBK	Block processing.
	.... 1...	OPNOPTKS	Keyed processing.
	.... .1..	OPNOPTCR	Addressed processing.
	.... ..1.	OPNOPTDR	Direct processing.
	1	OPNOPTSK	Skip sequential processing.
1(1)	. 1	OPNRFM	Non-VSAM output record format. Required in DOS/VS; optional in OS/VS:
	1...	OPNRFMFX	Fixed.
	.1..	OPNRFMVR	Variable.
	..1.	OPNRFMUN	Undefined.
	....1	OPNRFMSF	Spanned.
	.... 1...	OPNRFMBK	Blocked.
2(2)	. . 1	OPNTYP	Data set type:
	1...	OPNTYPSI	System input (SYSIN) is to be opened. OPNIOC is the only other required field.
	.1..	OPNTYPSO	System output (SYSPRINT) is to be opened. OPNIOC is the only other required field.
	..1.	OPNTYPCI	Catalog to be opened.
	...1	OPNTYPXM	Export/import data set to be opened.
	.... 1...	OPNTYPRA	Catalog recovery area.
	.... .1..	OPNTYPEX	Data set is to be opened for exclusive control.
	.... ..1.	OPNTYPRV	VSAM recovery processing.

	Bytes and		Description:
Offset	Bit Pattern	Field Name	Content, Meaning, Use
	1	OPNTYPSY	Bypass security checking.
3(3)	. . . 1	OPNMOD	Open modifiers:
	1...	OPNMODPD	Process ISAM dummy (deleted) records.
	.1..	OPNMODAC	ANSI control character.
	..1.	OPNMODRC	Return control block address.
	...1	OPNMODRR	Return RPL address.
	.... 1...	OPNMODAX	Open alternate index.
	.... .1..	OPNMODRS	Open with reset.
	.... ..1.	OPNMODUB	User buffering.
	1	OPNMODRP	Replace processing.
4(4)	4	OPNIOC	Address of pointer of IOCSTR. This field is always present. After a successful open, the pointer contains the address of the IOCSTR built by the I/O adapter.
8(8)	4	OPNDDN	Address of 8-byte ddname (not present when SYSIN or SYSPRINT is being opened but required at all other times).
12(C)	4	OPNPWA	Address of an optional 8-byte password, used only with VSAM data sets.
16(10)	4	OPNDSN	Address of 44-byte data set name.
20(14)	4	OPNAGLEX	Address of the open argument list extension.
24(18)	4	OPNDEVDT	Address of data device type.
28(1C)	4	OPNDEVIX	Address of index device type.
32(20)	4	OPNREC	Logical record length, optional.
36(24)	4	OPNBLK	Block size, optional.
40(28)	1	OPNKYL	Key length.
41(29)	. 1	OPNDSO	Data set organization:
	1...	OPNDSOAM	VSAM.
	.1..	OPNDSOPS	Non-VSAM.
	..1.	OPNDSOIS	Indexed sequential (ISAM).
	...1	OPNDSOPO	Partitioned (BPAM).
42(2A)	. . 1	OPNOPT2	Second option byte.
	1...	OPNOPTAS	Asynchronous processing.
	.1..	OPNOPTNM	DSNAME supplied.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	..1.	OPNOPTJM	Modify JFCB.
	...1	OPNOPTFB	Force blocksize
	.... 1...	OPNNOVER	Suppress OPEN VERIFY.
43(2B)	. . . 1	OPNSTRNO	Number of strings.
44(2C)	4	OPNVOL	Pointer to volume serial number.

OPNAEXT

Open Argument List Extension

The OPNAGL extension is usually built immediately after the OPNAGL. However, for flexibility and to make the OPNAGL easily extensible, field OPNAGLEX points to the OPNAEXT.

Created by	Modified by	Used by	Size
Routines that request an open	All routines	IDCIO02	36

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	OPNRFM2	Record format flags.
	1...	*	Reserved—DOS.
	.xxx xxxx	*	Reserved.
1(1)	. 1	OPNOPT3	Option flags.
	1...	*	Reserved—DOS.
	.1..	OPNOPTAC	Open an ICF catalog as a data set.
	..xx xxxx	*	Reserved.
2(2)	. . 1 1	*	Reserved.
4(4)	4	OPNMEM	Address of partitioned data set member name.
8(8)	4	OPNCDT	Address of 44-byte ICF catalog name.
12(C)	24	*	Reserved.

OCARRAY

Open Close Address Array

The open close address array is used to pass the address of the OPNAGL or IOCS for up to four data sets at once from IDCIO01 to IDCIO02. It is used within the I/O adapter.

Created by	Modified by	Used by	Size
IDCIO01	None	IDCIO02	20

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	OCATYP	Type of operation: 1 means open, 2 means close.
1(1)	. 1	OCAOPT	Options:
	1...	OCAOPTCA	Close all open data sets.
2(2)	. . 1	OCANUM	Number of data sets to open.
3(3)	. . . 1	*	Reserved—contains zeros.
4(4)	4	OCADDR*1*	Address of first OPNAGL for open or address of first IOCSTR for close.
8(8)	4	OCADDR*2*	Address of second OPNAGL for open or address of second IOCSTR for close.
12(C)	4	OCADDR*3*	Address of third OPNAGL for open or address of third IOCSTR for close.
16(10)	4	OCADDR*4*	Address of fourth OPNAGL for open or address of fourth IOCSTR for close.

PLST

Processor List

The PLST contains the names of all entries processed during DIAGNOSE.

Created by	Modified by	Used by	Size
IDCDA01 DIAGNOSE	IDCDA02 IDCDA03	IDCDA01 IDCDA02 IDCDA03	512 bytes

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	PLSTID	ID='PLST'.
4(4)	1	*	Reserved.
5(5)	3	PLSTLEN	Length of PLST block.
8(8)	4	PLSTAVAL	Address of next PLST entry.
12(C)	4	PLSTLAST	Address of last PLST entry.
16(10)	4	PLSTNEXT	Address of next PLST block.
20(14)	*var*	PLSTNAMS	Entries of PNTRY.

Each PLSTNAMS entry contains the following:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	PNEPTR	ELST entry pointer.
4(4)	4	PNRAB	RBA record.
8(8)	1	PNTYPE	Record type.
9(9)	1	PNFLAG	Flag bits.
	1...	FNDUMP	Entry was dumped.
	.1..	FNFASSOC	Valid association cell found.
	..1.	FNFVOL	Valid volume cell found.
	...1	FNFEROVR	Error overflow.
	 xxxx	*	Reserved.
10(A)	6	PNERROR	Error code/type.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
16(10)	1	PNFERNUM	Relative error number.
17(11)	1 *x n*	PNFERROR	Up to five error codes.

OPRARG

Positioning Argument List

OPRARG contains the address of the IOCSTR for a data set that is to be positioned or for a partitioned data set whose directory is to be altered. It is used within the I/O adapter.

Created by	Modified by	Used by	Size
IDCIO01	None	IDCIO03	12

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	OPRTYP	Type of operation: 1 indicates positioning; 2 indicates alter a partitioned data set directory; 3 indicates return information (UIOINFO).
1(1)	. 1	OPRPNO	Number of arguments passed to UIOINFO.
2(2)	. . 1	OPROPT	Option byte.
3(3)	. . . 1	*	Reserved—contains zeros.
4(4)	4	OPRICS	Address of input IOCSTR (the data set to be positioned).
8(8)	4	OPROCS	Address of output IOCSTR (the data set to be positioned).

PUAGL

Post UCB Argument List

The PUAGL is passed when a UMSSUNIT macro is issued with a post request. It defines a request to post a specific UCB.

Created by	Modified by	Used by	Size
Calling routine	IDCSA06	IDCSA06	20

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	PUHEAD	Set with 'POSTUCBƀ' by IDCSA06.
8(8)	4	PUUCBPTR	Address of the UCB to be posted.
12(C)	4	PUTTRPTR	Address of the 4-byte TTR to the VTOC to be posted in the UCB.
16(10)	4	PULABELP	Address of the 6-byte volume serial number to be posted in the UCB.

PCARG

Print Control Argument List

The print control argument list is used to build a PCT (print control table). This list is an argument of the UESTS macro or the UESTA macro, used to establish a PCT. The list is in a static text module or in storage.

Created by	Modified by	Used by	Size
Calling routine	None	IDCTP04	33

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	PCMTLP	If PCARG is in a static text module, this is an offset from the beginning of the PCARG to a main title line, fully formatted. If PCARG is in storage, this is the address of a main title line, fully formatted.
4(4)	4	PCSTLP	If PCARG is in a static text module, this is an offset from the beginning of the PCARG to one, two, or three contiguous, fully formatted lines for the subtitle. If PCARG is in storage, this is the address of subtitle lines. The first byte of each line contains the spacing character (0, 1, 2, or 3), and the number of lines is found in PCSTLC.
8(8)	4	PCFLP	If PCARG is in a static text module, this is an offset from the beginning of the PCARG to one, two, or three contiguous, fully formatted footing lines. If PCARG is in storage, this is the address of footing lines. The first byte of each line contains the spacing character (0, 1, 2, or 3), and the number of lines is found in PCFLC.
12(C)	4	PCPCP	If PCARG is in a static text module, this is an offset from the beginning of the PCARG to a 256-byte print chain translate table. If PCARG is in storage, this is the address of a 256-byte print chain translate table.
16(10)	2	PCPNL	Print column number where the page number field begins.
18(12)	..2	PCPTL	Time field location.
20(14)	2	PCPDL	Date field location.
22(16)	..2	PCMTLC	Number of lines at PCMTLP.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
24(18)	2	PCSTLC	Number of lines at PCSTLP.
26(1A)	..2	PCFLC	Number of lines at PCFLP.
28(1C)	2	PCLW	Print line width.
30(1E)	..2	PCPD	Page depth.
32(20)	1	PCDSC	Default space character, used when space character is not given; invalid, or on overflow. Valid values are 1, 2, or 3.

PCT

Print Control Table

The print control table contains the current page specifications for printing: page width and depth, pointers to heading and footing lines, etc. One PCT, called the *primary* PCT, contains the default values established at processor initialization time. An optional PCT, called the *secondary* PCT, contains page specifications that are unique to a particular FSR, and is cleared between commands. Both PCTs have the same format.

Created by	Modified by	Used by	Size
IDCTP04	IDCTP05 IDCTP01	IDCTP01	112

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	PCTIDN	Identification field: the primary PCT contains "PCT1" in this field; the secondary PCT contains "PCT2."
4(4)	4	PCTFLG	Action flags:
	1...	PCTH1F	A new header is being entered. This bit is set by IDCTP05 and reset by IDCTP01 as soon as the first header line is printed.
	.1..	PCTH2F	More than one header line is to be saved. This bit is set when the first line is printed by IDCTP01 and reset when the last line has been printed. The count in PCTHLC controls this bit.
	..1.	PCTHAF	A header has been set up.
	...1	PCTLLM	Last line was a message.
	.... 1...	PCTAPF	Alternative print file flag.
8(8)	4	PCTSPP	Address of secondary PCT. This field is ignored in the secondary PCT.
12(C)	4	PCTIOC	Address of IOCSTR to be used with UPUT macro.
16(10)	2	PCTCPN	Current page number on active data set.
18(12)	. . 2	PCTNLI	Next absolute line number on the current page of active data set.
20(14)	4	PCTIOS	Address of IOCSTR for SYSPRINT.
24(18)	2	PCTSPN	Current page number on standard data set.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
26(1A)	. . 2	PCTSNL	Next absolute line number on the current page of standard data set.
28(1C)	4	PCTIOP	Address of IOCSTR for alternative print data set.
32(20)	2	PCTAPN	Current page number on alternative data set.
34(22)	. . 2	PCTANL	Next absolute line number on the current page of alternative data set.
36(24)	8	PCTSTM	Name of the static text module presently in virtual storage.
44(2C)	4	PCTSME	Entry point for static text module presently in virtual storage.
48(30)	4	PCTSQP	Address of queue of format structures that are retained until the completion of the function or the issuance of a URESET.
52(34)	4	PCTAHP	Address of the last header line that was used, needed on an overflow.
56(38)	4	PCTMLP	Address of main title lines, already fully formatted.
60(3C)	4	PCTSLP	Address of subtitle lines, already fully formatted.
64(40)	4	PCTTRP	Address of translate table.
68(44)	4	PCTPLW	Print line width for the output device.
72(48)	2	PCTMLC	Number of main title lines.
74(4A)	. . 2	PCTSLC	Number of subtitle lines.
76(4C)	4	PCTFLP	Address of footing lines, already fully formatted.
80(50)	2	PCTFLC	Number of footing lines.
82(52)	. . 1	PCTHLC	Number of heading lines.
83(53)	. . . 1	PCTHSC	Total number of lines consumed by the currently active header and the first data line.
84(54)	2	PCTPNL	Page number location in the main title line.
86(56)	. . 2	PCTPMN	Signals that this is a message. Before writing a message it contains -1. During writing a message it contains the message number.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
88(58)	2	PCTAPC	"Floating" print column number, used with blank suppression.
90(5A)	..2	PCTPPD	Total number of lines and spaces that may be printed on one page.
92(5C)	2	PCTDSC	Default space count, used for overflow or in place of an invalid spacing request.
94(5E)	2	PCTPNI	Page number increment, added to PCTCPN at each page eject.
96(60)	2	PCTFDL	Absolute line number for the first data line on each page.
98(62)	..2	PCTLDL	Absolute line number of the last data line.
100(64)	2	PCTFLN	Absolute line number for the first footing line.
102(66)	..2	PCTLNM	Lines in print stack.
104(68)	4	PCTBUF	Buffer address.
108(6C)	4	PCTBNL	Address in buffer for next line.

RLST

Records List

The RLST identifies all records dumped during DIAGNOSE.

Created by	Modified by	Used by	Size
IDCDA02	IDCDA01 IDCDA02 IDCDA03	IDCDA01 IDCDA02 IDCDA03	Variable

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	RLSTID	ID=‘RLST’.
4(4)	1	*	Reserved.
5(5)	3	RLSTLEN	Length of the RLST.
8(8)	4	RLSTAVAL	Address of next RLST entry.
12(C)	4	RLSTLAST	Address of last RLST entry.
16(10)	4	RLSTNEXT	Address of next RLST block.
20(14)	*var.*	RLSTNAMS	Entries of RNTRY.

Each RLSTNAMS entry contains the following:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	2	RNHDR	Size of fixed header.
	1	RNFLAG	Flag bits.
	1...	RNBCS	Record is from BCS.
	.1..	RNVVR	Record is from VVDS.
	..1.	RNVTOC	Record is from VTOC.
	...x xxxx	*	Reserved.
	1	RNMLL	Length of the name.
	var.	RNAME	Name of record dumped.

RACFAGL

RACF URACHECK Argument List

The RACFAGL defines the URACHECK parameter list and is passed by the caller who issues the URACHECK Umacro to tell IDCSA08 what data set to RACHECK.

Created by	Modified by	Used by	Size
Calling routine	None	IDCSA08	20

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	RACFFIG1	Flag byte 1.
	...1	RACFDSTV	Data set type: 0 = Non-VSAM data set; 1 = VSAM data set.
	.... .1..	RACFLOGF	No logging on RACF failures.
	.... ..1.	RACFLOGN	No logging at all.
	1	RACFCSA	Return profile in storage request.
1(1)	1	RACFFIG2	Flag byte 2.
	.1... ...	RACFTALT	Requested alter attribute.
	.... 1...	RACFTCTI	Requested control attribute.
	.... .1..	RACFTUPD	Requested update attribute.
	.... ..1.	RACFTRD	Requested read attribute.
2(2)	1	RACFFIG3	Flag byte 3.
	..1.	RACFPRF	Profile address giver.
3(3)	1	RACFFIG4	Flag byte 4. Reserved—contains zeros.
4(4)	4	RACFENT	Address of data set name.
8(8)	4	RACFPROF	Address of resource profile in storage. Present when RACFPRF is on.
12(C)	4	RACFCIN	Address of CLASS name.
16(10)	4	RACFVOLS	Address of volume serial number.

COMMAREA

Reader/Interpreter Communication Area

The COMMAREA is only used within the reader/interpreter for batched jobs to pass information between reader/interpreter modules.

Created by	Modified by	Used by	Size
IDCRI01	IDCRI01 IDCRI02 IDCRI03	IDCRI01 IDCRI02 IDCRI03	55

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	RECRDPTR	Address of the beginning of the record currently being scanned.
4(4)	4	FDTADDR	Address of the primary pointer vector for the FDT.
8(8)	4	DESCPTR	Address of the command descriptor currently being used.
12(C)	4	WORKPTR	Address of local work area.
16(10)	2	RISTATUS	Internal error code for the reader/interpreter; set to nonzero if an error is discovered.
18(12)	2	SCANINDX	Offset into the current record of the last character that was extracted.
20(14)	2	SCNLIMIT	Location of the final character in the current record that may be scanned.
22(16)	2	LASTCC	Last processor condition code.
24(18)	2	MAXCC	Maximum processor condition code.
26(1A)	8	FSRLNAME	FSR load module name to be invoked if this command is executed.
34(22)	4	POOLID	Storage area identification code for all space used for the FDT.
38(26)	8	VERBNAME	Verb from the current input command.
46(2E)	8	DESCNAME	Module name for the current command descriptor.

Offset	**Bytes and Bit Pattern**	**Field Name**	**Description: Content, Meaning, Use**
54(36)	1	*	Miscellaneous flags:
	1...	GOODCMD	Current command is valid; have executive invoke the FSR.
	.1..	EOFOK	End of input stream may legitimately occur.
	..1.	OPTSFLAG	Current command came from parameter options specified by the invoker of access method services.
	...1	SCANONLY	Current command is being scanned only for syntax errors.
	.... 1...	SKIPPAST	Current command has just been bypassed.

HDAREA

Reader/Interpreter Historical Area

The reader/interpreter historical area is created and initialized on the first call to the reader/interpreter for batched jobs. It contains information that must be saved across commands, such as input source margins and table locations.

Created by	Modified by	Used by	Size
IDCRI01	IDCRI01 IDCRI02 IDCPM01	IDCRI01 IDCRI02	46

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	LEFTMGN	Leftmost column to use in the input statement. Default to column 2.
2(2)	. . 2	RIGHTMGN	Rightmost column to use in the input statement. Default to column 72.
4(4)	4	LOADTPTR	Address of the command name table, IDCRILT.
8(8)	4	KWTBLPTR	Address of modal command verb table, IDCRIKT.
12(C)	4	ADDRIOCS	Address of IOCSTR for input data set.
16(10)	1	NESTLVL	IF-THEN nesting level where current command appears.
17(11)	.2x*n*	MODLFLGS *n*	Modal flags. A set of modal flags is used for each level of IF–THEN nesting. *n* is the number in NESTLVL.

Each set contains the following:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	1	NULLDO	Number of unneeded "DO" commands for which no matching "END" commands have been encountered at the current NESTLVL.

Offset	**Bytes and Bit Pattern**	**Field Name**	**Description: Content, Meaning, Use**
	.1	*	Flags:
	1...	DOFLAG	Current command is part of a "DO" group.
	.1..	THENFLAG	Current commands are associated with a true "IF" condition.
	..1.	ELSEFLAG	Current commands are associated with a false "IF" condition.
	...1	SKIPFLAG	Current commands are to be only checked for proper syntax.

EXWRARG

REPAIRV Argument List

The EXWRARG is passed when UEXCP is issued with a REPAIRV request. It defines a request to repair a 3850 data set.

Created by	Modified by	Used by	Size
Calling routine	IDCIO05	IDCIO05	40

	Bytes and		Description:
Offset	**Bit Pattern**	**Field Name**	**Content, Meaning, Use**
0(0)	8	EXRWRES	Reserved.
8(8)	4	EXRWCTBL	Address of the open I/O control blocks (IOCTLBLK) that were created in the OPEN call.
12(C)	4	EXRWIOAR	Address of an I/O area: for READCNT and READKD, the read I/O area (RIOAREA); for WRITEREC and FWRITE, the write space area (WIOAREA).
16(10)	4	EXRWDARE	Address of the data read (set when the REPAIRV function's routine exits).
10(14)	4	EXLOCPTR	Address of the location table, set when the routine exits.
24(18)	2	EXCCWCNT	The number of count fields read by READCNT when the routine exits, or The number of data fields read by READKD when the routine exits, or The record number of the record to be operated on by SPACCR. This field is not used by WRITEREC or FWRITE.
26(IA)	5	EXRWCHR	Address of the record to be operated on:
26(IA)	2	EXCC	Cylinder address.
28(IC)	2	EXHH	Track address.
30(IE)	1	EXRECNUM	Record number.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
31(IF)	1		Reserved.
32(20)	1	EXRWFUN	For WRITEREC only–the type of write operation: 2—WRITE SPECIAL 3—WRITE KEY DATA
32(21)	3		Reserved.
36(24)	4	EXRWKDLN	For SPACE only—address of an area that contains the key length and data length of the record whose count field is to be bypassed. If EXRWKDLN is zero, SPACCR gets the length information from the count field previously read in by READCNT, and saves the length in the location table entry for the requested record.

RCTAGL

Recatalog Argument List

The RCTAGL is passed when a URECAT macro is issued. It defines a request to recatalog a non-VSAM data set.

Created by	Modified by	Used by	Size
Calling routine	IDCSA07	IDCSA07	76

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	RCTHEAD	Set with 'RCTAGLƀ ƀ' by IDCSA07.
8(8)	2	RCTOPT	Recatalog options:
	1...	RCTDEV	Indicates that the device type is to be changed. RCTNDEV contains the new device type.
	.1...	RCTVOL	Indicates that the volume serial number is to be changed. RCTNVOL contains the new volume serial number.
10(A)	..4	RCTODEV	Contains the current device type for the data set.
14(E)	..4	RCTNDEV	Contains the new device type for the data set. This field is ignored if RCTDEV is not set.
18(12)	..6	RCTOVOL	Contains the current volume serial number for the data set.
24(18)	6	RCTNVOL	Contains the new volume serial number for the data set. This field is ignored if RCTVOL is not set.
30(1E)	..44	RCTDSET	Contains the data set name of the data set to be recataloged.

AUTOTBL

Storage Table

The storage table contains the address for storage areas for modules IDCIO01, IDCIO05, IDCSA02, IDCSA06, IDCSA07, IDCSA08, IDCSA09, IDCSA10, and IDCTP01. The modules almost always use the same storage instead of getting and freeing storage each time the module gets control.

Created by	Modified by	Used by	Size
IDCSA01	IDCSA02 IDCSA03	IDCSA02 IDCSA03	108

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	108	AUTOARAY	There are nine sets of the following four fields. One set for each module IDCIO01, IDCIO05, IDCSA02, IDCSA06, IDCSA07, IDCSA08, IDCSA09, IDCSA10, and IDCTP01.

Each set contains the following:

	4	AREAID	Four-byte CSECT identification for the module using this set of 12 bytes. The field contains ‘IO01’, ‘IO05’, ‘SA02’, ‘SA06’ ‘SA07’, ‘SA08’, ‘SA09’, ‘SA10’, or ‘TP01’
	2	STATUS	Indicator of the number of storage areas being used for this module. The field contains: **0** No storage being used. **1** Only the storage area addressed in PTR1 is being used. **>1** The number is a count of storage areas in use for this module. There are more storage areas than are addressed in PTR1.
	..2	ASIZE	Number of bytes the module uses.
	4	PTR1	Address of the first storage area for this module.

SELAGL

Selecting a DDname Argument List

The SELAGL is passed when a UMSSUNIT macro is issued with a SELECTDD request. It defines a request to select a DD statement and UCB that can be used for the volume if the volume is already mounted.

Created by	Modified by	Used by	Size
Calling routine	IDCSA06	IDCSA06	23

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	SELHEAD	Set by IDCSA06 with 'SELECTDD'.
8(8)	4	SELUCBP	Address of the 4-byte area, provided by the caller, in which IDCSA06 returns the address of the UCB associated with the specified volume.
12(C)	4	SELDDNP	Address of an 8-byte area provided by the caller in which IDCSA06 will return the name of the DD statement that was used in mounting the volume.
16(10)	6	SELVOL	Volume serial number of the volume used in the DD selection search. This volume serial number is passed by the caller.
22(16)	1	SELFLAGS	Option flags

SSCTARGL

Subsystem Control Argument List

The SSCTARGL is passed to IDCSS04 whenever a SETCACHE or BINDDATA command is issued.

Created by	Modified by	Used by	Size
IDCBD01 IDCSC01	IDCSS04	IDCSS04	26

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	SSCTHEAD	Set with SSCTARGL by IDCSS04.
8(8)	4	SSCTADDN	Address of the ddname, or zero if no ddname is specified.
12(C)	4	SSCTAVOL	Address of the volume serial number, or zero if no volume is specified.
16(10)	1	SSCTOPTN	Flags:
	1...	SSCTBSET	Indicates that bind or unbind is requested.
	.1..	SSCTVSET	Indicates that caching is to be enabled or disabled.
	..1.	SSCTON	Establish caching or high performance area.
	...1	SSCTOF	Terminate high performance area or caching.
	.... 1...	SSCTCCHH	Bind or unbind using cchh limits.
	.... .1..	SSCTDEV	Terminate high performance area for device.
	.... ..1.	SSCTSSYS	Subsystem option.
	x	*	Reserved.
17(11)	4	SSCTLCCHH	Lower storage cchh limit.
21(15)	4	SSCTHCCHH	Higher storage cchh limit.

SSGARGL

Subsystem Get Argument List

The SSGARGL is passed whenever a LISTDATA command is issued.

Created by	Modified by	Used by	Size
IDCLA01	IDCSS01	IDCSS01	32

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	SSGHEAD	Set with SSGARGL by IDCSS01.
8(8)	4	SSGADDN	Address of the ddname, or zero if no ddname is specified.
12(C)	4	SSGAVOL	Address of the volume serial number, or zero if no volume is specified.
16(10)	4	SSGUNIT	Unit type.
20(14)	4	SSGOLN	Length of the buffer storage.
24(18)	4	SSGOADR	Address of the buffer area.
28(1C)	1	SSGOPT	Options byte.
	1...	SSGRPD	Request to sense subsystem counts.
	.1..	SSGRSS	Request to sense subsystem status.
	..xx	*	Reserved.
	.... 1...	SSGALL	Read for all subsystems.
	.... .1..	SSG1SS	Read for specified subsystem.
	.... ..1.	SSGDEV	Read for specified device.
	x	*	Reserved.
29(1D)	3	*	Reserved.

SSWKAREA

Subsystem Work Area

The subsystem work area is used to build channel command words and is passed to IDCSS02 to find UCBs associated with the request.

Created by	Modified by	Used by	Size
IDCSS01 IDCSS04	IDCSS02	IDCSS02	476

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	SSWKCCW	The channel command word is built in this field.
8(8)	6	SSWKVOLA	Volume serial number argument for IDCSS02.
14(E)	1	SSWKOPTN	Option byte.
	1...	SSWKALL	All like subsystem requests.
	.1..	SSWKSYSS	All UCBs for specified subsystem.
	..1.	SSWKDEV	Only UCBs for this device.
	 xxxx	*	Reserved.
15(F)	1	*	Reserved.
16(10)	4	SSWKDEVT	Device type in subsystem.
20(14)	4	SSWKNEXT	Pointer to next volume list if one exists.
24(18)	448	SSWKDATA	Volume, UCB, and unit address list.
472(1D8)	4	SSWKFFFF	End of list delimiter.

Information about each volume entry:

0(0)	6	SSWKVOL	Volume serial number for the device.
6(6)	2	*	Reserved.
8(8)	4	SSWKUCB1	First UCB entry.
12(C)	4	SSWKUCB2	Second UCB entry.

SAHIST

System Adapter Historical Area

The system adapter's historical area is pointed to by the field GDTSAH. It contains information that is shared between system adapter modules.

Created by	Modified by	Used by	Size
IDCSA01	IDCSA02 IDCSA03 IDCSA10	IDCIO05 IDCSA02 IDCSA03 IDCSA06 IDCSA10	20

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	GPFIRST	First UGPOOL storage area pointer.
4(4)	4	GPLAST	Last UGPOOL storage area pointer.
8(8)	4	AUTOPTR	Address of AUTOTBL.
12(C)	4	SAHSTA	Address of ESTAE argument list (STAEPARM).
16(10)	4	LLBLKPTR	pointer to first load list block.

TEST Option Data Area

The TEST option data area is used to gather debugging information requested by a PARM command with TRACE, AREAS, or FULL options. The TEST options data area is three tables. The first table, TESTDATA, is present if any PARM command with TRACE, AREAS, or FULL has been executed. The address of TESTDATA is in GDTDBH.

The second table, AREADATA, exists if a PARM command with an AREAS option has been executed. If AREADATA exists, it immediately follows TESTDATA.

The third table, FULLDATA, exists if a PARM command with a FULL option has been executed. If FULLDATA exists, it immediately follows AREADATA, or if AREADATA does not exist, FULLDATA immediately follows TESTDATA.

Created by	Modified by	Used by	Size
IDCPM01	IDCPM01 IDCDB01	IDCPM01 IDCDB01	Variable

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
TESTAREA:			
0(0)	4	AREAPTR	Address of area's identifier table, AREADATA. Zero indicates the table does not exist.
4(4)	4	FULLPTR	Address of full dump table FULLDATA. Zero indicates the table does not exist.
8(8)	2..	SNAPID	Number of last full region dump.
10(A)	..2	TESTRACE	A nonzero value means print the trace tables each time a UDUMP macro is issued. A zero value means print the trace tables only for modules specified in AREAS and FULL options.
AREADATA:			
0(0)	4	AREAINDX	Number, *j*, of entries in area's identification array. One entry exists for each area identifier specified in the PARM command.
4(4)	2×*j*	AREADUMP*j*	Area's identifier array containing *j* entries.

Each entry contains the following:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	2	AREANAME	Two-character module identifier where information is gathered. If there is an odd number of area names, 2 bytes are added to the end of the array.

FULLDATA:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	FULLINDX	Number, *k*, of entries in full region dump array. One entry exists for each full dump.
4(4)	2×*k*	FULLDUMP*k*	Full region dump array containing *k* entries.

Each entry contains the following:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	4	FDUMPID	4-character module identifier where dump is taken.
	2..	FDUMPBEG	Number of the pass through the dump point when dumping is to begin—between 1 and 32767.
	..2	FDUMPCNT	Number of dumps to take—between 1 and 32767.
	2..	REALBEG	Current number of passes through this dump point.
	..2	REALCNT	Number of dumps already taken at this dump point.

Text Structures

Text structures are load modules that contain text (messages and static text items) and format information to use while preparing printed output. This information can be default page dimensions or layout, message text, headings for listings, and similar directions that are used by the text processor. There are 24 text structure modules, as named in the following table along with the function associated with each. Some FSRs use text structures from other FSRs.

IDCTSAL0	ALTER
IDCTSBI0	BUILDINDEX
IDCTSBD0	BINDDATA
IDCTSCC0	CNVTCAT
IDCTSCK0	CHKLIST
IDCTSDA0	DIAGNOSE
IDCTSDE0	DEFINE
IDCTSDL0	DELETE
IDCTSEX0	Executive
IDCTSIO0	I/O Adapter
IDCTSLA0	LISTDATA
IDCTSLC0	LISTCAT
IDCTSLC1	LISTCAT(messages)
IDCTSLR0	LISTCRA
IDCTSLR1	LISTCRA(messages)
IDCTSMP0	IMPORT/IMPORTRA
IDCTSPR0	PRINT/REPRO
IDCTSRC0	EXPORTRA
IDCTSRI0	Reader/Interpreter for batched jobs
IDCTSRI1	Reader/Interpreter for interactive jobs
IDCTSRS0	RESETCAT
IDCTSSC0	SETCACHE
IDCTSTP0	Text Processor(print chains)
IDCTSTP1	Text Processor(messages)
IDCTSTP6	Text Processor(UERROR messages)
IDCTSUV0	Universal(any module)
IDCTSXP0	EXPORT

A text structure consists of an index and text entries. The index is simply a list of halfword displacements from the beginning of the text structure to the beginning of the text entry being indexed. The text structure identification number is used as the index number. A halfword count of the number of entries precedes the index.

Note: An index entry of -1 indicates that the corresponding text entry is nonexistent.

All text entries contain heading fields and one of the following:

- A format list as described under FMTLIST immediately followed by any static text such as messages referenced by the format list.
- A print control argument list as described under PCARG immediately followed by any static text such as title lines and translate tables referenced by the print control argument list.
- Character code tables which support the GRAPHICS parameter of the PARM command.

Created by	Modified by	Used by	Size
IBM-Supplied	None	DCTP01 IDCTP05	Variable

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	INDEX	Number, *n*, of entries in this index.
2(2)	2×*n*	INDEX*n*	Offset to the appropriate text entry.

Text Entry

The following description shows only the header fields of each text entry. For the remainder of the description, see FMTLIST or PCARG. The text entry begins at offset 2 x *n* + 2 from the beginning of the Text Structure module.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	TXT	Length in bytes of the text entry that follows (not including these header fields).
2(2)	1	FLG	Flag byte:
	1...		Message entry.
	.1..		Header entry.
	..1.		Secondary message entry.
	...1		Do not transmit this entry to a TSO terminal.
	.... 1...		Secondary message entry.
3(3)	1	*	Reserved.

The following two fields only exist if this is a text entry for a header line:

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
4(4)	2	HDLI	The number of printable header lines.
6(6)	2	HDSP	The number of page lines occupied by header lines, intervening blank lines, and the first line of printed data.

UCRYPT Parameter List—CRYPTAGL

The CRYPTAGL defines a request to encipher or decipher data. The address of the CRYPTAGL ispassed as a parameter to the I/O adapter from any routine that requires the encipher/decipher function.

Created by	Modified by	Used by	Size
IDCRP01	IDCIO01	IDCIO01	148

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	CRPLEN	Length of CRYPTAGL.
2(2)	2	*	Reserved.
4(4)	4	CRPIOCSI	Input IOCS pointer.
8(8)	4	CRPIOCSO	Output IOCS pointer.
12(C)	8	CRPDK	Data Key.
20(14)	8	CRPEMDK	Data key enciphered under master key.
28(IC)	8	CRPEDK	Data key enciphered under file key.
36(24)	8	CRPFK	File keyname.
44(2C)	8	CRPSTEK	Keyname to be stored in header.
52(34)	8	CRPSEED	Seed.
60(3C)	8	CRPPRN	Pseudo random number.
68(44)	4	CRPWA	Work area pointer.
72(48)	4	CRPBF	Cipher buffer pointer.
76(4C)	4	CRPCRRN	Current RRN.
80(50)	4	CRPTRLN	Total record length.
84(54)	4	CRPCBFP	Current cipher buffer position.
88(58)	4	CRPLA	Cipher length array pointer.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
92(5C)	4	CRPRNA	Cipher relative number array pointer.
96(60)	4	CRPPLID	Pool id for UGPOOLs.
100(64)	4	CRPUSD	User data pointer.
104(68)	4	CRPSHPL	SHIP list pointer.
108(6C)	4	CRPCLRD	Clear data pointer.
112(70)	4	CRPENCD	Enciphered data pointer.
116(74)	4	CRPCHN	Chaining value pointer.
120(78)	4	CRPDATL	Length of data to be enciphered/deciphered.
124(7C)	2	CRPIOER	I/O error counter.
126(7E)	1	CRPFLG1	Flags.
	1...	CRPENCPH	Encipher.
	0...		Decipher.
	.1..	CRPSYSK	System key management.
	.0..		Private key management.
	..1.	CRPHDR	Read/write header.
	..0.		Encipher/decipher data.
	...1	CRPINT	INTERNAL system file key.
	...0		EXTERNAL system file key.
	.... 1...	CRPWRBF	Write buffer.
	.... .1..	CRPESHP	Encipher under ship key.
127(7F)	1	CRPFLG2	
	1...	CRPRRDS	Enciphered RRDS.
	.1..	CRPSTFKN	Store file keyname in header.
	..1.	CRPSTEDK	Store data key enciphered under file key in header.
	...1	CRPSPDK	User-supplied data key.
	...0		Generate data key.
	.... 1...	CRPSPKN	User-supplied file keyname.
	.... 0...		Obtain file keyname from header.
	.... .1..	CRPSPEDK	User-supplied data key enciphered under file key.
	.... .0..		Obtain data key enciphered under file key from header.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
128(80)	1	CPRFLG3	Flags.
	1...	CRPMACGK	GENKEY request.
	.1..	CRPMACRK	RETKEY request.
	..1.	CRPMACEM	EMK request.
	...1	CRPMACCP	CIPHER request.
129(81)	1	CRPMACRC	Return code from request.
130(82)	1	CRPCBN	Cipher block number.
131(83)	1	CRPCRN	Current record number.
132(84)	16	*	Reserved.

UGPOOL Area

When the UGPOOL Umacro is used, an area of storage is allocated to the user and this UGPOOL area is linked into a chain with other areas allocated by UGPOOL. Each such area is preceded by the first 16 bytes, as shown here.

For a page boundary request, the user's allocated area is not preceded by a UGPOOL area. A 24-byte UGPOOL area is linked into the chain and it points to the user's page boundary area.

Created by	Modified by	Used by	Size
IDCSA02	None	IDCSA02	24

	Bytes and		Description:
Offset	Bit Pattern	Field Name	Content, Meaning, Use
0(0)	4	GPFORWRD	Address of next UGPOOL area.
4(4)	4	GPBACK	Address of last UGPOOL area.
8(8)	4	GPLEN	Number of bytes requested plus 16. (For a page boundary request GPLEN = 24.)
12(C)	4	GPID	Area identification code.
16(10)	4	GPADRPG	Address of area on a page boundary.
20(14)	4	GPLENPG	Length of area on a page boundary.

UGSPACE Area

When the UGSPACE Umacro is used, an area of storage is allocated for the user of the Umacro. Each such area is preceded by 8 bytes of control information, as shown here.

Created by	Modified by	Used by	Size
IDCSA02	None	IDCSA02	8

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4	GSLEN	Number of bytes requested plus 8.
4(4)	4	GSID	'b̸ b̸ b̸' in first 3 bytes for UGSPACE area. The last byte is the subpool-ID.

UIOINFO Option Byte and Return Area

The UIOINFO option byte tells IDCIO02 the information desired by the caller who issues a UIOINFO macro.

Created by	Modified by	Used by	Size
Calling routine	None	IDCIO02	1

UIOINFO Option Byte Description

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	1	IOINFOPT	Option byte:
	1...	IOINFDVT	Return 8-byte device type.
	.1..	IOINFVOL	Return up to five 6-byte volume serial numbers.
	..1.	IOINFDSN	Return 44-byte data set name.
	...1	IOINFSUP	Suppress error message.
	.... 1...	IOINFTMS	Return time stamp from the Format 4 DSCB.
	.... .1..	IOINFLUB	Not used in MVS/XA.

UIOINFO Return Area Description

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	4		Header.

Bytes:

0-1 Length of entire area(including header).

2-3 Length of all data returned(including header).

Data returned for each type of information requested is placed consecutively in the work area. The format for the different types of information is shown below:

	Bytes and		Description:
Offset	Bit Pattern	Field Name	Content, Meaning, Use
	48		Data set name. Bytes: 0-1 Identifier—X'0001'. 2-3 Length of data returned. 4-47 Data set name.
	n		Volume serial number list (variable). Bytes: 0-1 Identifier—X'0002' 2-3 Length of data returned. 4-9 First volume serial number. . . . . . . n+1–n+6 Last volume serial number.
	12		Device type. Bytes: 0-1 Identifier—X'0003'. 2-3 Length of data returned. 4-7 Device type code. 8-11 Maximum block size for device.

20 Time stamp.

Bytes:

0-1	Identifier—X'0004'.
2-3	Length of data returned.
4-11	New time stamp.
12-19	Old time stamp.

UCTAGL

UNCATALOG Argument List

The UCTAGL is passed when a UUNCATLG macro is issued. It defines a request to uncatalog and scratch a non-VSAM data set.

Created by	Modified by	Used by	Size
Calling routine	IDCSA07	IDCSA07	69

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	UCTHEAD	Initialized with 'UCTAGL' by UUNCATLG.
8(8)	44	UCTDSN	Data set name of non-VSAM data set to be uncataloged and scratched.
52(34)	4	UCTACBP	Pointer to catalog ACB.
56(38)	4	UCTVOLP	Pointer to list of volume serial numbers.
60(3C)	8	UCTDD	ddname of volume on which data set resides.
68(44)	1	UCTFLAGS	UUNCATLG option flags
	1...	UCTVSCAT	Data set can be uncataloged with a VSAM catalog management request. (This bit is always set.)

Unit Table

The unit table is passed by a module that issues a USYSINFO Umacro. IDCSA08 returns in it the information for a unit from the system control blocks.

Created by	Modified by	Used by	Size
Calling routine	IDCSA08	Calling routine	11

	Bytes and		Description:
Offset	Bit Pattern	Field Name	Content, Meaning, Use
0(0)	4	UNITUCB	Address of the UCB.
4(4)	2	UNITADR	Channel and unit address:
	1	UNITCHA	Channel address.
	1	UNITUA	Unit address.
6(6)	1	UNITSTAT	Status of the unit:
	1...	UNITISHR	Sharable within the system.
	.1..	UNITXSHR	Sharable across systems.
	..1.	UNITUSHR	Sharable in update mode.
7(7)	4	UNITVTOC	Address of the VTOC—TTR0.

UREST Arguments

Any combination of the following structures can be passed to UREST as arguments. The UREST macro changes default items in the print control table. The structures determine which items UREST will change.

Created by	Modified by	Used by	Size
All routines	None	IDCTP01	Variable

PCRST—Change Subtitle Lines

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	PCRSST	Structure identifier; contains 'ST'.
2(2)	..2	PCRSTLC	Number of subtitle lines provided. The maximum is three.
4(4)	4	PCRSTLP	Address of from one to three contiguous, fully formatted subtitle lines. The number of bytes in each line is the line width plus one for the spacing character. The spacing character is first in each line and must be 1, 2, or 3.

PCRLWS—Change Line Width

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	PCRLWT	Structure identifier; contains 'LW'.
2(2)	..2	PCRLW	New line width in decimal.

PCRPDS—Change Page Depth

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	PCRPDT	Structure identifier; contains 'PD'.
2(2)	..2	PCRPD	New page depth in decimal.

PCRFTS—Change Footing Lines

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	PCRFT	Structure identifier; contains 'FT'.
2(2)	..2	PCRFLC	Number of footing lines provided. The maximum is three.
4(4)	4	PCRFLP	Address of from one to three contiguous, fully formatted footing lines. The number of bytes in each line is the line width plus one for the spacing character. The spacing character is first in each line and must be 0, 1, 2, or 3.

PCRDSCS—Change Default Spacing Character

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	PCRDSCT	Structure identifier; contains 'SC'.
2(2)	..1	PCRDSC	New default space character. Must be the character 1, 2, or 3.

PCRPCS—Change Translate Table

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	PCRPCT	Structure identifier; contains 'PC'.
2(2)	..2	PCRPCC	If the request is for a print chain provided by access method services, this field contains the characters for the print chain identification as in the GRAPHICS parameter of the PARM command. Otherwise, it contains zero.
4(4)	4	PCRPCP	Address of a load module name. The load module consists solely of a 256-byte translate table. If the request is for a standard print chain, this field contains zero.

PCRINP—Change Initial Page Number

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	PCRPNT	Structure identifier; contains 'PN'.
2(2)	..2	*	Reserved.
4(4)	4	PCRPNP	Address of page number field. The first 2 bytes of the page number field contain the number (from 1 to 4 in binary) of following bytes that contain the page number. The page number is 1 to 4 bytes in EBCDIC.

USCRATCH Volume List

The USCRATCH volume list is passed by the caller who issues a USCRATCH Umacro to tell IDCSA08 what data sets to delete.

Created by	Modified by	Used by	Size
Calling routine	None	IDCSA08	Variable

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	2	UVOLCNT	The number of arrays that follow (each array identifies a volume on which the data set resides).
Each entry contains:			
	10×*n*	UVOLENT*n*	Volume identifier. The following fields are repeated n times, where n = UVOLCNT.
	4	UVOLDEV	Device type.
	6	UVOLVOL	Volume serial number.

VS1AGL

Volume VTOC Service Argument List

The VS1AGL is passed whenever module IDCVS01 is called. It contains information needed for one of the following functions: Security Checking, Scratching the VTOC, Retrieving VTOC Fields, Updating VTOC Fields, and Recataloging a non-VSAM Data Set. The parameter list is described separately for each function.

Created by	Modified by	Used by	Size
Calling routine	IDCVS01	IDCVS01	74

Security Checking

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	VS1HEAD	Set with 'SECCHECK' by IDCVS01.
8(8)	4	VSUCBP	If a RESERVE is requested, this field must contain the address of a 4-byte area containing either zeros or the UCB address. The VSFILEP field must contain the address of the DD statement that IDCVS01 uses to obtain the UCB address. IDCVS01 fills in the field if the request is successful. If a RESERVE is not requested this field is ignored.
12(C)	4	VSVOLP	Address of a 6-byte volume serial number of the volume to be processed. The serial number must be the one known to the MSC.
16(10)	4	VSFILEP	Address of the 8-byte name of the DD statement to be used to open the VTOC.
20(14)	4	VSIOP	Upon return, this field contains the address of the IOCSTR for the VTOC data set.
24(18)	4	VSNUMATP	Ignored.
28(1C)	4	VSNXTATP	Ignored.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
32(20)	4	VSCATP	Address of a 44-byte field containing the name of the VSAM catalog that owns the volume. If VSACBP is specified, this field is ignored. If the catalog name is unknown and VSMASTER is set, this field contains zeros.
36(24)	4	VSACBP	Address of the catalog ACB. If this field is specified the VSCATDDP field is ignored. If the ACB is unknown and VSMASTER is set this field must contain zeros.
40(28)	4	VSCATDDP	Address of the 8-byte name of the DD statement used to open the VSAM catalog. If VSOPCAT is specified, this field contains the address of a catalog DD field. That field contains the catalog DD name or blanks if the catalog DD name is unknown.
44(2C)	4	VSPSWDP	Address of the 8-byte VSAM catalog password. If VSMASTER is set and the password is not supplied this field is set to zero.
48(30)	4	VSPFILEP	Address of the list of file names used when prompting for passwords. If VSREAD or VSWRITE are set and no file names are specified this field contains zeros.
52(34)	4	VSSERP	Ignored.
56(38)	4	VSDEVP	Ignored.
60(3C)	4	VSSECOPT	Security checking options:
	1...	VSNOVSAM	Indicates that the volume cannot be owned by a VSAM volume.
	.1..	VSMASTER	Indicates that the master password of the owning VSAM catalog will be verified, and if it fails, all VSAM data set master passwords will be verified.
	..1.	VSNONONV	Indicates that the volume must not contain any non-VSAM data sets.
	...1	VSREAD	Indicates that the read password of each non-VSAM password protected data set will be verified.
	.... 1...	VSWRITE	Indicates that the write password of each non-VSAM password protected data set will be verified.
	.... .1..	VSUCTEST	Indicates that the VSUCAT flag be set upon return indicating whether the volume contains a VSAM user catalog. This flag is interrogated only if VSMASTER is set.
	.... ..1.	VSOPCAT	Indicates that the VSAM catalog will be opened. If this flag is set, VSCATDDP must contain the address of a field containing either blanks or the name of the catalog DD. If IDCVS01 opened the catalog, the catalog DD name is returned in the field set to blanks.
	1	VSUCMAST	Indicates that the master password of the owning VSAM user catalog must be verified.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
64(40)	4	VSODEVP	Ignored.
68(44)	1	VSSECOPT	Ignored.
69(45)	.1	VSVTOPT	Ignored.
70(46)	..1	VSCATOPT	Ignored.
71(47)	...1	VSRETURN	This field is set upon return only if security checking was successful.
	1...	VSVSAM	This bit is set to 1 if the volume is owned by a VSAM catalog. It is set to 0 if the volume is not owned by a VSAM catalog.
	.1..	VSUCAT	If VSUCTEST is set, this bit is set to 0 if the VSAM user catalog is not on the volume. It is set to 1 if it is on the volume.
72(48)	1	VSMSG	Message options:
	1...	VSFROMV	Indicates that the words "FROM VOLUME" are put into any error message indicating the FROM volume is in error.
	.1..	VSTOV	Indicates that the words "TO VOLUME" are put into any error message indicating the TO volume is in error.
73(49)	.1	VSRESOPT	Reserve options:
	1...	VSRES	Indicates that the volume is to be reserved prior to reading or updating the VTOC.
	.1..	VSREAL	Indicates that the RESERVE be done on a real volume. If this bit is off, the volume is virtual.

Scratching the VTOC

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	VSHEAD	Set with 'SCRVTOCƀ' by IDCVS01.
8(8)	4	VSUCBP	Same as Security Checking.
12(C)	4	VSVOLP	Same as Security Checking.
16(10)	4	VSFILEP	Same as Security Checking.

Offset	**Bytes and Bit Pattern**	**Field Name**	**Description: Content, Meaning, Use**
20(14)	4	VSIOP	Address of the IOCSTR for the VTOC data set. This field must be specified if the VTOC is already open. If the VTOC is not open it contains zeros. Upon return it contains zeros if it is not opened and the IOCSTR address if it is opened.
24(18)	4	VSNUMATP	Ignored.
28(1C)	4	VSNXTATP	Ignored.
32(20)	4	VSCATP	Ignored.
36(24)	4	VSACBP	Ignored.
40(28)	4	VSCATDDP	Ignored.
44(2C)	4	VSPSWDP	Ignored.
48(30)	4	VSPFILEP	Ignored.
52(34)	4	VSSERP	Ignored.
56(38)	4	VSDEVP	This field must contain the address of a 4-byte area containing the device type of the volume whose VTOC is to be scratched.
60(3C)	4	VSOSERP	Ignored.
64(40)	4	VSODEVP	Ignored.
68(44)	1	VSSECOPT	Ignored.
69(45)	.1	VSVTOPT	Ignored.
70(46)	..1	VSCATOPT	Ignored.
71(47)	...1	VSRETURN	Ignored.
72(48)	1	VSMSG	Same as Security Checking.
73(49)	.1	VSRESOPT	Same as Security Checking.

Retrieving VTOC Fields

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	VSHEAD	Set with 'GETVTOCb̸' by IDCVS01.
8(8)	4	VSUCBP	Same as Security Checking.
12(C)	4	VSVOLP	Same as Security Checking.
16(10)	4	VSFILEP	Same as Security Checking.
20(14)	4	VSIOP	Same as Scratching the VTOC.
24(18)	4	VSNUMATP	If VSALTTRK is set, this field contains the address of a 2-byte fixed field, where the number of alternate tracks is returned by IDCVS01.
28(1C)	4	VSNXTATP	If VSALTTRK is set, this field contains the address of a 4-byte fixed field, where the CCHH of the next alternate track is returned by IDCVS01.
32(20)	4	VSCATP	Ignored.
36(24)	4	VSACBP	Ignored.
40(28)	4	VSCATDDP	Ignored.
44(2C)	4	VSPSWDP	Ignored.
48(30)	4	VSPFILEP	Ignored.
52(34)	4	VSSERP	Ignored.
56(38)	4	VSDEVP	Ignored.
60(3C)	4	VSOSERP	Ignored.
64(40)	4	VSODEVP	Ignored.
68(44)	1	VSSECOPT	Ignored.
69(45)	.1	VSVTOPT	VTOC field options:
	1...	VSTIME	Ignored.
	.1..	VSALTTRK	Indicates that the number of alternate tracks and the address of the next alternate track should be retrieved. VSNUMATP and VSNXTATP must contain the address of the return area.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	..1.	VSVSFLAG	Indicates that the VSAM ownership be returned. The VSVSAM bit in field VSRETURN is set by IDCVS01.
70(46)	..1	VSCATOPT	Ignored.
71(47)	...1	VSRETURN	This field is modified if the function was successful and VSVSFLAG was set:
	1...	VSVSAM	This bit is set to 1 if the volume is owned by a VSAM catalog. It is set to 0 if the volume is not owned by a VSAM catalog.
	.1..	VSUCAT	Ignored.
72(48)	1	VSMSG	Same as Security Checking.
73(49)	.1	VSRESOPT	Same as Security Checking.

Updating VTOC Fields

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	VSHEAD	Set with 'PUTVTOCƀ' by IDCVS01.
8(8)	4	VSUCBP	Same as Security Checking.
12(C)	4	VSVOLP	Same as Scratching the VTOC.
16(10)	4	VSFILEP	Same as Security Checking.
20(14)	4	VSIOP	Same as Scratching the VTOC.
24(18)	4	VSNUMATP	If VSALTTRK is set, this field contains the address of a 2-byte fixed field which contains the CCHH of the next alternate track.
28(1C)	4	VSNXTATP	If VSALTTRK is set, this field must contain the address of a 4-byte fixed field which contains the CCHH of the next alternate track.
32(20)	4	VSCATP	Ignored.
36(24)	4	VSACBP	Ignored.
40(28)	4	VSCATDDP	Ignored.
44(2C)	4	VSPSWDP	Ignored.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
48(30)	4	VSPFILEP	Ignored.
52(34)	4	VSSERP	Ignored.
56(38)	4	VSDEVP	Ignored.
60(3C)	4	VSOSERP	Ignored.
64(40)	4	VSODEVP	Ignored.
68(44)	1	VSSECOPT	Ignored.
69(45)	.1	VSVTOPT	VTOC field options:
	1...	VSTIME	Indicates that the VSAM time stamp is updated with the current time of day.
	.1..	VSALTTRK	Indicates that the number of alternate tracks and the address of the next alternate track should be updated. VSNUMATP and VSNXATP contain the address of the information.
	..1.	VSVSFLAG	
70(46)	..1	VSCATOPT	Ignored.
71(47)	...1	VSRETURN	Ignored.
72(48)	1	VSMSG	Same as Security Checking.
73(49)	.1	VSRESOPT	Same as Security Checking.

Recataloging a Non-VSAM Data Set

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	VSHEAD	Set with 'RECATLGb̸' by IDCVS01.
8(8)	4	VSUCBP	Same as Security Checking.
12(C)	4	VSVOLP	Same as Security Checking
16(10)	4	VSFILEP	Same as Security Checking.
20(14)	4	VSIOP	Same as Scratching the VTOC.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
24(18)	4	VSNUMATP	Ignored.
28(1C)	4	VSNXTATP	Ignored.
32(20)	4	VSCATP	Ignored
36(24)	4	VSACBP	Ignored.
40(28)	4	VSCATDDP	Ignored.
44(2C)	4	VSPSWDP	Ignored.
48(30)	4	VSPFILEP	Ignored.
52(34)	4	VSSERP	If the VSSERIAL bit is set, this field must contain the address of a 6-byte 11 character field that contains the new volume serial number.
56(38)	4	VSDEVP	If VSDEVICE is set, this field must contain the address of the new 4-byte device code obtained from the UCBTYP field.
60(3C)	4	VSOSERP	Contains the address of the current 6-byte volume serial number.
64(40)	4	VSODEVP	Contains the address of the current 4-byte device code obtained from the UCBTYP field.
68(44)	1	VSSECOPT	Ignored.
69(45)	.1	VSVTOPT	Ignored.
70(46)	..1	VSCATOPT	Recatalog options:
	1...	VSDEVICE	Indicates that the device type is changed. If this bit is set, VSDEVP must contain the address of the new device type.
	.1..	VSSERIAL	Indicates that the serial number is changed. If this bit is set, VSSERP must contain the address of the new serial number.
	..1.	VSLIST	Indicates that the names of all data sets successfully recataloged be listed.
71(47)	...1	VSRETURN	Ignored.
72(48)	1	VSMSG	Same as Security Checking.
73(49)	.1	VSRESOPT	Same as Security Checking.

VS2AGL

Volume Label Service Argument List

The VS2AGL argument list is passed whenever IDCVS02 is called. It contains information needed for one of the following functions:

- The INITVOL function defines a request to have a mass storage volume initialized for use by the operating system.
- The GETLABEL function defines a request to retrieve one or more of the following fields from the volume label: owner or volume serial number.
- The PUTLABEL function defines a request to update one of the following fields in the volume label: owner or volume serial number. The parameter list is described separately for each function.

Created by	Modified by	Used by	Size
Calling routine	IDCVS02	IDCVS02	45

Initializing Mass Storage Volumes

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	VS2HEAD	Set with 'INITVOLb̸' by IDCVS02.
8(8)	4	VSUCBPTR	Address of the UCB.
12(C)	4	VSVTOCSZ	Number of tracks for the VTOC.
16(10)	4	VSOWNPTR	Address of the owner name to be placed in the volume label.
20(14)	4	VSVOLPTR	Address of the volume serial number to be placed in the volume label.
24(18)	4	VSIOBKPT	Address of the area in which the I/O control block address is returned. If I/O control block address is nonzero when control is returned, the UEXCP (CLOSE) macro must be issued to close the DCB. The I/O control block address must be initialized to zero before a request.
28(1C)	1	VSLABOPT	Ignored.
29(1D)	.6	VSVOLUME	Ignored.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
35(23)	...1	VS2MSG	Ignored.
36(24)	1	VS2RESOP	Ignored.
37(25)	.8	VSDDNAME	DD name for the volume.

Retrieving Volume Label Fields

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	VS2HEAD	Set with 'GETLABEL' by IDCVS02.
8(8)	4	VSUCBPTR	Address of a 4-byte area containing the UCB address, or zeros if the UCB address is unknown. If the UCB address is zero, the UCB address will be put in the 4-byte area upon return.
12(C)	4	VSVTOCSZ	Ignored.
16(10)	4	VSOWNPTR	Address of the area for the owner name to be returned if VSOWNER is set.
20(14)	4	VSVOLPTR	Address of an area for the volume serial number to be returned if VSVOLSER is set.
24(18)	4	VSIOBKPT	Ignored.
28(1C)	1	VSLABOPT	Label options:
	1..	VSOWNER	Indicates that the owner name is to be returned.
	.1..	VSVOLSER	Indicates that the serial number is to be returned.
29(1D)	.6	VSVOLUME	Volume serial number used when opening the VTOC for label processing. It must be the serial number known to the MSC.
35(23)	...1	VS2MSG	Ignored.
36(24)	1	VS2RESOP	Reserve options:
	1...	VS2RES	Indicates that the volume is to be reserved prior to reading or updating the label.
	.1..	VS2REAL	Indicates that the RESERVE is done on a real volume. If this bit is off, the volume is virtual.
37(25)	.8	VSDDNAME	ddname for the volume.

Updating Volume Label Fields

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	VS2HEAD	Set with 'PUTLABEL' by IDCVS02.
8(8)	4	VSUCBPTR	Same as Retrieving Volume Label Fields.
12(C)	4	VSVTOCSZ	Ignored.
16(10)	4	VSOWNPTR	Address of owner name to be put in the volume if VSOWNER is set.
20(14)	4	VSVOLPTR	Address of the volume serial number to be placed in the volume label if VSVOLSER is set.
24(18)	4	VSIOBKPT	Ignored.
38(1C)		VSLABOPT	Label options:
	1...	VSOWNER	Indicates that the owner is to be updated.
	.1..	VSVOLSER	Indicates that the serial number is updated.
29(1D)	.6	VSVOLUME	Same as Retrieving Volume Label Fields.
35(23)	...1	VS2MSG	Message options:
	1...	VS2TOV	Indicates that error messages contain the words "TO VOLUME."
36(24)	1	VS2RESOP	Same as Retrieving Volume Label Fields.
37(25)	.8	VSDDNAME	ddname for the volume.

VS3AGL

Volume Data Set Service Argument List The VS3AGL is passed whenever IDCVS02 is called. It contains information to perform a SELECTDS function. This function searches the VTOC for specific data sets. The data set names selected in accordance with the caller's criteria are returned in a buffer.

Created by	Modified by	Used by	Size
Calling routine	IDCSV03	IDCSV03	59

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
0(0)	8	VS3HEAD	Set with 'SELECTDS' by IDCVS03.
8(8)	4	VS3LISTP	Pointer to the data set name array.
12(C)	4	VS3EXDSP	Pointer to an array of data sets to be excluded from the selection search.
16(10)	4	VS3LEVP	Pointer to data set name qualifier.
20(14)	2	VS3LEVLN	Length of qualifier.
22(16)	2	VS3AST	Displacement to asterisk within the qualifier pointed to in VS3LEVP.
24(18)	4	VS3EXPIR	Pointer to expiration date.
28(1C)	4	VS3CREAT	Pointer to creation date.
32(20)	4	VS3UCBP	Pointer to the UCB.
36(24)	4	VS3DEV	Device type of volume.
40(28)	4	VS3POOL	Storage pool ID for data set list (VSDBLOCK).
44(2C)	6	VS3VOL	Volume serial number.
50(32)	8	VS3DD	ddname for opening the VTOC.
58(3A)	1	VS3FLAGS	Flags.
	1...	VS3STAT	Return only VSDSTAT portion of VSDBLOCK. This bit is mutually exclusive with VS3USAGE and VS3SCR.
	.1..	VS3USAGE	Return only VSDUSAGE portion of VSDBLOCK. This bit is mutually exclusive with VS3STAT and VS3SCR.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
	..1.	VS3SCR	Return only VSDSCR portion of VSDBLOCK. This bit is mutually exclusive with VS3STAT and VS3USAGE.
	...1	VS3UNCAT	Return only data sets not cataloged.
	.... 1...	VS3SYSNM	Return only data sets with system generated names.

Diagnostic Aids

This chapter explains the diagnostic aids provided for access method services, explains how to find key areas in a dump, and offers suggestions for isolating different types of problems. Before attempting to diagnose a problem with the aids in this chapter, you should be familiar with *Debugging Handbook*. This and other publications that may be helpful are listed in the preface to this book.

Several large figures referred to throughout this chapter are located at the end of the chapter.

Four major diagnostic aids are provided by the processor:

- Trace tables, which provide a trace of the flow of control between modules and within modules.
- Dump points, which provide the facility to dump selected areas of virtual storage and take a full region dump.
- The test option, which you can set to print out the trace tables or to obtain dumps at selected points if access method services is invoked with a batched job.
- ABORT codes and full region dumps, which are produced when the processor detects an unrecoverable condition.

Trace Tables

The processor maintains two trace tables during each execution: the inter-module trace table, which records the flow of control *between* modules, and the intra-module trace table, which records the flow of control *within* modules.

You can find the trace tables in any full region dump, you can print them using the test option, or you can display them on a TSO terminal. The section "Reading a Dump" explains how to find the tables in a dump; "The Test Option" explains how to print them; "The TSO TEST Command" explains how to display them on a TSO terminal.

Inter-Module Trace Table

The inter-module trace table begins with the characters INTER and contains the IDs of the last 20 modules that had control. The module IDs are the last 4 characters of the module name. For example, if the trace looks like this:

```
INTER ... SA01 EX01 RI01 RI02
```

then you know that IDCRI02 had control at the time of the dump.

The inter-module trace table is updated by the system adapter not only as each module is entered, but also upon return from a module. Thus, if RI01 calls TP01 which calls IO01 and then returns back to RI01, the trace table looks like this:

```
INTER ... RI01 TP01 IO01 TP01 RI01
```

Intra-Module Trace Table

The intra-module trace table begins with the characters INTRA and contains the last 20 trace points encountered within modules. Each module has trace points placed at key locations, for example, at the start of procedures and around calls to other modules.

The IDs of the trace points consist of 4 characters: the first 2 characters are the mnemonic identifier of the module being traced, and the last 2 characters identify a specific point within the module. (The mnemonic identifiers are listed in the section "Naming Conventions" in the "Introduction.")

The section "Trace and Dump Points to Module Cross-Reference" in this chapter contains a list of all the trace points, identifies the module and procedure in which the trace point occurs, and explains the situation at the trace point. For example, if the intra-module trace table looks like this:

```
INTRA ... SAGS IOOP SACL SAGP
```

then, using this list, you would know that the last trace point encountered was at the start of the routine in module IDCSA02 that processes a UGPOOL macro request.

During the time the test option is on, the dumping routine (IDCDB01) places dump points in the intra-module trace table; thus, the trace table contains all the dump points encountered as well as the trace points. All the dump points you may find in the intra-module trace table, in addition to the trace points, are explained in the section "Trace and Dump Points to Module Cross-Reference" in this chapter.

Trace points within a module can be found by examining the microfiche listings for occurrences of the UTRACE macro; the UTRACE macro sets the trace IDs into the trace table. The expansion of the UTRACE macro for trace ID DLLC looks like this:

```
OLDERID2 = NEWERID2;
NEWID2 = 'DLLC';
```

Dump Points

Each module has built-in dump points that invoke diagnostic dumping routines if the test option is in effect. The dump points, set up by the UDUMP macro, have been placed at key locations in each module (for example, around calls to other processor and nonprocessor modules). Each dump point specifies the information that can be dumped at that point. Some dump points allow symbolic dumping of selected areas of virtual storage (for example, parameter lists or return codes); all dump points allow dumping of the full region and printing of the trace tables.

Dump points can be found by examining microfiche listings for occurrences of the UDUMP macro. The expansion of the UDUMP macro for the dump point DLVL looks like this:

```
IF GDTDBG = NULLPTR
  THEN;
  ELSE
     CALL IDCB010(GDTTBL,'DLVL');
```

Only the trace tables and the full region can be dumped at this point because only two parameters, the GDTTBL and the dump ID, are passed to the dumping routine.

The section "Module to Dump Points Cross-Reference" in this chapter contains a list of all the dump points within each module, indicates what information can be dumped, and explains the situation at the dump point. The section "Test Option" in this chapter explains how to take a full region dump.

Dumping Selected Areas of Virtual Storage

Certain access method services modules have the dumping of selected areas of virtual storage built in. Dumping of these selected areas occurs at a dump point as described above. The areas dumped vary with each dump point and are identified with descriptive codes. The list in the section "Module to Dump Points Cross-Reference" in this chapter indicates which modules contain dumps of selected areas and the footnotes to that list describe the areas dumped.

Dump points at which selected areas are printed can be found by examining the microfiche listings for occurrences of the UDUMP macro. The expansion is as described above for a full region dump except that the address of a parameter list describing the areas to be dumped is passed to the dumping routing as a third parameter.

Dumping of selected areas can occur with or without a full region dump in addition, as described in the section "Test Option" in this chapter.

Test Option

If you invoked access method services in a batched job, you can use the test option to activate the printing of diagnostic output at selected points within access method services. The test option is controlled by the TEST keyword as explained in the following section "TEST Keyword."

The test option provides you with the ability to print:

- The inter-module and intra-module trace tables. The format and interpretation of these tables are described in the section "Trace Tables" in this chapter.
- Selected areas of virtual storage. The facility for dumping selected areas of virtual storage is described in the section "Dump Points" in this chapter.
- Full region dump. The facility for taking a full region dump is described in the section "Dump Points" in this chapter.

Each variation of the test option provides an additional level of information. The possible variations are: (1) print the trace tables only; (2) print the trace tables and selected areas of virtual storage; (3) print the trace tables and selected areas of virtual storage and take a full region dump.

TEST Keyword

You can enter the TEST keyword either in the PARM field of the EXEC card that invokes the processor, or on a PARM command. By using the PARM command, you can turn the test option on and off or change the Test option for different function commands.

The format of the TEST keyword and its subparameters is:

```
PARM TEST( ([TRACE]
            [AREAS( ID-list )...)]
            [FULL(( dumplist)...)] |
            [ OFF] ))
```

where the subparameters are defined as follows:

TRACE specifies that the inter-module and intra-module trace tables are to be printed at every dump point encountered.

AREAS names the modules for which selected areas are to be printed, *in addition* to the trace tables. The trace tables are printed at each dump point encountered within the named modules; if a dump point specifies selected areas to be dumped, these areas are printed also. *ID-list* is a string of 2-character mnemonic identifiers separated by commands and/or blanks. The mnemonic identifiers are listed in the section "Naming Conventions" in the chapter "Introduction." The mnemonic identifier, however, for the dump points within system adapter dump points is ZZ. The maximum number of identifiers is 10. For example, AREAS(EX,PR) specifies that selective dumping is to occur in the executive modules and the PRINT FSR.

FULL names the dump points at which full region dumps are to be produced, *in addition* to the selected areas and the trace tables. The trace tables and selected areas are produced each time the dump point is encountered; a full region dump is produced as specified in *dumplist*. *dumplist* consists of a string of triplets enclosed in parentheses. The maximum number of triplets is 10. Each triplet is of the form:

(*ident* [*begin* [*count*]])

where the arguments of the triplet are defined as follows:

ident is a 4-character dump point. The dump points are identified in UDUMP macros and are listed in the module to dump points cross-reference list.

begin specifies the iteration through the named dump point at which you wish the full region dump to be produced. For example, a *begin* value of 2 specifies that a full region dump is not to be produced until the second encounter of the dump point. The default value is 1, and the maximum is 32767.

count specifies the number of times the full region dump is to be produced, once the value of *begin* has been satisfied. The default value is 1, and the maximum is 32767.

For example, FULL((EX1F,4,2),(AL01)) specifies that one full region dump is to be produced the fourth time that point EX1F is encountered, another full region dump is to be produced the fifth time the point is encountered, and one full region dump is to be produced the first time that point AL01 is encountered. Trace tables and any selected areas are to be printed each time dump points EXIF and AL01 is encountered. If the FULL keyword is used, then an AMSDUMP DD statement must be provided. for example:

```
//AMSDUMP DD SYSOUT=A
```

OFF turns off the test option. No further dumping of trace tables, selected areas, or region will occur until another PARM command specifies one of the other subparameters. This subparameter must occur alone; it may not be coded with any other subparameter of the TEST keyword.

Each time a PARM command is specified, the TEST parameters override the TEST parameters in effect from the previous PARM command.

Figure 14 on page 248 shows a section of the output from the command:

```
PARM TEST ( FULL (LCTP,2,1) )
```

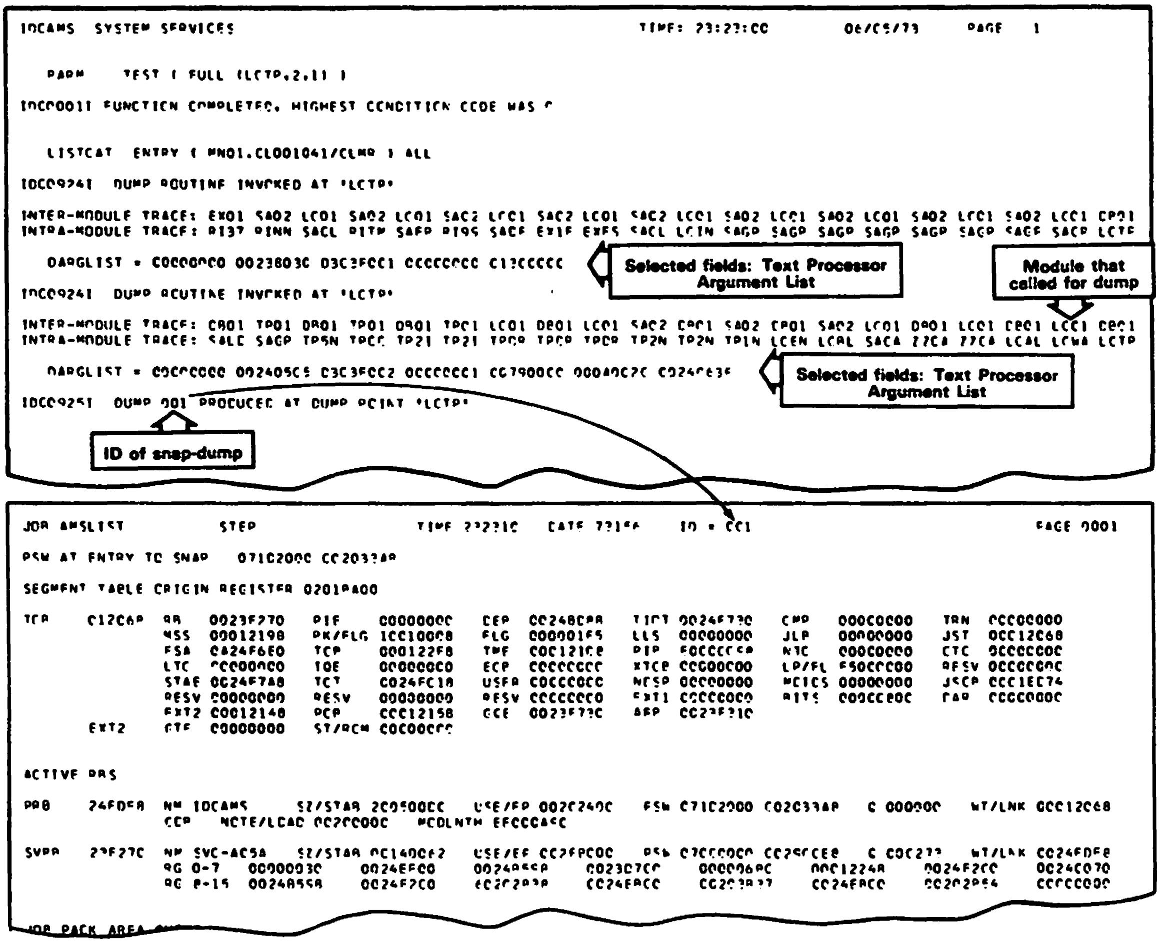
The trace tables and the selected area, DARGLIST, are printed each time the dump point LCTP is encountered. A full region dump is produced the second time that dump point LCTP is encountered.

How to Use the Test Option

If a problem occurs and you have no idea which modules are involved, run the job again with the TRACE keyword. From the inter-module trace table you should be able to tell the modules involved. The TRACE keyword, however, produces a large amount of output.

If you suspect which modules are involved, you can rerun the job with the AREAS keyword and specify the identifiers of several suspected modules. You will obtain trace output for only the specified modules.

Once you know the procedure within a module that has caused the problem, select the dump points at which you would like a full dump (using the module to dump points cross-reference list or by examining the microfiche for dump points), and rerun the job with the FULL keyword. The AREAS and FULL keywords can be used in combination to obtain trace tables and selected areas throughout several modules, but a full region dump only at selected points.



```
IDCAMS  SYSTEM SERVICES                                          TIME: 23:23:CC        06/C5/73      PAGE   1

   PARM    TEST ( FULL (LCTP,2,11 )

IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

   LISTCAT  ENTRY ( MN01.CL001041/CLMP ) ALL

IDC0924I  DUMP ROUTINE INVOKED AT 'LCTP'

INTER-MODULE TRACE: EX01 SA02 LC01 SA02 LC01 SAC2 LCC1 SAC2 LC01 SAC2 LCC1 SA02 LCC1 SA02 LC01 SA02 LCC1 SA02 LCC1 CP01
INTRA-MODULE TRACE: R137 RINN SACL RITM SAFP R195 SACF EX1F EXFS SACL LCIN SAGP SAGP SAGP SAGP SAGP SAGP SACF SACP LCTP

   DARGLIST = COCCOOCO 0023803C D3C3FCC1 CCCCCCCC C13CCCCC

IDC0924I  DUMP ROUTINE INVOKED AT 'LCTP'

INTER-MODULE TRACE: CR01 TP01 DR01 TP01 DR01 TPC1 LC01 DE01 LCC1 SAC2 CRC1 SA02 CR01 SAC2 LC01 DR01 LCC1 CRC1 LCC1 CRC1
INTRA-MODULE TRACE: SALC SAGP TP5N TPCC TP21 TP21 TPCR TPCR TPCR TP2N TP2N TP1N LCEN LCAL SACA ZZCA ZZCA LCAL LCWA LCTP

   DARGLIST = C9CCCCCC 002405C5 C3C3FCC2 OCCCCCC1 CC7900CC 000A0C2C C024CE3F

IDC0925I  DUMP 001 PRODUCED AT DUMP POINT 'LCTP'
```

Selected fields: Text Processor Argument List
Module that called for dump
Selected fields: Text Processor Argument List
ID of snap-dump

```
JOB AMSLIST            STEP                      TIME 232210   DATE 77156      ID = CC1                              PAGE 0001

PSW AT ENTRY TO SNAP   071C200C CC2033AP

SEGMENT TABLE ORIGIN REGISTER 0201PA00

TCB     C12C6P   RB    0023F270   PIE     C000000C   DEB   CC24BCPR   TIOT 0024F73C   CMP    000C0C00   TRN   CCCOC000
                 MSS   00012198   PK/FLG  1CC1000B   FLG   000001F5   LLS  00000000   JLP    00000000   JST   0CC12C68
                 FSA   CA24F6E0   TCB     000122F8   TMF   C0C121CP   PIP  F0CCCCSA   NTC    000C0CC0   OTC   0CCCCC0C
                 LTC   CCC000C0   TOE     00000CC0   ECP   CCCCCCCC   XTCB CCG00C00   LP/FL  F50CCC00   RFSV  0CCCCC0C
                 STAE  0C24F7A8   TCT     C024FC18   USFR  C0CCCOCC   NFSP 0CCC0000   MCICS  00000000   JSCB  CCC1EC74
                 RESV  C0000000   RESV    00000000   RESV  CCCCCCC0   EXT1 CCCCC0C0   RITS   C09CCE0C   DAR   CCCC000C
                 EXT2  C0012140   RCP     CCC12158   CCE   0023F73C   AFP  CC23F21C
        EXT2     GTF   C0000000   ST/RCM  C0C00CCC

ACTIVE RBS

PRB     24FDFR   NM  IDCAMS      SZ/STAB 2C0500CC    USE/EP 002C240C    PSW C71C2000 C02C33AP    C 00000C    WT/LNK 0CC12C68
                 CCP    NCTE/LCAC CC2CC00C    MCDLNTH EFCCCASC

SVRB    23F27C   NM  SVC-AC5A    SZ/STAB 0C140C62    USE/EF CC2FPCOC    PSW C7CCC0CC CC2SCCE8    C C0C233    WT/LNK CO24FDF8
                 RG 0-7   0000003C      0024EFC0      0024R5SA      C023C7CC      000C06AC      00C1224R      0024F2CC      0C24C070
                 RG 8-15  0024R55R      0C24F7C0      CC2C2R3P      CC24FACC      CC2C3R37      CC24FACC      CC202PE4      CCCCC000

JOB PACK AREA
```

Figure 14. Example of Test Option Output

Trace and Dump Points to Module Cross-Reference

The following list contains all trace and dump points, identifies the containing module and procedure, and explains the situation at the trace or dump point. When the test option is set, both the trace and dump points are placed in the intra-module trace table. The trace tables are printed with all variations of the test option as explained in the section "TEST Keyword."

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
AA00	IDCSA09	ISSUEMAC	dump	After issuing SVC 126.
AA01	IDCSA09	CHKCODE	dump	After delayed response for mass storage control order.
ALMR	IDCAL01	MEMRENAM	trace	Start of procedure that renames members of partitioned data sets.
AL01	IDCAL01	IDCAL01	dump	Before calling the catalog to alter an object.
			trace	Start of ALTER FSR.
AL02	IDCAL01	IDCAL01	dump	End of ALTER FSR.
AL03	IDCAL01	LOCATPRC	dump	After calling the catalog to locate an object.
AL04	IDCAL01	IDCAL01	dump	Before issuing ALTER request for index object if KEYS specified.
AL31	IDCAL01	LOCATPRC	trace	Start of procedure that locates the entry to be altered.
AL41	IDCAL01	ALTERPRC	trace	Start of procedure that alters the catalog entry.
AL51	IDCAL01	CHECKPRC	trace	Entry to CHECKPRC.
			dump	After locating data component of the alternate index for which UPGRADE has been specified.
AL52	IDCAL01	CHECKPRC	dump	After locating associated cluster or alternate index of the data object specified on ALTER command.
AL53	IDCAL01	CHECKPRC	dump	After locating associated index component.
AL54	IDCAL01	CHECKPRC	dump	After locating the data component of the path's base cluster.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
AL55	IDCAL01	CHECKPRC	dump	After locating the cluster component of the alternate index's base cluster.
AL56	IDCAL01	CHECKPRC	dump	After locating the data component of the alternate index's base cluster.
AL61	IDCAL01	INDEXPRC	trace	On entry to INDEXPRC.
AL71	IDCAL01	DALCPROC	dump	Start of procedure that allocates and deallocates volumes.
AL81	IDCAL01	PARAMCHK	trace	On entry to parameter checking procedure.
AL91	IDCAL01	ASSOCPRC	trace	Start of procedure that determines index association of an ICF catalog.
AL92	IDCAL01	ASSOCPRC	dump	After the association, locate call to CMS.
AL93	IDCAL01	ASSOCPRC	dump	After the DSATTR, locate call to CMS.
BD01	IDCBD01	IDCBD01	trace	Start of BINDDATA FSR.
BDST	IDCBD01	IDCBD01	dump	Start of BINDDATA FSR.
BDG4	IDCBD01	IDCBD01	both	Before link to IDCSS04.
BDR4	IDCBD01	IDCBD01	both	After return from IDCSS04.
BDED	IDCBD01	IDCBD01	both	End of BINDDATA FSR.
BIB1	IDCBI01	BLDPROC	trace	First entry to procedure that builds and writes the alternate index records.
BIC1	IDCBI01	CNTLPROC	trace	Start of procedure that controls reading base cluster, sorting and writing alternate index.
BIC2	IDCBI01	CNTLPROC	dump	After completion of sort if an internal sort; after completion of sort phase and before merge passes if an external sort.
BIDL	IDCBI01	DELTPROC	trace	Start of procedure that deletes sort work files.
			dump	After return from UCATLG to delete each sort work file.
BID1	IDCBI01	DEFPROC	trace	Start of procedure that defines sort work files.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
BID2	IDCBI01	DEFPROC	dump	After return from UCATLG to define each sort work file.
BIF1	IDCBI01	FINPROC	trace	Start of procedure that closes alternate index and prints status message.
BII1	IDCBI01	INITPROC	trace	Start of procedure that obtains resources for building alternate index.
BII2	IDCBI01	INITPROC	dump	After obtaining or failing to obtain sort core.
BIJ1	IDCBI01	JCPROC	trace	Start of procedure that issues UIOINFO to obtain sort work file job control data.
BIJ2	IDCBI01	JCPROC	dump	After return from each call to UIOINFO.
BIL1	IDCBI01	LOCPROC	trace	Start of procedure that controls catalog locates to obtain information about the base cluster and alternate index.
BIL2	IDCBI01	CATPROC	dump	After return from UCATLG for each locate request.
BIM1	IDCBI01	MERGPROC	trace	Start of procedure that performs the merge passes of an external sort.
BIM2	IDCBI01	MERGPROC	trace	Start of each merge pass of an external sort.
BIM3	IDCBI01	MERGPROC	dump	After the tree of nodes has been initialized for each merge pass of an external sort.
BIM4	IDCBI01	MERGPROC	dump	After processing one set of strings during the merge pass of an external sort.
BIP1	IDCBI01	OPENPROC	trace	Start of procedure that opens data sets.
BIP2	IDCBI01	OPENPROC	dump	After return from UOPEN to open a data set.
BISP	IDCBI01	SPILPROC	trace	Start of procedure that writes out a sorted string in the sort phase of an external sort.
BISR	IDCBI01	SORTPROC	dump	Before sorting the records in the record sort area.
BI01	IDCBI01	IDCBI01	trace	Start of BLDINDEX FSR.
BI02	IDCBI01	MAINPROC	trace	Start of procedure that controls building of one alternate index.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
BI03	IDCBI01	MAINPROC	dump	After return from procedure which locates information about the base cluster and alternate index.
BI04	IDCBI01	MAINPROC	dump	After the alternate index has been built; before close.
CCAC	IDCCC01	ALTRLSTS	dump	After building parameter lists to alter a VSAM or ICF catalog.
CCAI	IDCCC02	CNVTALIS	trace	Start of procedure that processes alias entries.
CCAL	IDCCC01	ALTRLSTS	trace	Before building parameter lists to alter a VSAM or ICF catalog.
CCAP	IDCCC01	AEPROC	trace	Ready to process an 'AE' catalog entry from an OS/VS catalog.
CCAS	IDCCC02	ASSOCTNS	trace	Start of procedure that processes associations for base spheres.
CCAT	IDCCC01	CATALOG	trace	Ready to build VSAM parameter lists and alter the non-VSAM entry in the target catalog.
CCAV	IDCCC01	ALTRLSTS	dump	After building parameter lists to alter a VSAM or ICF catalog.
CCAX	IDCCC02	CNVTAIX	trace	Start of procedure that processes alternate index entries.
CCBE	IDCCC02	IDCCC02	trace	Start of IDCCC02.
CCBG	IDCCC01	IDCCC01	trace	Start of CNVTCAT FSR.
CCCE	IDCCC01	CONTINUE	trace	Ready to get a block at the same index level as the last block from an OS/VS catalog.
CCCG	IDCCC02	CTLGPROC	trace	Start of procedure that calls catalog management.
CCCI	IDCCC01	CNVTINIT	trace	Ready to initialize for scanning OS/VS catalog.
CCCK	IDCCC01	CATCHECK	trace	Start of procedure verifying that catalog types are valid for convert.
CCCL	IDCCC02	CNVTCLUS	trace	Start of procedure that processes VSAM cluster entries.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
CCCN	IDCCC01	CONTINUE	dump	After obtained block at the same index level as the last block from an OS/VS catalog.
CCCP	IDCCC01	CVPEPROC	trace	Ready to process a 'CVPE' catalog entry from an OS/VS catalog.
CCCT	IDCCC01	CATALOG	trace	Ready to build VSAM argument lists and define converted OS/VS catalog entry in the target catalog.
CCCV	IDCCC01	CONVERT	trace	Ready to convert OS/VS catalog entries to VSAM or ICF catalog entries.
CCDA	IDCCC02	DALCPROC	trace	Start of procedure that dynamically allocates and deallocates volumes.
CCDC	IDCCC01	DEFNLSTS	dump	After building VSAM argument lists for define.
CCDE	IDCCC02	DEFCLUS	trace	Start of procedure that defines VSAM clusters and AIXs into the ICF catalog.
CCDL	IDCCC01	DEFNLSTS	trace	Ready to build VSAM argument lists for define.
CCDN	IDCCC02	DEFNVSM	trace	Start of procedure that defines non-VSAM data sets.
CCDP	IDCCC01	DSPEPROC	trace	Ready to process a 'DSPE' catalog entry from an OS/VS catalog.
CCDS	IDCCC02	DELSPHRE	trace	Start of procedure that deletes a sphere from a catalog.
CCDV	IDCCC01	DEFNLSTS	dump	After building VSAM argument lists for define.
CCER	IDCCC01	ERRPROC	trace	Start of procedure that invokes UERROR when a catalog error (SVC26) is encountered.
CCFP	IDCCC02	LFPLBLD	trace	Start of procedure that builds the catalog field parameter lists.
CCFV	IDCCC02	FVTPROC	trace	Start of procedure that builds a field vector table for VSAM entries.
CCGA	IDCCC02	GDGASOC	trace	Start of procedure that processes GDG associations.
CCGD	IDCCC02	CNVTGDG	trace	Start of procedure that processes GDG base entries.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
CCGP	IDCCC01	GIPERPROC	trace	Ready to process a 'GIPE' catalog entry from an OS/VS catalog.
CCIC	IDCCC01	CNVINIT	trace	After initializing for scanning of OS/VS catalog.
CCIE	IDCCC01	PUSHDOWN	dump	No 'ICE' entry found on a lower level scanning in the OS/VS catalog.
CCIN	IDCCC01	INITPROC	trace	Start of initialization procedure.
CCIP	IDCCC01	ILEPROC	trace	Ready to process a 'ILE' catalog entry from an OS/VS catalog.
CCLC	IDCCC01	LOCTLSTS	dump	After building parameter lists to locate information in the VSAM or ICF catalog.
CCLI	IDCCC02	LISTCHK	trace	Start of procedure that prints name of entry defined.
CCLL	IDCCC01	LOCTLSTS	trace	Before building parameter lists to locate information in the VSAM or ICF catalog.
CCLT	IDCCC01	CATALOG	trace	Ready to build VSAM argument lists to locate information in the VSAM or ICF catalog about a duplicate entry.
CCLV	IDCCC01	CATALOG	dump	After VSAM returns volume information on duplicate entry.
CCMV	IDCCC02	MVDAPROC	trace	Start of procedure that moves data.
CCNF	IDCCC02	NFVTPROC	trace	Start of procedure that builds a field vector table for non-VSAM entries.
CCNV	IDCCC02	CNVTNVSM	trace	Start of procedure that processes non-VSAM data sets.
CCOL	IDCCC01	CVOLCNVT	trace	Start of process to convert OS/VS catalog entries to VSAM or ICF catalog entries.
CCPA	IDCCC02	CNVTPATH	trace	Start of procedure that processes path entries.
CCPL	IDCCC02	CPLPROC	trace	Start of procedure that builds catalog parameter list.
CCPD	IDCCC01	PUSHDOWN	dump	After obtaining block at the same index level than the last block from the OS/VS catalog.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
CCPH	IDCCC01	PUSHDOWN	trace	Ready to obtain a block at a lower index level than the last block from the OS/VS catalog.
CCPP	IDCCC01	POPUP	trace	Ready to return to higher level index block in OS/VS catalog.
CCPU	IDCCC01	POPUP	dump	Ready to continue scan of higher level index block in OS/VS catalog.
CCRG	IDCCC02	RANGPROC	trace	Start of procedure that builds high and low KEYRANGE lists.
CCSE	IDCCC01	CONVERT	dump	Invalid entry type in the OS/VS catalog.
CCSL	IDCCC01	CONVERT	dump	Finished converting OS/VS catalog entries to VSAM or ICF catalog entires.
CCSP	IDCCC02	DELSPACE	trace	Start of procedure that deletes data spaces.
CCTM	IDCCC01	TERMPROC	trace	Start of termination procedure.
CCTY	IDCCC01	CATTYPE	trace	Start of procedure that determines catalog type.
CCUC	IDCCC02	CNVTUCAT	trace	Start of procedure that processes user catalog connector entries.
CCVC	IDCCC01	VSAMCNVT	trace	Start of procedure that controls conversion of VSAM catalog entries to ICF catalog entries.
CCVE	IDCCC01	VOLINDEX	dump	No 'VICE' entry in the OS/VS catalog.
CCVI	IDCCC01	VOLINDEX	dump	First block of volume index is obtained from the OS/VS catalog.
CCVL	IDCCC02	LVLRPROC	trace	Start of procedure that builds volume serials list, device types list, and file sequence numbers list.
CCVP	IDCCC01	VCBPPROC	trace	Ready to process a 'VCBPE' and 'VCB' entry from the OS/VS catalog.
CCVV	IDCCC02	DEFVVDS	trace	Start of procedure that defines VSAM volume data sets.
CCVX	IDCCC01	VOLINDEX	trace	Ready to get first volume index block from the OS/VS catalog.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
CCXA	IDCCC02	XASSOCNS	trace	Start of procedure that processes associations for alternate indexes and GDG base associations.
CCZY	IDCCC02	CTLGPROC	dump	Dump point prior to catalog management call.
CCZZ	IDCCC02	CTLGPROC	dump	Dump point following catalog management call.
CKBD	IDCCK01	BUILDTAB	trace	Start of procedure that builds the checkid table.
CKCK	IDCCK01	IDCCK01	trace	Start of procedure that builds the checkid table.
CKCK	IDCCK01	IDCCK01	trace	Start of CHKLIST FSR.
CKCP	IDCCK01	CHRPROC	trace	Start of procedure that processes CHR records.
CKDI	IDCCK01	DSDRLIST	trace	Start of procedure that prints the volume serial numbers.
CKDP	IDCCK01	DSDRPROC	trace	Start of procedure that processes DSDR records.
CKDV	IDCCK01	DSDRVOLS	trace	Start of procedure that processes type 2 DSDR records.
CKGC	IDCCK01	GETCHR	trace	Start of procedure that reads CHR records.
CKGN	IDCCK01	GETNEXT	trace	Start of procedure that locates the next logical DSDR record.
CKHS	IDCCK01	HSKGPROC	trace	Start of housekeeping procedure.
CK10	IDCCK01	IDCCK01	dump	Before calling CHR procedure.
CK20	IDCCK01	IDCCK01	dump	Upon return from CHR procedure.
CK30	IDCCK01	DSDRPROC	dump	After reading a type 1 DSDR record.
CK40	IDCCK01	DSDRPROC	dump	Before processing a type 1 DSDR record for a tape data set.
CK50	IDCCK01	DSDRVOLS	dump	Before processing a type 2 DSDR record.
CP14	IDCRP01	VERIFYC	trace	When either the source or target catalog cannot be verified during a reload.
DAZA	IDCDA02	MAIN	trace	GETMAIN the DA02 work area.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
DAZB	IDCDA03	MAIN	trace	GETMAIN the DA03 work area.
DAZY	IDCDA01	DUMPDAZY	dump	Procedure called before any termination code gets control.
DAZZ	IDCDA01	TERM	dump	Before storage is FREEMAINed.
DAA1	IDCDA03	CHKGAT	trace	Routine that searches the GAT, ensuring that all GDSs have been flagged as found.
DAA2	IDCDA03	CHKREL	trace	Procedure that locates any relationship cell entries which were not 'found.'
DAA3	IDCDA03	BLDGOVO	trace	Procedure to build a GDG name (in EBCDIC) from the passed values.
DAA4	IDCDA01	INITSAVE	trace	Procedure for saving the GAT and REL cells.
DAA5	IDCDA02	INITILST	trace	Routine that builds in the ILST from the command input.
DAA6	IDCDA02	ILSTCHK	trace	Routine that scans the ILST looking for match on "FULLNAME."
DAA7	IDCDA03	FINDQUAL	trace	Routine called to scan a character string, looking for a blank or a period or end of a pointer.
DAA8	IDCDA02	GETSEQ	trace	Elementary GET sequential procedure.
DAA9	IDCDA03	RECDUMP	trace	Procedure for the display of records: catalog, VVR, or VTOC.
DAB1	IDCDA03	CMPRVTOC	trace	Procedure for obtaining a VTOC DSCB set and comparing the extents with the VVR.
DAB2	IDCDA03	COMPAREV	trace	Procedure that will compare the VVR record with the corresponding BCS catalog record.
DAB3	IDCDA03	GETCMPRV	trace	Procedure that gets the catalog record for the VVR passed.
DAB4	IDCDA03	CMPRVVRC	trace	Procedure that compares the catalog record component volume cell entry against the VVR data.
DAB5	IDCA03	SKIPCMP	trace	If an error has occurred, jump to next component.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
DAB6	IDCDA03	XNCLDCHK	trace	Procedure that is called to determine if the component in FULLNAME should be kept or skipped.
DAB7	IDCA03	BLDRLST	trace	Procedure for obtaining an area large enough for an RLST if needed.
DAB8	IDCDA03	ADDRLST	trace	Procedure for adding an entry into a RLST.
DAB9	IDCDA03	SRCHRLST	trace	Procedure for finding a name in the RLST.
DAC1	IDCDA03	SUMCMPR	trace	Procedure for summarizing the compares not found.
DAC2	IDCDA03	XNVVRCHK	trace	Procedure for checking the VVR for INCLUDE/EXCLUDE.
DAC3	IDCDA01	CROUTINE	trace	Procedure for the 'C' component content checking after a format check has completed.
DAC4	IDCDA03	SCANVVCI	trace	Procedure that is called to scan a VVDS CI for a given component.
DAC5	IDCDA03	TESTVVR	trace	Procedure that is called to test a VVR for NAMECMP and matching record (Z/Q).
DAC6	IDCDA03	JOTERROR	trace	Procedure for handling any records which have an invalid or unrecognized record type.
DAC7	IDCDA03	CHKTSKIP	trace	Procedure for checking the name in the 'T' record for INCLUDE/EXCLUDE.
DAC8	IDCDA01	GROUTINE	trace	Procedure for the 'G' component content checking after a format check has completed.
DAC9	IDCDA01	HROUTINE	trace	Procedure for the 'H' component content checking after a format check has completed.
DA01	IDCDA01	MAIN	trace	At entry to diagnose.
DA02	IDCDA01	INIT	trace	Procedure responsible for the front end logic of the high-level main procedure.
DA03	ID0DA02	BLDELST	trace	Procedure for obtaining an area large enough for the ELST.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
DA04	IDCDA02	BLDILST	trace	Procedure for obtaining an area large enough for the ILST if needed.
DA05	IDCDA01	ICFPROC	trace	Procedure responsible for diagnose of an ICFCATALOG.
DA06	IDCDA01	VVDSPROC	trace	Procedure responsible for the diagnose of a VSAM volume data set (VVDS).
DA07	IDCDA01	INITCMPR	trace	Procedure responsible for initializing relative to 'COMPARE.'
DA08	IDCDA01	PRINTSUM	trace	Procedure for printing summary results of prior processing.
DA09	IDCDA01	TERM	trace	Procedure for a close of the alternate print file and a free all of allocated storage.
DA10	IDCDA01	INITICF	trace	Procedure responsible for front end logic for ICFPROC.
DA11	IDCDA02	BLDDLST	trace	Procedure for building the domain list (DLST).
DA12	IDCDA02	GETNEXT	trace	Procedure for obtaining the next record in sequential order.
DA13	IDCDA01	CHKREC	trace	Procedure for determining record type and calling the routine.
DA14	IDCDA01	INITVVDS	trace	Procedure responsible for initialization of logic for VVDSPROC.
DA15	IDCDA01	XROUTINE	trace	Procedure for processing the 'X' type record.
DA16	IDCDA01	FORMATCK	trace	Procedure for scanning a record, checking cell, component, subrecord, and record contents.
DA17	IDCDA01	GETTCELL	trace	Procedure for obtaining a given cell type from a component.
DA18	IDCDA03	TESTGDG	trace	Procedure for checking if FULLNAME is a GDG data set name.
DA19	IDCDA03	XPNDNAME	trace	Procedure for changing a compressed name to a 45-byte name.
DA20	IDCDA02	SRCHILST	trace	Procedure for looking at ILST's trying to find a name.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
DA21	IDCDA02	LOGERROR	trace	Procedure for logging error in the PLST and writing error messages.
DA22	IDCDA02	GETDIR	trace	Elementary get direct procedure.
DA23	IDCDA02	SRCHPLST	trace	Procedure for finding a name in the PLST.
DA24	IDCDA03	GTNXTCMP	trace	Procedure for looping through a record to find the start of the next component.
DA25	IDCDA02	SHRKNAME	trace	Procedure for reducing a name to a compressed name.
DA26	IDCDA01	SCANCMP	trace	Procedure for format checking a component in a record.
DA27	IDCDA01	TERMICF	trace	Procedure for the ending logic for an ICFCATALOG.
DA28	IDCDA01	TERMVVDS	trace	Procedure for ending logic for a VVDS.
DA29	IDCDA01	AROUTINE	trace	Procedure for the 'A' component content checking after a format check has completed.
DA30	IDCDA01	BROUTINE	trace	Procedure for the 'B' component content checking after a format check has completed.
DA31	IDCDA01	IROUTINE	trace	Procedure for the 'I' component content checking after a format check has completed.
DA32	IDCDA01	DROUTINE	trace	Procedure for the 'D' component content checking after a format check has completed.
DA33	IDCDA01	EROUTINE	trace	Procedure for the 'E' component content checking after a format check has completed.
DA34	IDCDA01	IROUTINE	trace	Procedure for the 'J' component content checking after a format check has completed.
DA35	IDCDA01	QROUTINE	trace	Procedure for the 'Q' component content checking after a format check has completed.
DA36	IDCDA01	RROUTINE	trace	Procedure for the 'R' component content checking after a format check has completed.
DA37	IDCDA01	TROUTINE	trace	Procedure for the 'T' component content checking after a format check has completed.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
DA38	IDCDA01	UROUTINE	trace	Procedure for the 'U' component content checking after a format check has completed.
DA39	IDCDA01	ZROUTINE	trace	Procedure for the 'Z' component content checking after a format check has completed.
DA40	IDCDA01	RINVALID	trace	Procedure for handling any records that have an invalid or unrecognized record type.
DA41	IDCDA02	BLDPLST	trace	Procedure for obtaining an area large enough for the PLST.
DA42	IDCDA02	BLDLSTS	trace	Procedure for building the initial ELST, PLST, ILST, and RLST.
DA44	IDCDA02	ADDDLST	trace	Procedure for adding an entry into a DLST.
DA45	IDCDA02	ADDPLST	trace	Procedure for adding an entry into a PLST.
DA46	IDCDA01	OPENALT	trace	Procedure for the open of OUTDD if present.
DA47	IDCDA01	BLDENQ	trace	Procedure for building and verifying the enqueue names.
DA48	IDCDA02	ADDEDLST	trace	Procedure for adding and entry to the ELST.
DA49	IDCDA02	ADDILST	trace	Procedure for inserting an entry into the ILST.
DA50	IDCDA02	GETCORE	trace	Procedure for obtaining the storage requested in GETSIZE.
DA51	IDCDA01	NFOVINFL	trace	Procedure that receives control from BLDENQ when an INFILE VVDS was coded.
DA52	IDCDA01	NFOCINDS	trace	Procedure that receives control from BLDENQ when INDS catalog was coded.
DA53	IDCDA01	NFOVINDS	trace	Procedure responsible for INDS, VVDS.
DA54	IDCDA01	NFOCINFL	trace	Procedure responsible for INFILE, CATALOG.
DA55	IDCDA01	CELL01	trace	Procedure responsible for cell type '01'.
DA56	IDCDA01	CELL02	trace	Procedure responsible for cell type '02'.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
DA57	IDCDA01	CELL03	trace	Procedure responsible for cell type '03'.
DA58	IDCDA01	CELL04	trace	Procedure responsible for cell type '04'.
DA59	IDCDA01	CELL05	trace	Procedure responsible for cell type '05'.
DA60	IDCDA01	CELL06	trace	Procedure responsible for cell type '06'.
DA61	IDCDA01	CELL21	trace	Procedure responsible for cell type '07'.
DA62	IDCDA01	CELL23	trace	Procedure responsible for cell type '23'.
DA63	IDCDA01	CELL60	trace	Procedure responsible for cell type '60'.
DA64	IDCDA01	CELLFF	trace	Procedure responsible for any cell types that are unrecognized.
DA65	IDCDA02	FLAGFCT	trace	Procedure for noting entries in an FCT. Also detects some errors.
DA66	IDCDA02	CKSRFCT	trace	Procedure for checking the subrecord FCT.
DA67	IDCDA02	CKRECFCT	trace	Procedure for checking the record FCT.
DA68	IDCDA02	CKCMPFCT	trace	Procedure for checking the component FCT.
DA69	IDCDA02	FAKEFCT	trace	Procedure for forcing the FCT to a valid state.
DA70	IDCDA03	SUMNOERR	trace	Procedure for summarizing entries with no errors.
DA71	IDCDA03	SUMINCL	trace	Procedure for summarizing the INCLUDES not found.
DA72	IDCDA03	SUMEXCL	trace	Procedure for summarizing the EXCLUDES not found.
DA73	IDCDA03	SUMVOLNF	trace	Procedure for summarizing the volumes not found.
DA74	IDCDA03	SUMCATNF	trace	Procedure for summarizing the catalogs not found.
DA75	IDCDA03	SUMERROR	trace	Procedure for summarizing the entries with errors.
DA76	IDCDA02	BLDCLST	trace	Procedure for building the CLST.
DA77	IDCDA01	TERMCMPR	trace	Procedure for closing/unallocating compare data sets.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
DA78	IDCDA02	XCLDCHK	trace	Procedure for checking the current component to see if it should be slipped.
DA79	IDCDA02	FINDCMP	trace	Procedure for finding a COMP with the same name as CMPNAME (a fullname).
DA80	IDCDA02	FINDAUXR	trace	Procedure for obtaining the base record and finding the requested component.
DA81	IDCDA01	CHKRXASC	trace	Procedure for checking the associations in R or X records.
DA82	IDCDA01	CHKTASC	trace	Procedure for checking associations in the 'T' record.
DA83	IDCDA01	FNDTASC	trace	Procedure for confirming that G, D, or I components are associated with a 'T' record.
DA84	IDCDA01	FINDGASC	trace	Procedure for confirming that a 'G' is associated with an 'R' record, as needed. Called by the GROUTINE.
DA85	IDCDA02	SRCHCLST	trace	Procedure for finding a 44-byte name in the CLST or a 6-byte volser in the CLST.
DA86	IDCDA01	FNDRXASC	trace	Procedure for association check of 'A', 'C', 'H', or 'U' records.
DA87	IDCDA02	COMPARE	trace	Routine called by BCS component routines to test and execute any comparison logic needed.
DA88	IDCDA02	CMPRCVVR	trace	Routine called by other routines to compare the BCS record volume entry against the VVR.
DA89	IDCDA01	GETCMPR	trace	Procedure called to UGET the VVR record needed for compare.
DA90	IDCDA02	ALLOCMPR	trace	Routine for allocating a compare data set.
DA91	IDCDA02	OPENCMPR	trace	Routine for opening a compare data set.
DA92	IDCDA02	FLAGDLST	trace	Procedure that will find or add a name to the domain list (DLST).
DA93	IDCDA02	SRCHDLST	trace	Procedure for finding a name in the domain list (DLST).

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
DA94	IDCDA02	INITDLST	trace	Procedure that initializes the domain list for ICFCAT and VVDS.
DA95	IDCDA03	SAVEGAT	trace	Routine called by BROUTINE to save off the GAT for later checking.
DA96	IDCDA03	SAVEREL	trace	Procedure that saves off the relationship cell for later checking.
DA97	IDCDA02	FLAGGAT	trace	Procedure that finds a G0000V00 in the GAT and flags it as 'found'.
DA98	IDCDA02	FLAGREL	trace	Procedure that locates an entry in the relationship cell copy and flags it as 'found'.
DA99	IDCDA03	BASECHK	trace	Procedure that is called to confirm either the save GAT cell or the saved REL cell.
DB2A	IDCDB02	ARRAYHDR	trace	Start of procedure that processes an array header dump element.
DB2B	IDCDB02	BCONVERT	trace	Start of procedure the converts a dump item to binary representation.
DB2C	IDCDB02	CCONVERT	trace	Start of procedure that converts a dump item to character representation.
DB2F	IDCDB02	FCONVERT	trace	Start of procedure that converts a dump item to fixed representation.
DB2H	IDCDB02	HCONVERT	trace	Start of procedure that converts a dump item to hex representation.
DB2I	IDCDB02	ITEMDUMP	trace	Start of procedure that processes an individual dump list element.
DB2N	IDCDB02	NAMEFLD	trace	Start of procedure that processes the dump element symbolic name.
DE01	IDCDE01	IDCDE01	dump	Before calling the catalog to define an object.
DE02	IDCDE01	IDCDE01	dump	End of DEFINE FSR, before completion message is issued.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
DE03	IDCDE02	MODELPRC	dump	After calling the catalog to locate a model object.
DE04	IDCDE02	MODELPRC	dump	End of procedure that built the model table.
DE11	IDCDE01	IDCDE01	trace	Start of DEFINE FSR.
DE20	IDCDE03	IDCDE03	trace	Entry to IDCDE03.
DE21	IDCDE03	CTLGPROC	trace	Start of procedure that defines a master or user catalog.
DE22	IDCDE03	DSETPROC	trace	Start of procedure that defines a cluster.
DE23	IDCDE03	DSPACPRC	trace	Start of procedure that defines a data space.
DE24	IDCDE03	NVSAMPRC	trace	Start of procedure that defines a non-VSAM data set.
DE25	IDCDE03	AIXPROC	trace	Start of procedure that defines an alternate index.
DE26	IDCDE03	PATHPROC	trace	Start of procedure that defines a path.
DE27	IDCDE01	DALCPROC	trace	Start of procedure that allocates and deallocates volumes.
DE30	IDCDE02	IDCDE02	trace	Entry to IDCDE02.
DE31	IDCDE02	NAMEPROC	trace	Start of procedure that builds CTGFLs with name, date, and exception exit information.
DE32	IDCDE02	ALLCPROC	trace	Start of procedure that builds CTGFLs for allocation information.
DE33	IDCDE02	KEYPROC	trace	Start of procedure that builds CTGFLs for key range and AMDSBCAT information.
DE34	IDCDE02	PROTPROC	trace	Start of procedure that builds CTGFLs for protection and RGATTR information.
DE35	IDCDE02	IXOPPROC	trace	Start of procedure that initializes index fields in the AMDSBCAT.
DE36	IDCDE02	MODELPRC	trace	Start of procedure that locates the model object entry.
DE37	IDCDE02	FREESTG	dump	End of IDCDE02 CSECT.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
DLBC	IDCDL01	BUILDCPL	trace	Start of procedure that builds the CTGPL for the delete request.
DLBG	IDCDL01	IDCDL01	dump	Start of DELETE FSR.
DLCT	IDCDL01	CATCALL	trace	Start of procedure that calls the catalog with a delete request.
DLLC	IDCDL01	FINDTYPE	trace	Start of procedure that locates the type of the entry to be deleted.
DLMS	IDCDL01	MORESP	trace	Entry to MORESP.
DLPC	IDCDL01	PARAMCHK	trace	Start of procedure that checks for invalid parameters.
DLLB	IDCDL01	FINDTYPE	dump	Before calling the catalog to locate the entry type.
DLLA	IDCDL01	FINDTYPE	dump	After calling the catalog to locate the entry type.
DLDB	IDCDL01	CATCALL	dump	Before calling the catalog to delete an entry.
DLDA	IDCDL01	CATCALL	dump	After calling the catalog to delete an entry.
DLMD	IDCDL01	MEMDLETE	trace	Start of procedure that deletes PDS members.
DLAL	IDCDL01	ALLOPROC	trace	Start of procedure that dynamically allocates a data set or volumes.
DLAC	IDCDL01	RC240PRC	trace	Start of procedure that processes a VSAM catalog management return code of 240 when the CAT parameter is coded.
EXFS	IDCEX01	CALLFSR	dump	Before each call to an FSR.
EXIF	IDCEX01	CALLFSR	trace	Before each call to an FSR.
EXIM	IDCEX01	MAIN	trace	Before calling the reader/interpreter for the first time.
EXIR	IDCEX01	CALLRI	trace	Before each call to the reader/interpreter.
EXMN	IDCEX01	IDCEX01	dump	All reader/interpreter and FSR processing is complete.
EXRI	IDCEX01	CALLRI	dump	Before each call to the reader/interpreter.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
EX2S	IDCEX02	SCANPARM	trace	Before processing the caller's parameter list.
EX3S	IDCEX03	SCANPARM	trace	Before processing the caller's parameter list.
IOAB	IDCIO05	OCABEND	dump	Start of OPEN/CLOSE ABEND routine that sets flag in IOXCTLBK.
			trace	Start of procedure that sets IOXABEND flag in IOXCTLBK.
IOAC	IDCIO02	BUILDACB	dump	After ACB and EXLST have been built, at end of procedure.
			trace	Start of procedure that builds the ACB and EXLST.
IOAJ	IDCIO05	ADJCCW	trace	Start of procedure that adjusts a CCW channel program.
IOBB	IDCIO05	BLDBLK	trace	Start of procedure that issues UGSPACE, calls READJFCB, and writes IOXCTLBK.
IOBJ	IDCIO03	OBTNRTN	trace	Start of routine that obtains a DSCB by data set name or CCHHR.
IOCB	IDCIO01	CRYPTWH	trace	Start of procedure that builds and writes the encipher data set header.
IOCD	IDCIO01	CRYPTCP	trace	Start of procedure that enciphers/deciphers data.
IOCI	IDCIO01	IDCIOCR	trace	Start of encipher/decipher routine.
IOCK	IDCIO05	CKOPN	trace	Start of procedure that tests OPEN ABEND and issues UPRINT macros.
IOCL	IDCIO01	IDCIOCL	trace	Start of routine that closes data set.
IOCL	IDCIO05	CLOSEPRC	trace	Start of procedure that closes data sets.
IOCN	IDCIO05	CLOSENEW	trace	Start of procedure that issues SVC 82 to free the DEB.
IOCP	IDCIO01	IDCIOCO	trace	Start of routine that copies a data set.
IOCR	IDCIO01	CRYPTRH	trace	Start of procedure that reads and verifies the encipher data set header.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
IOCS	IDCIO05	CLOSESTD	trace	Start of procedure that issues CLOSE.
IOCT	IDCIO05	READCNT	trace	Start of procedure that reads the count field of each record on a track.
IOCW	IDCIO05	SETCCW	trace	Start of procedure that builds the CCW channel program to read the count field of each record.
IOC1	IDCIO05	READCNT	trace	Beginning of routine that determines the number of record read.
IOC2	IDCIO01	CRYPTWH	dump	End of procedure that builds and writes the encipher data set header.
IOC3	IDCIO01	CRYPTRH	dump	End of procedure that reads and verifies the encipher data set header.
IOC4	IDCIO01	IDCIOCR	dump	After detection of an error while processing the encipher data set header.
IODS	IDCIO02	DSDATA	dump	After obtaining data set information from the JFCB.
IOEG	IDCIO01	GETEXT	dump	End of procedure that gets a record from the user routine.
			trace	Start of procedure that gets a record from the user routine.
IOEP	IDCIO01	PUTEXT	dump	After control returns from an external user routine.
			trace	Before record is passed to an external user routine.
IOEX	IDCIO05	IDCIO05	dump	End of IDCIO05 module.
IOFB	IDCIO05	FWRITE	trace	Before calling UEXCP to do I/O.
IOFW	IDCIO05	FWRITE	trace	Beginning of procedure that writes multiple records on a track.
IOGR	IDCIO01	PUTREP	dump	After the GET for update.
IOGT	IDCIO01	IDCIOGT	trace	Beginning of routine that gets a data record from a data set.
IOIF	IDCIO03	DSINFO	trace	Entry to UIOINFO processing.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
IOIN	IDCIO05	ICIOO05	dump	Start of IDCIO05 module that performs steps necessary to open, close, read, or write volumes.
IOIT	IDCIO01	IDCIOIT	trace	Start of initialization routine.
IOKD	IDCIO05	READKD	trace	Start of procedure that reads the count, key, and data fields of each record on a track.
IOKS	IDCIO05	SETKDCCW	trace	Start of procedure that builds the CCW channel program to read the count, key, and data fields of each record on a track.
IOK2	IDCIO05	READKD	trace	Start of procedure that calculates the number of records to read.
IOK3	IDCIO05	SETKDCCW	trace	Start of procedure that builds read count-key-data CCWs.
IOK4	IDCIO05	SETKDCCW	trace	Start of procedure bypasses bad count fields.
IOOC	IDCIO05	OCABEND	trace	Start of open/close ABEND routine.
IOOE	IDCIO05	OPENPROC	trace	Start of procedure that determines type of open.
IOOG	IDCIO01	GETNONVS	trace	Start of procedure to get a non-VSAM logical record.
IOOL	IDCIO05	OPNLAB	trace	Start of procedure that opens VTOC to read or write volume label.
IOON	IDCIO05	OPNNEW	trace	Start of procedure that issues SVC 82 to create a DEB.
IOOP	IDCIO01	IDCIOOP	trace	Start of routine that opens data sets.
IOOP	IDCIO05	OPNPASS	trace	Start of procedure that opens data set to verify passwords or expiration dates.
IOOR	IDCIO05	OPENR	trace	Start of procedure that opens data sets, VTOCs, and staging packs for REPAIRV.
IOOT	IDCIO03	PTISDS	trace	Before SETL macro is issued.
IOOT	IDCIO05	OPNTAB	trace	Start of procedure that opens data set to copy the MSC tables.
IOOV	IDCIO05	OPNVTOC	trace	Start of procedure that opens the VTOC.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
IOOW	IDCIO01	PUTNONVS	trace	Start of procedure to write a non-VSAM logical record.
IOPG	IDCIO05	PUTGET	trace	Start of procedure that frees CCW storage.
IOPL	IDCIO01	PUTREP	trace	Entry to PUT (replace) routine.
IOPO	IDCIO01	IDCIOPO	trace	Start of routine that positions to a data record in an opened VSAM or ISAM data set.
IOPO	IDCIO03	IDCIO03	dump	After positioning is complete, before returning control to IDCIOPO.
IOPR	IDCIO01	PUTREP	dump	After the PUT for update.
IOPT	IDCIO01	IDCIOPT	trace	Start of routine that writes data records to an opened data set.
IOP1	IDCIO05	BLDCCWP1	trace	Start of procedure that builds CCWs to write count, key, and data.
IOP2	IDCIO05	BLDCCWP2	trace	Start of procedure that builds CCWs to write key and data.
IORJ	IDCIO05	READJFCB	trace	Start of procedure that issues RDJFCB and prints error messages.
IORP	IDCIO02	BUILDRPL	dump	After RPL is built, at end of procedure.
IORT	IDCIO05	RETRY	trace	Start of procedure that issues a message that password is invalid.
IOR1	IDCIO05	OPENR	dump	Start of procedure that opens data sets, VTOCs, and staging packs for REPAIRV.
IOR2	IDCIO05	READCNT	dump	Start of procedure that reads the count field of each record on a track.
IOR3	IDCIO05	READKD	dump	Start of procedure that reads the count, key, and data fields of each record on a track.
IOR4	IDCIO05	SPACCR	dump	Start of procedure that spaces over a defective count field on a track.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
IOR5	IDCIO05	SETCCW	dump	Start of procedure that builds the CCW channel program to read the count field of each record on a track.
IOR6	IDCIO05	SETKDCCW	dump	Start of procedure that builds the CCW channel program to read the count, key, and data fields of each record on a track.
IOR7	IDCIO05	ADJCCW	dump	Start of procedure that adjusts a CCW channel program.
IOSN	IDCIO05	SYNAD	trace	Start of procedure that issues SYNADAF, UPRINT, and SYNADRLS macros.
IOSO	IDCIO03	PTISDS	trace	End of routine that positions to an ISAM or BSAM record.
IOSP	IDCIO05	SPACCR	trace	Start of procedure that spaces over a defective count field on a track.
IOSR	IDCIO03	STOWRTN	trace	Before STOW macro is issued.
IOST	IDCIO01	IDCIOST	trace	Entry to the STOW routine.
IOSY	IDCIO05	SYNAD	dump	Start of SYNAD routine that issues SYNADAF, SYNADRLS, and UPRINT macros.
IOS2	IDCIO01	GETNONVS	trace	Start of SYNAD routine for non-VSAM read error.
IOS4	IDCIO01	PUTNONVS	trace	Start of SYNAD routine for non-VSAM put error.
IOTM	IDCIO01	IDCIOTM	trace	Start of termination routine that closes all data sets and frees space.
IOUO	IDCIO01	IDCIOSI	trace	Entry to UIOINFO entry processing.
IOVE	IDCIO01	GETVSAM	trace	Start of end-of-file exit routine for a VSAM file.
IOVG	IDCIO01	GETVSAM	dump	End of procedure that gets a record or control interval from a VSAM data set.
			trace	Start of procedure that gets a record or control interval from a VSAM data set.
IOVH	IDCIO05	OPNVTH	trace	Start of procedure that performs a pseudo open on a VTOC header or a VSAM data set.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
IOVP	IDCIO01	PUTVSAM	dump	End of procedure that writes a VSAM record.
			trace	Start of procedure that writes a VSAM record.
IOVR	IDCIO01	VSAMERR	dump	After detection of a VSAM I/O error.
IOVT	IDCIO03	PTAMDS	trace	Start of procedure that positions to a VSAM record or control interval.
IOVY	IDCIO01	IDCIOVY	dump	After VERIFY macro is issued.
			trace	After VERIFY macro is issued.
IOWB	IDCIO05	WRITEREC	trace	Before calling UEXCP to perform I/O.
IOWR	IDCIO05	WRITEREC	trace	Beginning of procedure which writes a record on a track.
IOW1	IDCIO03	STOWRTN	trace	After STOW macro is issued.
IOXC	IDCIO05	ISSUEXCP	trace	Start of procedure that issues EXCP and WAIT macros and tests for errors.
IO0O	IDCIO03	DSINFO	dump	After OBTAIN macro is issued.
IO00	IDCIO03	DSINFO	dump	After RDJFCB macro is issued.
IO01	IDCIO03	DSINFO	dump	After DEVTYPE macro is issued.
IO02	IDCIO03	DSINFO	dump	After formatting work area.
IO05	IDCIO05	IDCIO05	trace	Start of IDCIO05 module.
IO1C	IDCIO02	CLOSERTN	dump	Before CLOSE macro is issued.
IO1O	IDCIO02	OPENRTN	dump	Before OPEN macro is issued.
IO20	IDCIO02	OPENRTN	dump	After OPEN macro is issued.
IO21	IDCIO02	OPENRTN	dump	At completion of all UOPEN processing.
IO2C	IDCIO02	CLOSERTN	dump	At completion of all UCLOSE processing.
IO30	IDCIO02	OPENRTN	dump	After OPEN TYPE=J macro is issued.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
LA01	IDCLA01	IDCLA01	both	Start of LISTDATA FSR.
LAG1	IDCLA01	IDCLA01	both	Before link to IDCSS01.
LAR1	IDCLA01	IDCLA01	both	After return from IDCSS01.
LARS	IDCLA01	LISTSTAT	trace	Start of LISTSTAT procedure.
LARC	IDCLA01	LISTCNTS	trace	Start of LISTCNTS procedure.
LALG	IDCLA01	LEGEND	trace	Start of LEGEND procedure.
LCAL	IDCLC02	LOCPROC	dump	After calling the catalog to locate an entry.
LCAP	IDCLC02	AUPROC	dump	Start of procedure that lists information about alias, non-VSAM, user catalog, or generation data group.
LCAS	IDCLC02	ASLPROC	trace	Start of procedure that locates and prints alias names.
LCAU	IDCLC02	AUPROC	trace	Start of procedure that formats catalog fields for a non-VSAM, alias, BDG, or user catalog entry.
LCCD	IDCLC02	CDIPROC	dump	Start of procedure that lists information about cluster, AIX, pagespace, data, index, or path.
LCBL	IDCLC02	LOCPROC	dump	Before calling the catalog to locate an entry.
LCCK	IDCLC01	CKDTPROC	trace	Start of procedure that tests the creation and expiration LISTCAT options against the catalog entry's creation and expiration date fields.
LCCL	IDCLC02	CDIPROC	trace	Start of procedure that formats catalog fields for a cluster, AIX, data, index, or path entry.
LCDT	IDCLC01	DATEPROC	trace	Start of procedure that calculates the creation and expiration LISTCAT options to be used in the testing of the catalog entry's creation and expiration date fields.
LCEN	IDCLC01	ENTPROC	trace	Before retrieving each entry in a list of entries.
LCER	IDCLC02	ERRPROC	trace	Start of procedure that issues messages.
LCFP	IDCLC02	FPLPROC	trace	Start of procedure that reinitializes CTGFLs for each locate request.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
LCLA	IDCLC02	ANLTPROC	trace	Start of procedure that lists catalog entry associations.
LCLO	IDCLC02	LOCPROC	dump	Ready to do a catalog locate for information.
LCIN	IDCLC01	INITPROC	trace	Start of procedure that initializes the catalog parameter list and work areas.
LCLT	IDCLC02	LISTPROC	trace	Start of procedure that prints catalog data.
LCMG	IDCLC02	ERRPROC	dump	Before UPRINT macro is issued to print a message.
LCNX	IDCLC01	GNXTPROC	trace	Before retrieving each entry when processing a full catalog.
LCOJ	IDCLC01	ENTPROC	dump	After preparing to locate information about a name.
LCRA	IDCLC01	RTEPROC	dump	Processing catalog information for associations of a cluster, AIX, pagespace, or generation data group.
LCRP	IDCLC01	RTEPROC	dump	Start of procedure that determines which procedure will list the catalog entry.
LCRT	IDCLC01	RTEPROC	trace	Start of procedure that directs the retrieved entry to the proper formatting procedure.
LCR2	IDCLC01	RTEPROC	trace	Start of section of procedure that processes associations of a cluster, or AIX.
LCSA	IDCLC02	ANSVPROC	trace	Start of procedure that retrieves the list of types and CI numbers.
LCSH	IDCLC02	SHORTLST	trace	Start of procedure that formats an abbreviated list of catalog entry names from the UCIR workarea.
LCTM	IDCLC01	TIMEPROC	trace	Start of procedure that converts the time of day to a packed decimal format.
LCTP	IDCLC02	LISTPROC	dump	Before UPRINT macro is issued to print catalog data.
LCVL	IDCLC02	VPROC	trace	Start of procedure that formats catalog fields of a space entry.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
LCVO	IDCLC02	VOLLIST	trace	Start of procedure that formats an abbreviated list of catalog entrynames and volume serials from the UCIR workarea.
LCWA	IDCLC02	LOCPROC	dump	After calling the catalog to locate an entry.
LC02	IDCLC02	IDCLC02	dump	When IDCLC02 is called the first time to establish addressability.
LC98	IDCLC02	FREESTG	dump	End of LISTCAT FSR, before freeing storage in IDCLC02.
LC99	IDCLC01	IDCLC01	dump	End of LISTCAT FSR, before freeing storage in IDCLC01.
LRAA	IDCLR01	AATOPLR	dump	Entry point for IDCLR01.
LRAD	IDCLR01	ADDASOC	dump	Start of procedure that adds an association to the association table.
LRBL	IDCLR01	BLDVEXT	dump	Start of procedure that builds virtual extension table.
LRBU	IDCLR01	BUFSHUF	dump	Start of procedure that moves a record to its "home" buffer.
LRCA	IDCLR01	CATOPEN	dump	Start of procedure that prepares to open the catalog.
LRCK	IDCLR01	CKEYRNG	dump	Start of procedure that checks for keyrange.
LRCR	IDCLR01	CRAOPEN	dump	Start of procedure that opens the CRA.
LRCT	IDCLR01	CTTBLD	dump	Start of procedure that builds CI translate table.
LRC1	IDCLR01	CLEANUP	dump	Start of procedure that cleans up before exit.
LRC2	IDCLR01	CLENCRA	dump	Start of procedure that closes the CRA and prints completion message.
LRDO	IDCLR01	DOOTHR	dump	Start of procedure that controls printing non-VSAM information.
LRDV	IDCLR01	DOVSAM	dump	Start of procedure that controls printing VSAM information.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
LRER	IDCLR01	ERROR	dump	Start of procedure that handles errors.
LRGE	IDCLR01	GETPRT	dump	Start of procedure that gets and print records.
LRIA	IDCLR01	INTASOC	dump	Start of procedure that initializes association tables.
LRIN	IDCLR01	INITLZE	dump	Start of procedure that initializes the FSR.
LRIS	IDCLR01	INTSORT	dump	Start of procedure that initializes the sort table.
LRIV	IDCLR01	INTVEXT	dump	Start of procedure that initializes the virtual extension table.
LRME	IDCLR01	MEMSORT	dump	Start of procedure that sorts the entries in sort table.
LRPA	IDCLR01	PRTAAXV	dump	Start of procedure that prints associated AIXs and volumes.
LRPC	IDCLR01	PRTCMP	dump	Start of procedure that prints and compares information.
LRPD	IDCLR01	PRTDMP	dump	Start of procedure that prints dump if specified.
LRPE	IDCLR01	PRTDMPC	dump	Start of procedure that prints dump of catalog record and underscores miscompared records.
LRPF	IDCLR01	PRTFIFO	dump	Start of procedure that prints CRA in order of CI number.
LRPJ	IDCLR01	PRTOJAL	dump	Start of procedure that prints object aliases.
LRPK	IDCLR01	PRTOJVL	dump	Start of procedure that prints an object's volumes.
LRPM	IDCLR01	PRTMCWD	dump	Start of procedure that prints miscompared words.
LRPO	IDCLR01	PRTOTHR	dump	Start of procedure that prints non-VSAM objects.
LRPT	IDCLR01	PRTTIME	dump	Start of procedure that prints timestamps.
LRPV	IDCLR01	PRTVSAM	dump	Start of procedure that prints VSAM structures.
LRPW	IDCLR01	PRTVOL	dump	Start of procedure that prints volume records.
LRSM	IDCLR01	SUMIT	dump	Start of procedure that prints number of entries processed.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
LRTC	IDCLR01	TCICTCR	dump	Start of procedure that translates the catalog CI to the CRA.
LRVE	IDCLR01	VERTEXT	dump	Start of procedure that handles vertical extension records.
LRZY	IDCLR01	ERROR	dump	After error message has been printed.
LRZZ	IDCLR01	ERROR	dump	After error that forced an ABORT of this execution.
LR02	IDCLR02	IDCLR02	dump	Entry point for module that gets a record for recovery field management routine.
MPBF	IDCMP01	FPLPROC	trace	Start of procedure that constructs a CTGFL.
MPBG	IDCMP01	IDCMP01	trace	Start of IMPORT FSR.
MPCO	IDCMP01	COPYICF	trace	Start of procedure that copies an ICF catalog from the portability data set.
MPCP	IDCMP01	CLUSPROC	trace	Start of procedure that imports a cluster, alternate index, or ICF user catalog.
MPCT	IDCMP01	CLUSPROC	trace	Before processing information from the portable data set to define a cluster, alternate index, or ICF user catalog.
MPDA	IDCMP01	DALCPROC	trace	Start of procedure that dynamically allocates volumes.
MPDC	IDCMP01	DELTPROC	dump	After the first UCATLG.
MPDD	IDCMP01	DELTPROC	dump	After the second UCATLG.
MPDL	IDCMP01	DELTPROC	trace	Entry to DELTPROC.
MPDN	IDCMP01	DUPNPROC	trace	Start of procedure to process a duplicate entry found in the catalog.
MPFN	IDCMP01	IDCMP01	dump	End of IMPORT FSR, prior to closing data sets.
MPFV	IDCMP01	FVTPROC	trace	Start of procedure that constructs a CTGFV and CTGFLs.
MPLV	IDCMP01	LVLRPROC	trace	Start of procedure that constructs CTGFLs for device and volume information.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
MPMG	IDCMP01	MSGPROC	trace	Start of procedure that issues messages.
MPOP	IDCMP01	OPENPROC	trace	Start of procedure that opens either the portable data set or the newly defined data set.
MPPS	IDCMP01	BPASPROC	trace	Start of procedure that constructs the PASSWALL CTGFL for protection information.
MPPT	IDCMP01	CLUSPROC	trace	After imported cluster, alternate index, or ICF user catalog has been successfully defined and the contents of the portable data set copied into the new cluster, alternate index, or ICF user catalog.
MPSP	IDCMP01	CTLGPROC	trace	Start of procedure that calls the catalog to locate, alter, or define an entry.
MPUC	IDCMP01	CNCTPROC	trace	Start of procedure that connects a user catalog.
MPUQ	IDCMP01	IUNIQPRC	trace	After a data or index has been found to be unique.
MPZZ	IDCMP01	CTLGPROC	dump	Before and after calling the catalog to locate, alter, or define an entry.
PMGP	IDCPM01	GRPHPARM	trace	Start of procedure that processes the graphics option.
PMMG	IDCPM01	MARGPARM	trace	Start of procedure that processes the margins option.
PMTP	IDCPM01	TESTPARM	trace	Start of procedure that initializes the TEST option.
PMTS	IDCPM01	TESTSAVE	trace	Start of procedure that initializes the test option data area.
PR01	IDCPR01	IDCPR01	dump	End of PRINT FSR.
PR11	IDCPR01	IDCPR01	trace	Start of PRINT FSR.
PR18	IDCPR01	IDCPR01	trace	Before termination processing.
PR21	IDCPR01	TEXTPSET	trace	Start of procedure that sets up the text processor interface.
PR31	IDCPR01	DELIMSET	trace	Start of procedure that establishes the beginning and ending delimiters of the data set to be printed.
RCAC	IDCSA07	GETENT	dump	After calls to the catalog to locate an entry.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RCAC	IDCSA07	UPDATENT	dump	After calls to the catalog to recatalog an entry.
RCBG	IDCSA07	IDCSARC	dump	At entry point to IDCSARC procedure.
RCEX	IDCSA07	IDCSARC	trace	Start of IDCSARC procedure that calls GETENT.
RCGE	IDCSA07	GETENT	trace	Start of GETENT procedure that initializes the CAMLST parameter list.
RCND	IDCSA07	IDCSARC	dump	At return to caller.
RCTE	IDCSA07	TESTENT	trace	Start of TESTENT procedure that tests whether there is a need to recatalog.
RCUE	IDCSA07	UPDATENT	trace	Start of UPDATENT procedure that updates the catalog entry.
RC01	IDCRC02	IDCRC02	trace	Start of main procedure.
RC02	IDCRC02	IDCRC02	dump	Start of main procedure.
RC03	IDCRC02	IDCRC02	trace	Return in main procedure from procedures which processed catalog information for objects. Start of termination processing.
RC04	IDCRC02	IDCRC02	dump	Return in main procedure from procedures which processed catalog information for objects. Start of termination processing.
RC05	IDCRC02	CLUSPROC	trace	Start of procedure which processes VSAM objects.
RC06	IDCRC02	CLUSPROC	dump	Start of procedure which processes VSAM objects.
RC07	IDCRC02	CLUSPROC	trace	Before routine which calls LOCPROC for data and index processing.
RC09	IDCRC02	CLUSPROC	trace	Start build of timestamp information for portability data set.
RC11	IDCRC02	CLUSPROC	trace	Start of processing for path associations for VSAM objects.
RC13	IDCRC02	LOCPROC	trace	Start of procedure which builds CPL and FPL's for catalog locate functions.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RC15	IDCRC02	CTLGPROC	trace	Start of procedure which issues catalog locates.
RC16	IDCRC02	CTLGPROC	dump	Start of procedure which issues catalog locates.
RC17	IDCRC02	OPENPROC	trace	Start of procedure to open input and output data sets.
RC19	IDCRC02	PUTPROC	trace	Start of procedure which writes control records to the output data set.
RC21	IDCRC02	RECPROC	trace	Start of procedure which copies the data from the input data set to the output data set.
RC23	IDCRC02	MVDAPROC	trace	Start of procedure which moves control record information in core and clears work areas in storage.
RC25	IDCRC02	CONTRBL	trace	Start of procedure which builds control record information.
RC27	IDCRC02	NVSMPROC	trace	Start of procedure which processes non-VSAM objects.
RC28	IDCRC02	NVSMPROC	dump	Start of procedure which processes non-VSAM objects not associated to GDGs.
RC29	IDCRC02	NVSMPROC	trace	Before timestamp processing for non-VSAM objects not associated to GDGs.
RC31	IDCRC02	SAVEPROC	trace	Start of procedure which saves control record information and writes control information to the output data set.
RC33	IDCRC02	ALSPROC	trace	Start of procedure which processes catalog information for alias associations for non-VSAM objects.
RC35	IDCRC02	GDGPROC	trace	Start of procedure which processes catalog information for GDGs.
RC36	IDCRC02	GDGPROC	dump	Start of procedure which processes catalog information for GDGs.
RC37	IDCRC02	GDGPROC	trace	Before build of timestamp information for GDG's.
RC39	IDCRC02	ASOCPROC	trace	Start of procedure which processes catalog information for non-VSAM objects associated with GDGs.
RC40	IDCRC02	ASOCPROC	dump	Start of procedure which processes catalog information for non-VSAM objects associated with GDGs.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RC42	IDCRC02	PRNTPROC	trace	Start of procedure which prints error messages for associations.
RC79	IDCRC01	TERM	both	Before special processing to terminate request (closing output data set).
RC80	IDCRC01	INIT	both	Before initializing to begin processing.
RC81	IDCRC01	BUILDCRV	both	Before building the CRV.
RC82	IDCRC01	EXPORTDR	both	Before looping down name chain to call IDCRC02 to export data sets.
RC83	IDCRC01	SYNCH	both	Before scanning the name chain for a CRA to check it.
RC84	IDCRC01	OBJVOLCK	both	Before checking synchronization of an entry across multiple volumes.
RC85	IDCRC01	DUPNAMCK	both	Before checking the name chain for duplicates.
RC86	IDCRC01	BUILDNAM	both	Before constructing a block for the name chain.
RC87	IDCRC01	COMPNAME	both	Before compressing a name for the name list.
RC88	IDCRC01	SUBSP	both	Before allocating space for the name chain.
RC89	IDCRC01	MESSAGE	both	Before printing any message from IDCRC01.
RC90	IDCRC01	EXTRACT	both	Before using internal field management to get information from CRA.
RC91	IDCRC01	OPENCRA	both	Before opening or closing or CRA and doing all other work (e.g., build CTT).
RC92	IDCRC01	OPEN	both	Before the opening of the CRA.
RC93	IDCRC01	CKCATNM	both	Before checking owning catalog name of CRA being opened.
RC94	IDCRC01	TIMESTMP	both	Before obtaining format 4 timestamp for CRA being opened.
RC95	IDCRC01	SCANCRA	both	Before scanning CRA to build the CTT table.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RC96	IDCRC01	ERRCK	both	After opening a CRA.
RC97	IDCRC01	NAMETABL	both	Before marking or adding a name to the name chain.
RC98	IDCRC01	DIRECT	both	Before obtaining the directory for a volume.
RC99	IDCRC01	CKNAMES	both	Before gathering information on name in name list from CRA.
RIBT	IDCRI01	BYPASTRM	dump	Start of procedure that bypasses the remainder of the current modal or null command.
RICV	IDCRI01	CONVERT	dump	Start of procedure that converts a constant from EBCDIC to binary or hexadecimal.
RIDC	IDCRI01	DSPLCALC	dump	Start of procedure that calculates the position within a secondary FDT vector in which to place an FDT pointer.
RIDF	IDCRI01	DEFAULTS	dump	Start of procedure that adds default parameters to the FDT.
RIEX	IDCRI01	IDCRI01	dump	Start of reader/interpreter module.
RIE1	IDCRI01	ERROR1	dump	Start of procedure that issues a message without inserted text.
RIE2	IDCRI01	ERROR2	dump	Start of procedure that issues a message with inserted text.
RIGN	IDCRI01	GETNEXT	dump	Start of procedure that scans the input command.
RIGQ	IDCRI01	GETQUOTD	dump	Start of procedure that scans a quoted constant.
RIGR	IDCRI01	GETRECRD	dump	Start of procedure that obtains the next input record.
RIID	IDCRI01	DSIDCHK	trace	Check restrictions on a data set name and place in FDT.
RIIR	IDCRI01	INREPEAT	dump	Start of procedure that scans a repeated parameter set.
RIMC	IDCRI01	MORSPACE	dump	Start of procedure that allocates more FDT space for a list of constants.
RIME	IDCRI01	MODLELSE	dump	Start of procedure that scans an ELSE modal command.
RIMI	IDCRI01	MODALIF	dump	Start of procedure that scans an IF modal command.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RIMS	IDCRI01	MODALSET	dump	Start of procedure that scans a SET modal command.
RINN	IDCRI01	NEEDNOTS	dump	Start of procedure that checks the input command for conflicting or missing parameters.
RINS	IDCRI01	NAMESCAN	dump	Start of procedure that checks data set names.
RIPC	IDCRI01	PACKCVB	dump	Start of procedure that converts a decimal constant into a binary fullword.
RIPP	IDCRI01	POSPARM	dump	Start of procedure that scans a positional parameter.
RISC	IDCRI01	SCANCMD	dump	Start of procedure that scans the input command parameters.
RISD	IDCRI02	IDCRI02	dump	Start of module that prepares to scan a command parameter set.
RISE	IDCRI01	SCANENDS	dump	Start of procedure that checks the input record for a continuation delimiter and determines the scanning limits of the record.
RISF	IDCRI01	SETFLAG	dump	Start of procedure that notes the occurrence of a parameter in the FDT.
RISK	IDCRI01	SKIPCMD	dump	Start of procedure that bypasses the remainder of a function command.
RIST	IDCRI01	SETDFLT	dump	Start of procedure that puts parameter defaults in the FDT.
RITM	IDCRI03	IDCRI03	dump	Start of module that performs command termination functions.
RI01	IDCRI01	SCANCMD	trace	Start of scanning for a parameter.
RI02	IDCRI01	SCANCMD	trace	Scanning a first-level parameter.
RI03	IDCRI01	SCANCMD	trace	Scanning a subparameter.
RI04	IDCRI01	GETNEXT	trace	Modal command other than ELSE within an IF.
R105	IDCRI01	GETNEXT	trace	Found a functional command.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RI09	IDCRI01	KWDPARM	trace	Found a keyword subparameter.
RI10	IDCRI04	RISETUP	trace	Start of procedure that prepares to process a command.
RI11	IDCRI01	GETDATA	trace	Start of extracting a scalar value.
RI12	IDCRI01	GETDATA	trace	Extract a character string.
RI12	IDCRI04	MAINSCAN	trace	Start of procedure that processes nonrepeated parameters.
RI14	IDCRI04	SUBSCAN	trace	Start of procedure that processes repeated parameters.
RI16	IDCRI02	IDCRI02	trace	Prior to loading the command descriptor.
RI16	IDCRI04	TRNSLATE	trace	Start of procedure that translates the PDL into the FDT.
RI17	IDCRI02	IDCRI02	trace	Beginning of the code sequence to build the PARMINFO table.
RI18	IDCRI04	FINDPDE	trace	Start of procedure that finds the PDE offset into the PDL for the current parameter.
RI20	IDCRI04	REPLIST	trace	Start of procedure that saves repeated parameter information.
RI22	IDCRI04	BUILDFDT	trace	Start of procedure that builds FDT data substructure.
RI24	IDCRI01	CONVERT	trace	Start converting a binary number.
RI24	IDCRI04	GETSPACE	trace	Start of procedure that gets space for the FDT data structures.
RI26	IDCRI04	NEWPARM	trace	Start of procedure that gets a new value for the parameter.
RI27	IDCRI01	CONVERT	trace	Start converting a hexadecimal number.
RI28	IDCRI04	TOOMANY	trace	Start of procedure that handles error of excessive subparameters or data.
RI30	IDCRI01	CONVERT	trace	Change converted digits into a binary fullword.
RI30	IDCRI04	NOTPARMS	trace	Start of procedure that checks for conflicting parameters.
RI32	IDCRI04	RESOLVE	trace	Start of procedure that resolves conflicting parameters.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RI34	IDCRI04	DEFAULTS	trace	Start of procedure that selects parameter defaults.
RI35	IDCRI01	INREPEAT	trace	Loop to reset parameter occurrence flags for possible parameters in the sublist.
RI36	IDCRI01	INREPEAT	trace	End of last repeated sublist.
RI36	IDCRI04	SETDFLT	trace	Start of procedure that puts parameter defaults in the FDT.
RI38	IDCRI04	NEEDPRMS	trace	Start of procedure that checks for missing parameters.
RI40	IDCRI04	ADDPARM	trace	Start of procedure that prompts for missing parameters.
RI42	IDCRI04	MSGKWD	trace	Start of procedure that finds a keyword to put in an error message.
RI44	IDCRI01	SETDFLT	trace	Found that at default is allowable; ready to put in FDT.
RI44	IDCRI04	FAILSPAC	trace	Start of procedure that prints "no space" error message.
RI45	IDCRI01	SETDFLT	trace	Move a defaulted unquoted constant to FDT.
RI46	IDCRI04	RITERM	trace	Start of procedure that terminates the reader/interpreter for TSO.
RI49	IDCRI01	NXTFIELD	trace	Extract a field from input (verb, keyword, or scalar).
RI50	IDCRI01	NXTFIELD	trace	Extract a keyword field.
RI51	IDCRI01	NXTFIELD	trace	Extract a quoted scalar.
RI56	IDCRI01	NEXTCHAR	trace	End-of-file already found in input.
R157	IDCRI01	NEXTCHAR	trace	Extract first character of a new command.
RI59	IDCRI01	NEXTCHAR	trace	End-of-file found while looking for next character.
RI60	IDCRI01	SCANENDS	trace	Skip leading blanks and comments if preceding record indicated continuation.
RI61	IDCRI01	SCANENDS	trace	Bypass a leading comment.
RI62	IDCRI01	SCANENDS	trace	Bypass leading blanks.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RI66	IDCRI01	DSPLCALC	trace	Calculate displacement into the FDT for a parameter in a first-level repeated parameter list.
RI99	IDCRI03	IDCRI03	trace	End of IDCRI03.
RMAL	IDCRM01	ALISPROC	trace	Entry to ALISPROC.
RMAT	IDCRM01	ALTRPROC	trace	Entry to ALTRPROC.
RMBF	IDCRM01	BFPLPROC	trace	Entry to BFPLPROC.
RMBG	IDCRM01	IDCRM01	trace	Entry to IDCRM01.
RMCE	IDCRM01	CLUSPROC	trace	Exit from CLUSPROC.
RMCL	IDCRM01	CPLPROC	dump	After the CPL has been built.
RMCP	IDCRM01	CLUSPROC	trace	Entry to CLUSPROC.
RMCT	IDCRM01	CLUSPROC	trace	Begin reading of cluster or alternate index information from the portable data set.
RMDA	IDCRM01	DALCPROC	trace	Entry to DALCPROC.
RMDC	IDCRM01	DELTPROC	dump	After the first UCTALG in DELTPROC.
RMDD	IDCRM01	DELTPROC	dump	After the second UCATLG in DELTPROC.
RMDL	IDCRM01	DELTPROC	trace	Entry to DELTPROC.
RMDG	IDCRM01	GDGPROC	trace	A duplicate GDG entry has been found.
RMDN	IDCRM01	NVSMPROC	trace	Duplicate non-VSAM entry found.
RMDU	IDCRM01	UCATPROC	trace	Duplicate user catalog found.
RMDV	IDCRM01	CLUSPROC	trace	A duplicate VSAM entry has been found.
RMEL	IDCRM01	IDCRM01	trace	End of the loop for importing objects.
RMFN	IDCRM01	IDCRM01	dump	Termination of IDCRM01.
RMFV	IDCRM01	FVTPROC	trace	Entry to FVTPROC.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RMGD	IDCRM01	GDGPROC	trace	Entry to GDGPROC.
RMLV	IDCRM01	LVLRPROC	trace	Entry to LVLPROC.
RMOP	IDCRM01	OPENPROC	trace	Entry to OPENPROC.
RMNF	IDCRM01	NFVTPROC	trace	Entry to NFVTPROC.
RMNV	IDCRM01	NVSMPROC	trace	Entry to NVSMPROC.
RMPL	IDCRM01	CPLPROC	trace	Entry to CPLPROC.
RMPS	IDCRM01	BPASPROC	trace	Entry to BPASPROC.
RMPT	IDCRM01	CLUSPROC	trace	Beginning of path definition sequence.
RMRG	IDCRM01	RANGPRC	trace	Entry to RANGPROC.
RMSP	IDCRM01	CTLGPROC	trace	Entry to CTLGPROC.
RMUC	IDCRM01	UCATPROC	trace	Entry to UCATPROC.
RMUQ	IDCRM01	IUNIQPRC	trace	A unique data or index component has been detected.
RMZZ	IDCRM01	CTLGPROC	dump	Before and after the UCATLG in CTLGPROC.
RPAL	IDCRP01	ALTERICF	trace	Start of procedure that updates the ICF catalog pointer in all VVRs.
RPCI	IDCRP01	CNVRTCI	dump	On exit from procedure that translates control interval numbers on the backup catalog.
RPCP	IDCRP01	COPYICF	trace	Start of procedure that copies ICF catalog records.
RPDI	IDCRP01	CATRELOD	dump	At the end of all reload error checking before any updates have been done to the target catalog.
RPGA	IDCRP01	GETALL	trace	Start of procedure that moves all entries from the source to the target ICF catalog.
RPGN	IDCRP01	GETNAM	trace	Start of procedure that moves specific entries from the source to the target ICF catalog.
RPIO	IDCRP01	DUMPIT	dump	After read or write to backup or target catalog.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RPMI	IDCRP01	MERGEICF	trace	Start of procedure that performs the merge of an ICF catalog.
RPMO	IDCRP01	MOVEOBJ	trace	Start of procedure that moves a base object and all its associations from the source to the target ICF catalog.
RPRI	IDCRP01	REPROICF	trace	Start of procedure that performs ICF catalog REPRO functions.
RPRV	IDCRP01	RECOVCAT	dump	After initialization of locate parameter list to determine catalog recoverable attribute.
RPTU	IDCRP01	TRUENAME	dump	On exit from procedure, having built truename range table.
RPT1	IDCRP01	CATRELOD	trace	Start of procedure that performs catalog reload.
RPT2	IDCRP01	TRUENAME	trace	Start of procedure that the RBA boundaries of the backup truename ranges.
RPT3	IDCRP01	CATRANS	trace	On entry to procedure that locates control interval numbers to be translated.
RPT4	IDCRP01	CNVRTCI	trace	On entry to procedure that converts control interval numbers from the backup catalog.
RPT5	IDCRP01	CATCOMP	trace	On entry to procedure that compares truename records.
RPT6	IDCRP01	VERIFYC	trace	On entry to procedure that issues VERIFY against a catalog.
RP01	IDCRP01	IDCRP01	dump	End of REPRO FSR.
RP12	IDCRP01	IDCRP01	trace	After all data sets have not been opened successfully.
RP13	IDCRP01	IDCRP01	trace	Start of loop that copies the data set by issuing UGET and UPUT macros.
RP18	IDCRP01	IDCRP01	trace	After all records have been copied to output data set.
RP21	IDCRP01	DELIMSET	trace	Start of procedure that sets up the beginning and ending delimiters of the input data set.
RP25	IDCRP01	CRYPTSET	trace	Start of procedure that calls UCRYPT to write, or read and verify, the data set header.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RSAD	IDCRS05	ADDUPCR	trace	Upon entry to routine which updates the CRA for a particular period.
RSAE	IDCRS01	AERROR	trace	On entry to routine that exists if enough storage is not available to establish automatic storage required for RESETCAT modules.
RSAS	IDCRS02	ASSOC	trace	On entry to routine that initiates association checking.
RSAT	IDCRS05	ADDTN	trace	On entry to routine that adds a true name to the catalog.
RSA1	IDCRS02	ASSOC	dump	At end of procedure that initiates association checking.
RSA2	IDCRS05	ADDUPCR	dump	At end of procedure that prepares for update CRA processing.
RSBR	IDCRS05	BLDRLST	trace	On entry to routine that adds an entry to the reset volume table.
RSBV	IDCRS05	BLDVLST	trace	On entry to routine that adds an entry to the volume serial table.
RSB1	IDCRS05	BLDVLST	dump	End of procedure that adds an entry to the volume serial table.
RSB2	IDCRS05	BLDRLST	dump	At end of procedure that adds an entry to the reset volume table.
RSCA	IDCRS02	CINALTER	trace	On entry to routine that alters control interval numbers in catalog records.
RSCC	IDCRS07	CNVTCCHH	trace	On entry to routine that converts CCHH to TTnn.
RSCD	IDCRS06	CRADMNT	trace	On entry to routine that requests deallocation of a CRA volume.
RSCE	IDCRS07	CATEOV	trace	On entry to routine that extends the catalog.
RSCH	IDCRS03	CHKDSDIR	trace	On entry to routine that checks a data set directory entry against a DATA or INDEX component.
RSCI	IDCRS01	CATINIT	trace	On entry to routine that initializes RESETCAT's description of the catalog.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RSCK	IDCRS05	CKERR	trace	On entry to routine that prints a message if one is associated with the error message given.
RSCL	IDCRS01	CLEANUP	trace	On entry to routine that ensures all RESETCAT resources are freed.
RSCM	IDCRS06	CRAMNT	trace	On entry to routine that requests CRA deallocation.
RSCO	IDCRS01	COPYCAT	trace	On entry to procedure that copies the catalog to the workfile.
RSCR	IDCRS05	CRAUPCHN	trace	On entry to routine that adds a workfile record to a specific "update CRA" chain.
RSCU	IDCRS03	CATRCDSU	trace	On entry to routine that establishes base record offsets for catalog low key range records.
RSC1	IDCRS01	CATINIT	dump	End of procedure that builds CIN to RRN table.
RSC2	IDCRS01	COPYCAT	dump	End of procedure that copies the catalog to the workfile.
RSC3	IDCRS01	CLEANUP	dump	Before freeing the resources used by RESETCAT.
RSC4	IDCRS05	CKERR	dump	Before RESETCAT FSR is terminated due to an error.
RSC5	IDCRS06	CRAMNT	dump	End of procedure that requests CRA allocation.
RSC6	IDCRS06	CRADMNT	dump	End of procedure that requests CRA deallocation.
RSC7	IDCRS07	CATEOV	dump	At conclusion of routine that extends the catalog.
RSDC	IDCRS06	DSCLOSE	trace	On entry to procedure that closes a VSAM data set.
RSDE	IDCRS04	DELGO	trace	On entry to routine that deletes a group occurrence.
RSDO	IDCRS06	DSOPEN	trace	On entry to procedure that opens VSAM data sets.
RSDT	IDCRS05	DELTN	trace	On entry to procedure that deletes a true name from the catalog.
RSD1	IDCRS06	DSOPEN	dump	End of procedure that opens a VSAM data set.
RSD2	IDCRS06	DSCLOSE	dump	End of procedure that closes a VSAM data set.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RSD3	IDCRS04	DELGO	dump	End of procedure that deletes a group occurrence.
RSEN	IDCRS05	ENTNMCK	trace	On entry to routine that determines if a catalog record has a valid entry name.
RSES	IDCRS01	ENSURECI	trace	On entry to routine that ensures there are enough free CIs for reassignment.
RSE1	IDCRS05	ENTNMCK	dump	End of procedure that determines if a record has a true name.
RSE2	IDCRS01	ENSURECI	dump	A start of procedure prior to ensuring enough free CIs.
RSFI	IDCRS04	FIND	trace	On entry to routine that locates requested information from a set of catalog records.
RSFR	IDCRS07	FRCRCCR	trace	On entry to routine that forces reading of the CCR by catalog management.
RSF1	IDCRS04	FIND	dump	End of routine that finds one or all group occurrences.
RSGE	IDCRS05	GENNAME	trace	On entry to routine that generates a true name.
RSGF	IDCRS03	GETFIT	trace	On entry to routine that gets a free entry in tables for ASSOC.
RSGN	IDCRS03	GETNEXTE	trace	On entry to routine that translates an index into a table into a virtual address.
RSGT	IDCRS03	GETTAB	trace	On entry to routine that gets and initializes a table for ASSOC.
RSGV	IDCRS03	GETVIA	trace	On entry to routine that gets a record by control interval number via a specific CRA.
RSG1	IDCRS03	GETVIA	dump	End of procedure that locates records in the workfile.
RSIN	IDCRS01	INIT	trace	On entry to routine which performs the main initializations for RESETCAT.
RSI1	IDCRS01	INIT	dump	End of procedure that initializes data areas and obtains resources.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RSME	IDCRS01	MERGCRA	trace	On entry to routine that merges each reset CRA into the workfile.
RSMO	IDCRS04	MODGO	trace	On entry to procedure that modifies a group occurrence.
RSMU	IDCRS03	MARKUNUS	trace	On entry to routine that marks a volume group occurrence (VGO) unusable.
RSM1	IDCRS01	MERGCRA	dump	End of procedure that merges and resets CRA into the workfile.
RSM2	IDCRS04	MODGO	dump	End of procedure that modifies a group occurrence.
RSPC	IDCRS02	PROCTYPE	trace	On entry to routine that scans a catalog record for CINs.
RSPI	IDCRS02	PROCCI	trace	On entry to routine that ensures a CIN is in the list of CINs for records being processed.
RSPR	IDCRS01	PROCCRA	trace	On entry to routine that processes the records of the current CRA.
RSPV	IDCRS03	PROCVOL	trace	On entry to routine that resolves space conflicts.
RSP1	IDCRS01	PROCCRA	dump	End of procedure that merges the records of a reset CRA into the workfile.
RSP2	IDCRS03	PROCVOL	dump	Before freeing resources used by PROCVOL routine.
RSP3	IDCRS02	PROCTYPE	dump	After processing a set of records for associations.
RSP4	IDCRS02	PROCCI	dump	End of procedure that ensures that a CIN is in the list of CINs.
RSRC	IDCRS06	RECMGMT	trace	On entry to routine that performs all I/O operations for RESETCAT.
RSRE	IDCRS01	REASSIGN	trace	On entry to routine that performs control interval reassignment.
RSRN	IDCRS07	RENAMEP	trace	On entry to routine that renames duplicate true name entries.
RSR1	IDCRS01	REASSIGN	dump	End of procedure that assigns new CINs to records on the reassign chain.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RSR2	IDCRS06	RECMGMT	dump	End of procedure that performs all I/O requests.
RSR4	IDCRS07	RENAMEP	dump	Before freeing resources used by the RENAMEP procedure.
RSSB	IDCRS03	SETBMAP	trace	On entry to routine that checks space conflicts for D or I type catalog entries.
RSSC	IDCRS02	SCANCI	trace	On entry to routine that scans records for control intervals.
RSSE	IDCRS02	SETCI	trace	On entry to routine that updates the workfile to reflect new CINs for reassigned CINs.
RSSR	IDCRS05	SCNRLST	trace	On entry to routine that obtains the next CRA volser entry for reset.
RSST	IDCRS03	SETBITS	trace	On entry to routine that maps extents to a bit map.
RSSV	IDCRS05	SCNVLST	trace	On entry to routine that scans through the list of volumes.
RSS2	IDCRS02	SETCI	dump	End of procedure that updates the workfile records from the associations tables.
RSS3	IDCRS03	SETBITS	dump	At end of procedure that sets up a single bit map.
RSS5	IDCRS05	SCNVLST	dump	End of procedure that locates an entry in the volume serial table.
RSS6	IDCRS05	SCNRLST	dump	End of procedure that locates an entry in the reset volume table.
RSUA	IDCRS03	UNALLOC	trace	On entry to routine which unallocates suballocated space from temporary space maps.
RSUC	IDCRS01	UPDCRA	trace	On entry to routine which updates the CRAs from the workfile.
RSUP	IDCRS01	UPDCAT	trace	On entry to routine which updates the catalog from the workfile.
RSUR	IDCRS07	UPDCCR	trace	On entry to procedure which updates the CCR for the catalog.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RSU1	IDCRS01	UPDCAT	dump	End of procedure that updates the catalog from the workfile.
RSU2	IDCRS01	UPDCRA	dump	End of procedure that updates the CRAs from the workfile.
RSVB	ICDRS03	VERB	trace	On entry to routine which verifies associations for GDG base records.
RSVC	IDCRS02	VERC	trace	On entry to routine which verifies associations for clusters.
RSVE	IDCRS02	VERDSDIR	trace	On entry to routine which verifies the data set directory entries for VSAM data sets not on reset volumes.
RSVG	IDCRS02	VERG	trace	On entry to routine which verifies associations for AIXs.
RSVN	IDCRS03	VLNRESET	trace	On entry to routine which verifies space requested from objects being reset against nonreset volumes.
RSVO	IDCRS01	VOLCHK	trace	On entry to volume consistency routine (VOLCHK).
RSVP	IDCRS02	VERR	trace	Upon entry to routine which verifies associations for PATHs.
RSVR	IDCRS02	VERCI	trace	On entry to routine which checks validity of each CIN found in a set of records.
RSVS	IDCRS03	VLRESET	trace	On entry to routine which verifies space requested against reset volumes.
RSVU	IDCRS02	VERU	trace	On entry to routine which verifies associations for user catalogs.
RSVX	IDCRS02	VERX	trace	On entry to routine which verifies alias associations.
RSV1	IDCRS03	VOLCHK	dump	End of procedure that checks Format 1 DSCBs against space headers.
RSV2	IDCRS02	VERDSDIR	dump	After verifying initial space claims.
RSV3	IDCRS02	VERCI	dump	After verifying associations on a set of records.
RSV4	IDCRS03	VERB	dump	Before freeing resources used by procedure which verifies GDG data sets.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
RSWF	IDCRS06	WFDEF	trace	Upon entry to routine which defines an RRDS as a workfile for RESETCAT processing.
RSWL	IDCRS06	WFDEL	trace	On entry to routine which deletes the workfile.
RSWR	IDCRS01	WRAPUP	trace	On entry to routine which handles clean-up operations after successful RESETCAT processing.
RSW2	IDCRS06	WFDEF	dump	Before the UCATLG work area is freed.
RSW3	IDCRS06	WFDEL	dump	End of procedure that deletes the workfile.
RS00	IDCRS01	IDCRS01	dump	End of RESETCAT FSR.
RS01	IDCRS01	IDCRS01	trace	Upon entry to main RESETCAT module.
S1S3	IDCSS01	READSTAT	trace	Entry to IDCSS03.
S4B3	IDCSS04	IDCSS04	both	After return from IDCSS03 when a BINDDATA command has been issued.
S4ED	IDCSS04	IDCSS04	both	End of PROCESS routine.
S4S2	IDCSS04	IDCSS04	both	After return from IDCSS02.
S4V1	IDCSS04	IDCSS04	both	After return from IDCSS03 first unit address when a SETCACHE command has been issued.
S4V2	IDCSS04	IDCSS04	both	After return from IDCSS03 second unit address when a SETCACHE command has been issued.
SAAL	IDCSA02	IDCSA02	trace	Start of UALLOC macro processing.
SABB	IDCSA02	FINDNAME	dump	After a new Load List Block has been built.
SACA	IDCSA02	IDCSA02	trace	Start of routine that processes UCATLG macro.
SACA	IDCSA09	CHECKARG	trace	Start of CHECKARG procedure that determines the macro to be issued and checks for errors.
SACC	IDCSA09	CHKCODE	trace	Start of CHKCODE procedure that returns error codes to the caller.
SACK	IDCSA06	UCBCHECK	trace	Start of routine that checks for UCB.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
SACL	IDCSA02	IDCSA02	trace	Start of routine that processes UCALL macro.
SACR	IDCSA02	IDCSA02	trace	Start of UCIR macro processing.
SADE	IDCSA02	IDCSA02	trace	Start of routine that processes UDELETE macro.
SADL	IDCSA02	IDCSA02	trace	Start of UDEALLOC macro processing.
SADM	IDCSA06	DEMNTVOL	trace	Start of procedure that calls ISSUEDMT and UCBPOST.
SADQ	IDCSA08	IDCSA08	trace	Start of routine that processes UDEQ macro.
SAD1	IDCSA01	IDCSA01	trace	During termination processing before all loaded modules are deleted.
SAD2	IDCSA01	IDCSA01	dump	During termination processing after all loaded modules have been deleted.
SAD3	IDCSA02	RELECORE	trace	Before loaded modules with zero use count are deleted.
SAD4	IDCSA02	RELECORE	dump	After deletion of loaded modules with zero use count.
SAED	IDCSA06	ENQDEQ	trace	Start of procedure that enqueues and dequeues.
SAEX	IDCSA06	IDCSA06	dump	End of IDCSA06 module.
SAFE	IDCSA06	FINDEXCL	trace	Start of procedure that determines whether the volume is already allocated exclusively to the job step.
SAFP	IDCSA02	IDCSA02	trace	Start of routine that processes UFPOOL macro.
SAFS	IDCSA02	IDCSA02	trace	Start of routine that processes UFSPACE macro.
SAGE	IDCSA06	GETEXCL	trace	Start of procedure that determines whether the allocation of a unit and volume can be changed to exclusive.
SAGP	IDCSA02	IDCSA02	trace	Start of routine that processes UGPOOL macro.
SAGS	IDCSA02	IDCSA02	trace	Start of routine that processes UGSPACE macro.
SAID	IDCSA02	IDCSA02	trace	Start of UID macro processing.
SAID	IDCSA06	ISSUEDMT	trace	Start of routine that calls USSC macro to demount volume.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
SAIM	IDCSA06		trace	Start of routine that issues USSC macro to mount volume.
SAIN	IDCSA06	IDCSA06	dump	Start of IDCSA06 module that performs MOUNT, DEMOUNT, POST, and CHECK.
SALC	IDCSA07	IDCSALC	dump	At entry to load module.
SALD	IDCSA02	IDCSA02	trace	Start of routine that processes ULOAD macro.
SALE	IDCSA07	IDCSALC	dump	At exit from load module.
SALK	IDCSA02	IDCSA02	trace	Start of procedure that processes ULINK macro.
SALT	IDCSA07	IDCSALC	trace	At entry to load module.
SAMA	IDCSA09	ISSUEMAC	trace	Start of ISSUEMAC procedure that issues the execute form of the requested macro.
SAMC	IDCSA06	MOUNTCTL	trace	Start of control procedure that processes mount requests.
SAMT	IDCSA06	MOUNTVOL	trace	Start of procedure that calls ISSUEMNT, RDLABEL, and UCBPOST.
SANQ	IDCSA08	IDCSA08	trace	Start of routine that processes UENQ macro.
SAPR	IDCSA06	MDSETUP	trace	Start of procedure that sets up for a mount or demount request.
SAPT	IDCSA02	IDCSA02	trace	Start of procedure that processes UPROMPT macro.
SAQL	IDCSA02	IDCSA02	trace	Start of UQUAL macro processing.
SARD	IDCSA06	RDLABEL	trace	Start of routine that issues the UEXCP macro to read volume label.
SARV	IDCSA08	IDCSA08	trace	Start of routine that processes URESERVE macro.
SAR1	IDCSA08	IDCSA08	dump	After the RACHECK parameter list has been built; before the RACHECK macro is issued.
SAR2	IDCSA08	IDCSA08	dump	Upon return from the RACHECK macro.
SASC	IDCSA08	IDCSA08	trace	Start of routine that processes USCRATCH macro.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
SASC	IDCSA06	SELESCAN	trace	At entry to SELESCAN procedure which scans the TIOT.
SASD	IDCSA06	SELECTD	trace	At entry to SELECTD procedure which searches to see if the caller's volume is mounted.
SASE	IDCSA06	SETEXCL	trace	Start of procedure that changes the enqueue on a volume from shared to exclusive and marks the UCB nonshareable.
SASI	IDCSA08	IDCSA08	trace	Start of routine that processes USYSINFO macro.
SASN	IDCSA02	IDCSA02	trace	Start of routine that processes USNAP macro.
SASS	IDCSA09	IDCSA09	both	Start of USSC macro.
SAST	IDCSA06	SCANTIOT	trace	Start of procedure that scans the TIOT to find a UCB and DD statement that can be used to process the volume. Start of USSC macro.
SATI	IDCSA02	IDCSA02	trace	Start of routine that processes UTIME macro.
SAUC	IDCSA07	IDCSAUC	dump	Start of UUNCATLG macro.
			trace	Start of UUNCATLG macro.
SAUE	IDCSA07	IDCSAUC	dump	Exit from UUNCATLG macro.
SAUP	IDCSA06	UCBPOST	trace	Start of procedure that builds parameter list and issues SVC 82 to post or clear a UCB.
SAVL	IDCSA07	VSAMLOC	trace	At entry to VSAMLOC procedure.
SAVU	IDCSA07	VSAMUCT	trace	At entry to procedure that issues VSAM DELETE.
SAWO	IDCSA08	IDCSA08	trace	Start of routine that processes UWTO macro.
SA05	IDCSA05	IDCSA05	trace	Before the TIME macro is issued.
SA06	IDCSA06	IDCSA06	trace	Start IDCSA06 module.
SC01	IDCSC01	IDCSC01	trace	Start of SETCACHE FSR.
SCED	IDCSC01	IDCSC01	both	End of SETCACHE FSR.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
SCG4	IDCSC01	IDCSC01	both	Before link to IDCSS04.
SCR4	IDCSC01	IDCSC01	both	After return from IDCSS04.
SCST	IDCSC01	IDCSC01	dump	Start of SETCACHE FSR.
SS01	IDCSC01	IDCSC01	both	Start of IDCSS01.
SS04	IDCSC04	IDCSC04	trace	Start of IDCSS04.
SSST	IDCSC04	IDCSC04	dump	Start of IDCSS04.
STBG	IDCSA10	IDCSAST	dump	At start of module that establishes or cancels an ESTAE environment.
STCS	IDCSA10	CANSTAE	trace	At entry to CANSTAE procedure that issues an ESTAE macro to cancel the ESTAE environment.
STEN	IDCSA10	IDCSAST	dump	At end of module.
STEX	IDCSA10	STAEEXIT	trace	At entry to ESTAEEXIT procedure that determines whether it can retry.
STRY	IDCSA10	RECOVERY	trace	At entry to RECOVERY procedure that cleans up before ABEND.
STSS	IDCSA10	SETSTAE	trace	At entry to SETSTAE procedure that issues an ESTAE macro to establish an ESTAE environment.
STST	IDCSA10	IDCSAST	trace	At entry to USTAE macro that establishes or cancels an ESTAE environment.
TPCC	IDCTP01	IDCTP01	trace	Before the call to the CONVERT routine is issued.
TPEA	IDCTP06	IDCTP06	dump	Start of procedure that processes UERROR macro.
TPEB	IDCTP06	IDCTP06	dump	Before converted message is printed via the UPRINT macro.
TPER	IDCTP01	ERROR	dump	Start of procedure that prints a text processor error message.
TPE1	IDCTP06	IDCTP06	trace	Start of procedure that processes UERROR macro.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
TPE2	IDCTP06	CATERCNV	trace	Entry point to routine that converts catalog error messages to prose.
TPE4	IDCTP06	DAERCNV	trace	On entry to dynamic allocation error conversion routine.
TPE5	IDCTP06	DAERCNV	trace	Before call to DAIRFAIL service routine.
TPIN	IDCTP01	IDCTPPR	dump	At end of phase; the format structure for a UPRINT macro has been processed.
TPMS	IDCTP01	LINEPRT	dump	On return from segment routine.
TPSI	IDCTP01	IDCTPPR	dump	After initialization of text processor parameters.
TPXX	IDCTP01	STACKPUT	dump	Before call to UPUT.
TPX1	IDCTP01	LINEPRT	dump	After return from procedure that puts print lines into output buffer.
TP2I	IDCTP01	CONVERT	dump	Start of procedure that converts data to a printable form.
TP2N	IDCTP01	CONVERT	dump	End of procedure that converts data to a printable form.
TP3I	IDCTP01	LINEPRT	dump	Start of procedure that formats pages and prints titles, headings, footings, and other lines requested.
TP3N	IDCTP01	LINEPRT	dump	End of procedure that prints lines.
TP4A	IDCTP04	ESTACONT	dump	End of procedure that processes the UESTA macro.
TP4R	IDCTP04	RESTCONT	dump	End of procedure that processes UREST macro.
TP4S	IDCTP04	ESTSCONT	dump	End of procedure that processes UESTS macro.
TP5E	IDCTP05	IDCTP05	trace	Start of procedure to get a static text module.
TP5I	IDCTP05	IDCTP05	dump	Start of phase that loads the static text phase.
TP5N	IDCTP05	IDCTP05	dump	End of phase that loads the static text phase.
VSBG	IDCVS01	IDCBS01	dump	Start of IDCVS01 module.
VSBG	IDCVS02	IDCVS02	dump	Start of IDCVS02 module.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
VSCC	IDCVS01	CATCHK	trace	Start of CATCHK procedure that initializes the catalog parameter list.
VSCG	IDCVS03	CATLG	trace	Start of CATLG procedure.
VSCO	IDCVS01	CATOPEN	trace	Start of CATOPEN procedure that issues a UOPEN macro to open the catalog.
VSCR	IDCVS03	CREATION	trace	Start of CREATION procedure that checks data set creation date.
VSCT	IDCVS03	CRITERIA	trace	Start of CRITERIA procedure that checks if data set meets caller's criteria.
VSCU	IDCVS03	CLEANUP	trace	At entry to CLEANUP procedure that releases resources.
VSEC	IDCVS01	EXPIRCHK	trace	Start of EXPIRCHK procedure that opens an unexpired non-VSAM data set.
VSEL	IDCVS03	EXCLUDE	trace	At entry to EXCLUDE procedure which excludes data sets which caller does not want included in data set array list.
VSEX	IDCVS03	EXPIRATN	trace	At entry to procedure that checks expiration date.
VSF0	IDCVS02	FMTTRK0	trace	Start of FMTTRK routine that prepares the three records written to cylinder 0, track 0.
VSF1	IDCVS02	FMTTRK1	trace	Start of FMTTRK routine that prepares the 39 DSCBs written to cylinder 0, track 1.
VSF1	IDCVS03	GETFMT1	trace	Start of procedure that reads a format 1 DSCB from the VTOC (GETFMT1).
VSGB	IDCVS03	GETBLK	trace	Start of GETBLK procedure which computes and obtains storage for IDCVS03.
VSGL	IDCVS02	GETLAB	trace	Start of procedure that reads VTOC label.
VSIN	IDCVS02	INITVOLM	trace	Start of procedure that issues EXCP and calls FMTTRK1 and FMTTRK0.
VSIT	IDCVS03	INITIAL	trace	Start of initialization procedure (INITIAL).

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
VSMN	IDCVS02	IDCVS02	trace	Start of IDCVS02 module.
VSNC	IDCVS01	NVSAMCHK	trace	Start of NVSAMCHK procedure that issues a UEXCP macro to open non-VSAM data sets.
VSND	IDCVS01	IDCVS01	dump	End of IDCVS01 module.
VSOP	IDCVS03	OPENVTOC	trace	At entry to routine that opens the VTOC.
VSOT	IDCVS02	IDCVS02	dump	End of IDCVS02 module.
VSPI	IDCVS02	PUTLAB	trace	Start of procedure that writes VTOC label.
VSQL	IDCVS03	QUAL	trace	Start of routine which processes data set names with qualifiers (QUAL).
VSRC	IDCVS01	RECATLOG	trace	Start of RECATLOG procedure that recatalogs non-VSAM data sets.
VSSC	IDCVS01	SCRATCH	trace	Start of SCRATCH procedure that calls VTOCOPEN and FMT4SCR procedures.
VSSE	IDCVS01	SECHECK	trace	Start of SECHECK procedure that ensures that a volume is empty or has the correct password.
VSSN	IDCVS03	SYSNAMES	trace	Start of SYSNAMES procedure which checks for system names.
VSSR	IDCVS03	SELECTOR	trace	Start of SELECTOR procedure that selects eligible data sets from the VTOC.
VSST	IDCVS03	STATUS	trace	Start of STATUS routine which provides status information for the data sets selected.
VSTK	IDCVS03	TRKREAD	trace	Start of routine that reads a track into the track I/O buffer.
VSUS	IDCVS03	USAGE	trace	Start of USAGE routine that determines the organization and size of a data set.
VSVC	IDCVS01	VSAMCHK	trace	Start of VSAMCHK procedure that issues a UEXCP macro to open VSAM data sets.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
VSVG	IDCVS01	VTOCGET	trace	Start of VTOCGET procedure that calls the VTOCOPEN and FMTR4READ procedures.
VSVP	IDCVS01	VTOCPUT	trace	Start of VTOCPUT procedure that calls the VTOCOPEN and FMT4READ procedures.
VSVT	IDCVS01	VTOCOPEN	trace	Start of VTOCPUT procedure that opens the VTOC.
VS01	IDCVS01	IDCVS01	trace	Start of IDCVS01 procedure that performs VTOC services.
VS03	IDCVS03	IDCVS03	dump	At entry to load module.
VS1C	IDCVS01	FMT1CHK	trace	Start of FMT1CHK procedure that checks for format-1 DSCB.
VS1R	IDCVS01	FMT1READ	trace	Start of FMT1READ procedure that reads the format-1 DSCB.
VS1S	IDCVS01	FMT1SCR	trace	Start of FMT1SCR procedure that scratches the format-1 DSCB.
VS3X	IDCVS03	IDCVS03	dump	At exit from load module.
VS4C	IDCVS01	FMT4CHK	trace	Start of FMT4CHK procedure that checks the format-4 DSCB format for VSAM catalog ownership.
VS4R	IDCVS01	FMT4READ	trace	Start of FMT4READ procedure that reads the format-4 DSCB.
VS4S	IDCVS01	FMT4SCR	trace	Start of FMT4SCR procedure that scratches the format-4 DSCB.
VYBG	IDCVY01	IDCVY01	dump	Start of VERIFY FSR.
VYCL	IDCVY01	TERMPROC	trace	Start of procedure that closes the data set that was verified.
VYND	IDCVY01	IDCVY01	dump	End of VERIFY FSR.
VYOP	IDCVY01	OPENPROC	trace	Start of procedure that opens the VSAM data set to be verified.
VYST	IDCVY01	IDCVY01	trace	Start of VERIFY FSR.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
XM1A	IDCXM01	IDCXM01	trace	Start of EXAMINE FSR.
XM1B	IDCXM01	IDCXM01	trace	Return from IDCXM02.
XM1C	IDCXM01	IDCXM01	trace	Return from IDCXM03.
XM1X	IDCXM01	MSGDTEXT	trace	Start of message procedure.
XM11	IDCXM01	IDCXM01	dump	Data component was modified during testing.
XM12	IDCXM01	IDCXM01	dump	Error obtaining TSO qualifiers.
XM13	IDCXM01	IDCXM01	dump	Error obtaining volume information.
XM14	IDCXM01	IDCXM01	dump	Error obtaining Format-1 DSCB.
XM15	IDCXM01	IDCXM01	dump	OPEN indicator is on at OPEN.
XM16	IDCXM01	IDCXM01	dump	OPEN failure.
XM17	IDCXM01	IDCXM01	dump	Not key-sequenced data set (KSDS) cluster.
XM18	IDCXM01	IDCXM01	dump	In create mode at OPEN.
XM19	IDCXM01	IDCXM01	dump	Index component was modified during testing.
XM2A	IDCXM02	IDCXM02	trace	Prior to data initialization.
XM2B	IDCXM02	IDCXM02.	trace	At index ARDB loop.
XM2C	IDCXM02	SEQIXTST	trace	Prior to UGPOOL.
XM2D	IDCXM02	SEQIXTST	trace	Prior to diagnostic testing using FCICNT procedure.
XM2E	IDCXM02	MULIXTST	trace	Prior to UGPOOL, in multi-level index testing.
XM2F	IDCXM02	MULIXTST	trace	In index level loop (beginning).
XM2G	IDCXM02	MULIXTST	trace	At beginning of sequence set (SSTSTLP) test loop.
XM2H	IDCXM02	MULIXTST	trace	At end of SSTSTLP loop.
XM2I	IDCXM02	MULIXTST	trace	At beginning of intermediate level test.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
XM2J	IDCXM02	MULIXTST	trace	At beginning of intermediate level loop (ILTLOOP).
XM2K	IDCXM02	MULIXTST	trace	At terminal point of ILTLOOP.
XM2L	IDCXM02	SSTEST	trace	At beginning of procedure.
XM2M	IDCXM02	SSTEST	trace	Within index pointer loop, at final test of procedure.
XM2N	IDCXM02	STORIXP	trace	At beginning of main loop (STIXLP).
XM2O	IDCXM02	STORIXP	trace	At exit (STORIXP procedure).
XM2P	IDCXM02	XTEST	trace	At entry to procedure.
XM2Q	IDCXM02	XTEST	trace	Start of MISSCI1 routine.
XM2R	IDCXM02	FCICNT	trace	At entry to procedure.
XM2S	IDCXM02	GETIXCI	trace	At entry to procedure.
XM2T	IDCXM02	BRBTST	trace	At entry to procedure.
XM2U	IDCXM02	BRBTST	trace	At beginning of base relative byte address test loop (DUPLP).
XM2V	IDCXM02	DUPTEST	trace	At entry to procedure.
XM2W	IDCXM02	STORHHP	trace	At entry to procedure.
XM2X	IDCXM02	MSGHNDLR	trace	At entry to procedure.
XM2Z	IDCXM02	IBCXM02	trace	On return from SEQIXTST or from MULIXTST.
XM20	IDCXM02	IDCXM02.	dump	At UGPOOL, prior to index test selection.
XM21	IDCXM02	SEQIXTST	dump	After UGPOOL.
XM22	IDCXM02	MULIXTST	dump	After UGPOOL.
XM23	IDCXM02	MULIXTST	dump	After UGPOOL.
XM24	IDCXM02	MULIXTST	dump	After UGPOOL.
XM25	IDCXM02	GETIXCI	dump	After UGET.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
XM26	IDCXM02	MSGHNDLR	dump	After UPRINT.
XM3A	IDCXM03	IDCXM03	trace	After initialization and prior to reviewing the VSAM control blocks, access method data set statistics block, and address range definition block.
XM3B	IDCXM03	IDCXM03	trace	After reviewing the VSAM control blocks and prior to obtaining storage for index control interval, data control interval, index horizontal pointer table, index vertical pointer table, and vertical base relative byte address table.
XM3C	IDCXM03	IDCXM03	trace	Prior to calling INDXPROC.
XM3D	IDCXM03	IDCXM03	trace	Prior to displaying the statistical messages before returning to the calling module (IDCXM01).
XM3E	IDCXM03	INDXPROC	trace	Prior to reading all sequence set index control intervals and used for each read index.
XM3F	IDCXM03	DATAPROC	trace	Prior to reading a data control interval.
XM3G	IDCXM03	DATAPROC	trace	Prior to processing all record definition fields and used for each record definition field group.
XM3H	IDCXM03	VALAMDSB	trace	Prior to validating the information contained in the sequence set and data access method data set statistics block.
XM3I	IDCXM03	VALARDB	trace	Prior to validating the information contained in each of the address range definition blocks.
XM3J	IDCXM03	VALRBA	trace	Prior to validating the range of all relative byte addresses.
XM3K	IDCXM03	CKFRCI	trace	Prior to processing all free space control intervals and validating the absence of data.
XM3L	IDCXM03	VALRDF	trace	Prior to validating the accuracy of the information contained in a data control interval record definition field.
XM3M	IDCXM03	VINDXHDR	trace	Prior to validating the accuracy of the information contained in the index header of an index sequence set control interval.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
XM3N	IDCXM03	SPANNED	trace	Prior to processing the accuracy and positioning of spanned records.
XM3O	IDCXM03	UNSPNNED	trace	Prior to processing non-spanned records.
XM3P	IDCXM03	CKDLCI	trace	Prior to processing a deleted data control interval.
XM3Q	IDCXM03	INSEQTST	trace	Prior to processing a key sequence check on the index entries contained in the sequence set control interval.
XM3R	IDCXM03	DASEQTST	trace	Prior to processing a key sequence check on the records contained in the data control interval.
XM3S	IDCXM03	CRSSCHCK	trace	Prior to checking an index key against data key.
XM3T	IDCXM03	GPOOL	trace	Prior to obtaining storage via the UGPOOL macro.
XM3U	IDCXM03	RDDATACI	trace	Prior to reading and checking a data control interval for control interval split in progress.
XM3V	IDCXM03	DSPINDCI	trace	Prior to display of a sequence set control interval.
XM3W	IDCXM03	DSPDATCI	trace	Prior to display of a data control interval.
XM3X	IDCXM03	MSGDTEXT	trace	Prior to issuing a message containing only static text.
XM3Y	IDCXM03	MSGDDISP	trace	Prior to issuing a data display (dump) message.
XM3Z	IDCXM03	MSGDNSRT	trace	Prior to issuing a static text message containing insert data.
XM30	IDCXM03	INDXPROC	dump	An error was returned following an attempt to read an index sequence set control interval.
XM31	IDCXM03	INDXPROC	dump	An error was returned following an attempt to read a data control interval prior to testing for software end-of-file.
XM32	IDCXM03	VALAMDSB	dump	An error was detected in the sequence set access method data set statistics block.
XM33	IDCXM03	VALARDB	dump	An error was detected in an address range definition block.
XM35	IDCXM03	GPOOL	dump	An error was returned following an attempt to obtain storage for a data control interval.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
XM36	IDCXM03	RDDATACI	dump	An error was returned following an attempt to read a data control interval.
XPAO	IDCXP01	CLUSPROC	trace	Before retrieving from the catalog the entries associated with the cluster, alternate index, or ICF catalog being exported.
XPAP	IDCXP01	ALTRPROC	trace	Start of procedure that modifies the CTGPL to set the temporary export flag on.
XPBG	IDCXP01	IDCXP01	trace	Start of EXPORT FSR.
XPCP	IDCXP01	CLUSPROC	trace	Before retrieving the catalog entry for the object to be exported.
XPCR	IDCXP01	CONTRBL	trace	Before constructing control records for the portable data set.
XPCW	IDCXP01	CONTRBL	trace	Before writing control records to the portable data set.
XPDP	IDCXP01	DELTPROC	trace	Start of procedure that sets up the CTGPL to delete a cluster or disconnect a user catalog.
XPED	IDCXP01	IDCXP01	trace	End of EXPORT FSR.
XPFN	IDCXP01	IDCXP01	dump	End of EXPORT FSR, before data sets are closed and space freed.
XPLP	IDCXP01	LOCPROC	trace	Start of procedure that builds the CTGPL and CTGFLs for a locate request.
XPMS	IDCXP01	MORESP	trace	Entry to MORESP.
XPOP	IDCXP01	OPENPROC	trace	Start of procedure that opens either the portable data set or the cluster, alternate index, or ICF catalog to be exported.
XPPM	IDCXP01	CLUSPROC	trace	Before processing the permanent or temporary export option.
XPPP	IDCXP01	PUTPROC	trace	Start of procedure that writes a record to the portable data set.
XPRP	IDCXP01	RECPROC	trace	Entry to RECPROC.

Trace and Dump Points to Module or CSECT Cross-Reference

Trace or Dump Point	Module or CSECT	Procedure	Type	Situation at Dump or Trace Point
XPSP	IDCXP01	CTLGPROC	trace	Start of procedure that calls the catalog for a locate, alter, or delete request.
XPTM	IDCXP01	CLUSPROC	trace	Before calling the procedure to alter the CTGPL to set the temporary export flag.
XPUC	IDCXP01	DSCTPROC	trace	Start of procedure that disconnects a user catalog.
XPWC	IDCXP01	CLUSPROC	trace	Before writing the catalog information to the portable data set.
XPZY	IDCXP01	DELTPROC	dump	Just after the UCATLG macro.
XPZZ	IDCXP01	CTLGPROC	dump	After calling the catalog to locate, alter, or delete an entry.
XP01	IDCXP01	IDCXP01	dump	Start of EXPORT FSR.
ZZAL	IDCSA02	IDCSA02	dump	Before and after each time UALLOC invokes dynamic allocation.
ZZCA	IDCSA02	IDCSA02	dump	Before and after CATLG macro is issued to invoke catalog management routines.
ZZCR	IDCSA02	IDCSA02	dump	Start of procedure that processes UCIR macro and just before the UCIR macro returns to the module that issued the UCIR.
ZZDL	IDCSA02	IDCSA02	dump	Before and after the UDEALLOC macro invokes dynamic allocation.
ZZSC	IDCSA09	ISSUEMAC	dump	Before issuing SVC126.

Module to Dump Points Cross-Reference

The following list contains the dump points within each module and procedure, indicates what information can be dumped at each point (either a full dump or selected areas), and explains the situation at the dump point. As explained in the section "TEST Keyword" in this chapter, full region dumps are taken at all dump points in this list. Selected areas can be printed with either the AREAS or FULL variation of the test option. Details of the selected areas are given in the notes following the list.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
IDCAL01	ASSOCPRC	AL92	dump	After the associate locate call to CMS.
		AL93	dump	After the DSATTR locate call to CMS.
	CHECKPRC	AL51	dump	After locating data component of the alternate index for which UPGRADE has been specified.
		AL52	dump	After locating associated cluster or the alternate index of the data object specified on ALTER command.
		AL53	dump	After locating associated index component.
		AL54	dump	After locating the data component of the path's base cluster.
		AL55	dump	After locating the cluster component of the alternate index's base cluster.
		AL56	dump	After locating the data component of the alternate index's base cluster.
	IDCAL01	AL01	dump	Before calling the catalog to alter an object.
		AL02	dump	End of ALTER FSR.
		AL04	dump	Before issuing ALTER request for index objects if KEYS specified.
	LOCATPRC	AL03	dump	After calling the catalog to locate an object.
IDCBD01	IDCBD01	BDST	dump	Start of BINDDATA FSR.
		BDG4	dump	Before link to IDCSS04.
		BDR4	dump	After return from IDCSS04.
		BDED	dump	After return from IDCSS04.
IDCBI01	CATPROC	BIL2	dump	After return from UCATLG for each locate request.
	CNTRLPRC	BIC2	dump	After completion of sort if an internal sort. After completion of sort phase and before merge passes if an external sort.
	DEFPROC	BID2	dump	After return from UCATLG to define each sort work file.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
	DELTPROC	BIDL	dump	After return from UCATLG to delete each sort work file.
	INITPROC	BII2	dump	After obtaining or failing to obtain sort storage.
	JCPROC	BIJ2	dump	After return from each call to UIOINFO.
	MAINPROC	BI03	dump	After return from procedure which locates information about the base cluster and alternate index.
		BI04	dump	After the alternate index has been built, before CLOSE.
	MERGPROC	BIM3	dump	After the tree has been initialized for each merge pass of an external sort.
		BIM4	dump	After processing one set of strings during the merge pass of an external sort.
	OPENPROC	BIP2	dump	After return from UOPEN to open a data set.
	SORTPROC	BISR	dump	Before sorting the records in the record sort area.
IDCCC01	ALTRLSTS	CCAC	selected areas 1	After building parameter lists to alter a VSAM or ICF catalog.
		CCAV	selected areas 2	After building parameter lists to alter a VSAM or ICF catalog.
	CATALOG	CCLV	selected areas 3	After request to locate volume information on a duplicate entry.
	CNVTINIT	CCIC	dump	After initializing for scanning of OS/VS catalog.
	CONTINUE	CCCN	selected areas 4	After obtaining block at the same index level as the last block from an OS/VS catalog.
	CONVERT	CCSE	dump	Invalid entry type in the OS/VS catalog.
		CCSL	dump	Finished converting OS/VS catalog entries to VSAM or ICF catalog entries.
	DEFNLSTS	CCDC	selected areas 5	After building a VSAM parameter list for a define.
		CCDV	selected areas 6	After building a VSAM parameter list for a define.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
	LOCTLSTS	CCLC	selected areas 7	After building parameter lists to locate information in the VSAM or ICF catalog.
	POPUP	CCPU	dump	Ready to continue scan of higher level index block.
	PUSHDOWN	CCIE	dump	No index control entry found on a lower level scan in the OS/VS catalog.
		CCPD	selected areas 8	After obtaining a block at the same index level as the last block from the OS/VS catalog.
	VOLINDEX	CCVE	dump	No volume index control entry in the OS/VS catalog.
		CCVI	selected areas 9	First block of volume index is obtained from the OS/VS catalog.
IDCCC02	CTLGPROC	CCZY	dump	Dump point prior to catalog management call.
IDCCC02	CTLGPROC	CCZZ	dump	Dump point following catalog management call.
IDCCK01	DSDRPROC	CK30	dump	After reading a type 1 DSDR record.
		CK40	dump	Before processing a type 1 DSDR record for a tape data set.
	DSDRVOLS	CK50	dump	Before processing a type 2 DSDR record.
	IDCCK01	CK10	dump	Before calling CHR procedure.
		CK20	dump	Upon return from CHR procedure.
IDCDA01	DUMPDAZY	DAZY	dump	Procedure called before any termination code gets control.
	TERM	DAZZ	dump	Before storage is FREEMAINed.
IDCDE01	IDCDE01	DE01	dump	Before calling the catalog to define an object.
		DE02	dump	End of DEFINE FSR, before completion message is issued.
IDCDE02	MODELPRC	DE03	dump	After calling the catalog to locate a model object.
		DE04	dump	End of procedure that built the model table.
IDCDE02	FREESTG	DE37	dump	End of IDCDE02 CSECT.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
IDCDL01	CATCALL	DLDB	dump	Before calling the catalog to delete an entry.
	CATCALL	DLDA	dump	After calling the catalog to delete an entry.
	FINDTYPE	DLLB	dump	Before calling the catalog to locate the entry type.
	FINDTYPE	DLLA	dump	After calling the catalog to locate the entry type.
	IDCDL01	DLBG	dump	Start of DELETE FSR.
		DLND	dump	End of DELETE FSR, before data sets are closed and the completion message is issued.
IDCEX01	CALLFSR	EXFS	dump	Before each call to an FSR.
	CALLRI	EXRI	dump	Before each call to the reader/interpreter.
	IDCEX01	EXMN	dump	All reader/interpreter FSR processing is complete.
IDCIO01	CRYPTRH	IOC3	dump	End of procedure that builds and writes the encipher data set header.
	CRYPTWH	IOC2	dump	End of procedure that builds and writes the encipher data set header.
	GETEXT	IOEG	dump	End of procedure that gets a record from the user routine.
	GETVSAM	IOVG	dump	End of procedure that gets a record or control interval from a VSAM data set.
	IDCIOCR	IOC4	dump	After detection of an error while processing the encipher data set header.
	IDCIOVY	IOVY	dump	After VERIFY macro is issued.
	PUTEXT	IOEP	dump	After control returns from an external user routine.
	PUTNONVS	IO2P	dump	After writing a spanned record.
	PUTREP	IOPR	dump	After the PUT for update.
		IOGR	dump	After the GET for update.
	PUTVSAM	IOVP	dump	End of procedure that writes a VSAM record.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
	VSAMERR	IOVR	dump	After detection of a VSAM I/O error.
IDCIO02	BUILDACB	IOAC	dump	After ACB and EXLST have been built, at end of procedure.
	BUILDRPL	IORP	dump	After RPL is built, at end of procedure.
	CLOSERTN	IO1C	dump	Before CLOSE macro is issued.
		IO2C	dump	At end of all UCLOSE processing.
	DSDATA	IODS	dump	After obtaining data set information from the JFCB.
	OPENRTN	IO1O	dump	Before OPEN macro is issued.
		IO20	dump	After OPEN macro is issued.
		IO21	dump	After all UOPEN processing.
IDCIO03	DSINFO	IO00	dump	After RDJFCB macro is issued.
		IO01	dump	After return from DEVTYPE.
		IO02	dump	After formatting the work area.
	IDCIO03	IOPO	dump	After positioning is complete, before returning control to IDCIOPO.
	STOWRTN	IOW1	dump	Before STOW macro is issued.
		IOW2	dump	After STOW macro is issued.
IDCIO05	ADJCCW	IOR7	dump	Start of procedure that adjusts a CCW channel program.
	IDCIO05	IOIN	dump	Start of IDCIO05 module.
		IOEX	dump	End of IDCIO05 module routine.
	OCABEND	IOAB	dump	Start of OPEN/CLOSE ABEND routine.
	OPENR	IOR1	dump	Start of procedure that opens data sets, VTOCs, and staging packs for REPAIRV.
	READCNT	IOR2	dump	Start of procedure that reads the count field of each record on a track.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
	READKD	IOR3	dump	Start of procedure that reads the count, key, and data fields of each record on a track.
	SETCCW	IOR5	dump	Start of procedure that builds the CCW channel program to read the count, key, and data fields of each record on a track.
	SETKDCCW	IOR6	dump	Start of procedure that builds the CCW channel program to read the count, key, and data fields of each record on a track.
	SPACCR	IOR4	dump	Start of procedure that spaces over a defective count field on a track.
	SYNAD	IOSY	dump	Start of SYNAD routine.
IDCLC01	IDCLA01	LA01	dump	Start of LISTDATA FSR.
		LAG1	dump	Before link to IDCSS01.
		LAR1	dump	After return from IDCSS01.
IDCLC01	ENTPROC	LCOJ	selected areas 12	After preparing to locate information about a name.
	IDCLC01	LC99	dump	End of LISTCAT FSR, before freeing storage in IDCLC01.
	RTEPROC	LCRA	selected areas 19	Processing catalog information for associations of a cluster, pagespace, or generation data group, or AIX.
		LCRP	selected areas 20	Start of procedure that determines which procedure will list the catalog entry.
IDCLC02	AUPROC	LCAP	selected areas 10	Start of procedure that lists information about alias, non-VSAM, user catalog, or generation data group.
	CDIPROC	LCCD	selected areas 11	Start of procedure that lists information about cluster, pagespace, data, or index, AIX and PATH.
	ERRPROC	LCMG	selected areas 13	Before UPRINT macro is issued to print a message.
	FREESTG	LC98	dump	End of LISTCAT FSR, before freeing IDCLC02's automatic storage.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
	IDCLC02	LC02	dump	When IDCLC02 is called the first time to establish addressability.
	LISTPROC	LCTP	selected areas 14	Before UPRINT macro is issued to print catalog data.
	LOCPROC	LCAL	selected areas 15	After calling the catalog to locate an entry.
		LCBL	selected areas 16	Before calling the catalog to locate an entry.
		LCLO	selected areas 17	Ready to do a catalog locate for information.
		LCWA	selected areas 18	After calling the catalog to locate an entry.
IDCLR01	AATOPLR	LRAA	dump	Entry point for IDCLR01.
	ADDASOC	LRAD	dump	Start of procedure that adds an association to the association table.
	BLDVEXT	LRBL	dump	Start of procedure that builds vertical extension tables.
	BUFSHUF	LRBU	dump	Start of procedure that moves a record to its buffer.
	CATOPEN	LCRA	dump	Start of procedure that prepares to open the catalog.
	CKEYRNG	LRCK	dump	Start of procedure that checks for keyrange.
	CLEANUP	LRC1	dump	Start of procedure that cleans up before exit.
	CLENCRA	LRC2	dump	Start of procedure that closes the CRA and prints miscompared message.
	CRAOPEN	LRCR	dump	Start of procedure that opens the CRA.
	CTTBLD	LRCT	dump	Start of procedure that builds CI translate table.
	DOOTHR	LRDO	dump	Start of procedure that looks for non-VSAM information.
	DOVSAM	LRDV	dump	Start of procedure that looks for VSAM information.
	ERROR	LRER	dump	Start of procedure that handles errors.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
	GETPRT	LRGE	dump	Start of procedure that gets and prints records.
	INITLZE	LRIN	dump	Start of procedure that initializes the FSR.
	INTASOC	LRIA	dump	Start of procedure that initializes association tables.
	INTSORT	LRIS	dump	Start of procedure that initializes the sort table.
	INTVEXT	LRIV	dump	Start of procedure that initializes the vertical extension table.
	MEMSORT	LRME	dump	Start of procedure that sorts the entries in sort table.
	PRTAAXV	LRPA	dump	Start of procedure that prints associated AIXs and volumes.
	PRTCMP	LRPC	dump	Start of procedure that prints compare information.
	PRTDMP	LRPD	dump	Start of procedure that prints dump if specified.
	PRTDMPC	LRPE	dump	Start of procedure that prints dump of catalog record and underscores miscompared records.
	PRTFIFO	LRPF	dump	Start of procedure that prints CRA in order of CI number.
	PRTMCWD	LRPM	dump	Start of procedure that prints miscompared words.
	PRTOJAL	LRPJ	dump	Start of procedure that prints object aliases.
	PRTOJVL	LRPK	dump	Start of procedure that prints an object's volumes.
	PROTHR	LRPO	dump	Start of procedure that prints non-VSAM objects.
	PRTTIME	LRPT	dump	Start of procedure that prints timestamps.
	PRTVOL	LRPW	dump	Start of procedure that prints volume records.
	PRTVSAM	LRPV	dump	Start of procedure that prints VSAM structures.
	SUMIT	LRSM	dump	Start of procedure that prints number of entries processed.
	TCICTCR	LRTC	dump	Start of procedure that translates the catalog CI to the CRA.
	VERTEXT	LRVE	dump	Start of procedure that handles vertical extension records.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
IDCLR02	IDCLR02	LR02	dump	Entry point for module that gets a record for recovery field management routine.
IDCMP01	CTLGPROC	MPZZ	dump	Before and after calling the catalog to locate, alter, or define an entry.
	DELTPROC	MPDC	dump	After the first UCATLG.
		MPDD	dump	After the second UCATLG.
	IDCMP01	MPFN	dump	End of IMPORT FSR, prior to closing data sets.
IDCPR01	IDCPR01	PR01	dump	End of PRINT FSR.
IDCRC01	CKNAMES	RC99	dump	Before gathering information on name in name list from CRA.
	DIRECT	RC98	dump	Before obtaining a directory for a volume.
	NAMETABL	RC97	dump	Before marking or adding a name to the name chain.
	ERRCK	RC96	dump	After opening a CRA.
	SCANCRA	RC95	dump	Before scanning CRA to build the CTT table.
	TIMESTMP	RC94	dump	Before obtaining format-4 timestamp for CRA being opened.
	CKCATNM	RC93	dump	Before checking owning catalog name of CRA being opened.
	OPEN	RC92	dump	Before the opening of the CRA.
	OPENCRA	RC91	dump	Before opening or closing a CRA and doing all other work (for example, build CTT).
	EXTRACT	RC90	dump	Before using internal field management to get information from CRA.
	MESSAGE	RC89	dump	Before printing any message from IDCRC01.
	SUBSP	RC88	dump	Before allocating space for the name chain.
	COMPNAME	RC87	dump	Before compressing a name for the name list.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
	BUILDNAM	RC86	dump	Before constructing a block for the name chain.
	DUPNAMCK	RC85	dump	Before checking the name chain for duplicates.
	OBJVOLCK	RC84	dump	Before checking sync. of entry across multiple volumes.
	SYNCH	RC83	dump	Before scanning the name chain for a CRA to check it.
	EXPORTDR	RC82	dump	Before looping down name chain to call IDCRC02 to export data sets.
	BUILDCRV	RC81	dump	Before building the CRV.
IDCRC01	INIT	RC80	dump	Before initializing to begin processing.
	TERM	RC79	dump	Before special processing to terminate request (closing output data set).
IDCRC02	ASOCPROC	RC40	dump	Start of procedure which processes non-VSAM objects associated with GDGs.
	CLUSPROC	RC06	dump	Start of procedure which processes VSAM objects.
	CTLGPROC	RC16	dump	Start of procedure which issues catalog locates.
	GDGPROC	RC36	dump	Start of procedure which processes GDG objects.
	IDCRC02	RC02	dump	Start of main procedure.
	IDCRC02	RC04	dump	Before termination processing.
	NVSMPROC	RC28	dump	Start of procedure which processes non-VSAM objects not associated with GDGs.
IDCRI01	BYPASTRM	RIBT	dump	Start of procedure that bypasses the remainder of the current modal or null command.
	CONVERT	RICV	dump	Start of procedure that converts a constant from EBCDIC to binary or hexadecimal.
	DEFAULTS	RIDF	dump	Start of procedure that adds default parameters to the FDT.
	DSPLCALC	RIDC	dump	Start of procedure that calculates the position within a secondary FDT vector in which to place an FDT pointer.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
	ERROR1	RIE1	dump	Start of procedure that issues a message without inserted text.
	ERROR2	RIE2	dump	Start of procedure that issues a message with inserted text.
	GETNEXT	RIGN	dump	Start of procedure that scans the input command.
	GETQUOTD	RIGQ	dump	Start of procedure that scans a quoted constant.
	GETRECRD	RIGR	dump	Start of procedure that obtains the next input record.
	IDCRI01	RIEX	dump	Start of reader/interpreter module.
	INREPEAT	RIIR	dump	Start of procedure that scans a repeated parameter set.
	MODALIF	RIMI	dump	Start of procedure that scans an IF modal command.
	MODALSET	RIMS	dump	Start of procedure that scans a SET modal command.
	MODLELSE	RIME	dump	Start of procedure that scans an ELSE modal command.
IDCRI01	MORSPACE	RIMC	dump	Start of procedure that scans an FDT space for a list of constants.
	NAMESCAN	RINS	dump	Start of procedure that checks data set names.
	NEEDNOTS	RINN	dump	Start of procedure that checks the input command for conflicting or missing parameters.
	PACKCVB	RIPC	dump	Start of procedure that converts a decimal constant into a binary fullword.
	POSPARM	RIPP	dump	Start of procedure that scans a positional parameter.
	SCANCMD	RISC	dump	Start of procedure that scans the input command parameters.
	SCANENDS	RISE	dump	Start of procedure that checks the input record for a continuation delimiter and determines the scanning limits of the record.
	SETDFLT	RIST	dump	Start of procedure that puts defaults in the FDT.
	SETFLAG	RISF	dump	Start of procedure that notes the occurrence of a parameter in the FDT.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
	SKIPCMD	RISK	dump	Start of procedure that bypasses the remainder of a function command.
IDCRI02	IDCRI02	RISD	dump	Start of module that prepares to scan a command parameter set.
IDCRI03	IDCRI03	RITM	dump	Start of module that performs command termination functions.
IDCRM01	CPLPROC	RMCL	dump	After the CPL has been built.
	CTLGPROC	RMZZ	dump	Before and after the UCATLG in CTLGPROC.
	DELTPROC	RMDC	dump	After the first UCATLG in DELTPROC.
		RMDD	dump	After the second UCATLG in DELTPROC.
	IDCRM01	RMFN	dump	Termination of IDCRM01.
IDCRP01	IDCRP01	RP01	dump	End of REPRO FSR.
	CATRELOD	RPD1	dump	At the end of all reload error checking before any updates have been done to the target catalog.
	CNVRTCI	RPCI	selected areas 21	On exit from procedure that translates control interval numbers on the backup catalog.
	DUMPIT	RPIO	selected areas 22	After read or write to backup or target catalog.
	RECOVCAT	RPRV	selected areas 23	After setting up locate for recoverable catalog attributes.
	TRUENAME	RPTU	selected areas 24	On exit from procedure having built truename range table.
IDCRS01	CATINIT	RSC1	dump	End of procedure that builds CIN to RRN table.
	COPYCAT	RSC2	dump	End of procedure that copies the catalog to the workfile.
	CLEANUP	RSC3	dump	Before freeing the resources used by RESETCAT.
	ENSURECI	RSE2	dump	At start of procedure prior to ensuring enough free control intervals.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
	INIT	RSI1	dump	End of procedure that initializes data area and obtains resources.
	MERGECRA	RSM1	dump	End of procedure that merges and resets CRA into the workfile.
	PROCCRA	RSP1	dump	End of procedure that merges the records of a reset CRA into the workfile.
	REASSIGN	RSR1	dump	End of procedure that assigns new control interval numbers to records on the reassign chain.
	UPDCAT	RSU1	dump	End of procedure that updates the catalog from the workfile.
	IDCRS01	RS00	dump	End of RESETCAT FSR.
IDCRS02	ASSOC	RSA1	dump	End of procedure that initiates association checking.
	PROCTYPE	RSP3	dump	After processing a set of records for associations.
	PROCCI	RSP4	dump	End of procedure that ensures that a control interval number is in the list of control interval numbers.
	SETCI	RSS2	dump	End of procedure that updates the workfile records from the associations tables.
	VERDSDIR	RSV2	dump	After verifying initial space claims.
	VERCI	RSV3	dump	After verifying associations on a set of records.
IDCRS03	GETVIA	RSG1	dump	End of procedure that locates records in the workfile.
	PROCVOL	RSP2	dump	Before freeing resources used by PROCVOL routine.
	SETBITS	RSS3	dump	At end of procedure that sets up a single bit map.
	VOLCHK	RSV1	dump	End of procedure that checks format-1 DSCBs against space headers.
	VERB	RSV4	dump	Before freeing resources used by procedure which verifies GDG data sets.
IDCRS04	DELGO	RSD3	dump	End of procedure that deletes a group occurrence.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
	FIND	RSF1	dump	End of routine that finds one or all group occurrences.
	MODGO	RSM2	dump	End of procedure that modifies a group occurrence.
IDCRS05	ADDUPCR	RSA2	dump	End of procedure that prepares for update CRA processing.
	BLDVLST	RSB1	dump	End of procedure that adds an entry to the volume serial table.
	BLDRLST	RSB2	dump	End of procedure that adds an entry to the reset volume table.
	CKERR	RSC4	dump	Before RESETCAT terminates because of an error.
	ENTNMCK	RSE1	dump	End of procedure that determines if a record has a true name.
	SCNVLST	RSS5	dump	End of procedure that locates an entry in the volume serial table.
	SCNRLST	RSS6	dump	End of procedure that locates an entry in the reset volume table.
IDCRS06	CRAMNT	RSC5	dump	End of procedure that requests CRA allocation.
	CRADMNT	RSC6	dump	End of procedure that requests CRA deallocation.
	DAOPEN	RSD1	dump	End of procedure that opens a VSAM file.
	DSCLOSE	RSD2	dump	End of procedure that closes a VSAM file.
	RECMGMT	RSR2	dump	End of procedure that performs all I/O requests.
	WFDEF	RSW2	dump	Before the UCATLG work area is freed.
	WFDEL	RSW3	dump	End of procedure that deletes the workfile.
IDCRS07	CATEOV	RSC7	dump	At conclusion of routine that extends the catalog.
	RENAMEP	RSR4	dump	Before freeing resources used by the RENAMEP procedure.
	UPDCCR	RSU2	dump	End of procedure that updates the CRAs from the workfile.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
IDCSA01	IDCSA01	SAD2	dump	During termination processing; after all loaded modules have been deleted.
IDCSA02	FINDNAME	SABB	dump	After a new load list block has been built.
IDCSA02	IDCSA02	ZZAL	dump	Before and after each time the UALLOC macro invokes dynamic allocation.
		ZZCA	dump	Before and after CATLG macro is issued to invoke catalog management routines.
		ZZCR	dump	Start of procedure that processes UCIR macro and just before the UCIR macro returns to the module that issued the UCIR.
		ZZDL	dump	Before and after the UDEALLOC macro invokes dynamic allocation.
	RELECORE	SAD4	dump	After deletion of loaded modules with zero use count.
IDCSA06	IDCSA06	SAEX	dump	End of IDCSA06 module.
		SAIN	dump	Start of IDCSA06 module.
IDCSA07	IDCSARC	RCBG	dump	Start of IDCSARC procedure.
		RCND	dump	End of IDCSARC procedure.
	UPDATENT	RCAC	selected areas 25	After calls to the catalog.
	IDCSALC	SALC	dump	Start of IDCSALC procedure.
		SALE	dump	End of IDCSALC procedure.
	IDCSAUC	SAUC	dump	Start of IDCSAUC procedure.
		SAUE	dump	End of IDCSAUC procedure.
IDCSA08	IDCSA08	SAR1	dump	After the RACHECK parameter list has been built; before the RACHECK macro is issued.
		SAR2	dump	Upon return from the RACHECK macro.
IDCSA09	CHKCODE	AA01	dump	After delayed response for mass storage control order.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
	IDCSA09	SASS	dump	Start of USSC macro.
	ISSUEMAC	AA00	dump	After issuing SVC 126.
		ZZSC	dump	Before issuing SVC 126.
IDCSA10	IDCSAST	STBG	dump	At start of module that established or cancels an ESTAE environment.
		STEN	dump	At end of module.
IDCSC01	IDCSC01	SCST	dump	Start of SETCACHE FSR.
		SCG4	dump	Before link to IDCSS04.
		SCR4	dump	After return from IDCSS04.
		SCED	dump	End of SETCACHE FSR.
IDCSS01	IDCSS01	SS01	dump	Start of IDCSS01.
IDCSS04	IDCSS04	SSST	dump	Start of IDCSS04.
		S4S2	dump	After return from IDCSS02.
		S4V1	dump	After return from IDCSS03, first unit address when a SETCACHE command has been issued.
		S4V2	dump	After return from IDCSS03, second unit address when a SETCACHE command has been issued.
		S4B3	dump	After return from IDCSS03 when a BINDDATA command has been issued.
IDCTP01	CONVERT	TP2I	dump	Start of procedure that converts data to a printable form.
		TP2N	dump	End of procedure that converts data to a printable form.
	ERROR	TPER	dump	Start of procedure that prints a text processor error message.
	IDCTPPR	TPS1	dump	After initialization of text processor parameters.
		TPIN	dump	At end of phase; the format structure for a UPRINT macro has been processed.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
	LINEPRT	TPMS	dump	On return from segment routine.
	LINEPRT	TPX1	dump	After return from procedure that puts print lines into output buffer.
		TP3I	dump	Start of procedure that formats pages and prints titles, headings, footings, and other lines requested.
		TP3N	dump	End of procedure that prints lines.
	STACKPUT	TPXX	dump	Before call to UPUT.
IDCTP04	ESTACONT	TP4A	dump	End of procedure that processes the UESTA macro.
	ESTSCONT	TP4S	dump	End of procedure that processes the UESTS macro.
	RESTCONT	TP4R	dump	End of procedure that processes the UREST macro.
IDCTP05	IDCTP05	TP5I	dump	Start of phase that loads the static text phase.
		TP5N	dump	Start of phase that loads the static text phase.
IDCTP06	IDCTP06	TPEA	dump	Start of procedure that processes UERROR macro.
		TPEB	dump	Before converted message is printed via the UPRINT macro.
IDCVS01	IDCVS01	VSBG	dump	Start of IDCVS01 module.
		VSND	dump	End of IDCVS01 module.
IDCVS02	IDCVS02	VSBG	dump	Start of IDCVS02 module.
		VSOT	dump	End of IDCVS02 module.
IDCVS03	IDCVS03	VS8G	dump	Entry to load module.
		VS3X	dump	Exit from load module.
IDCVY01	IDCVY01	VYBG	dump	Start of VERIFY FSR.
		VYND	dump	End of VERIFY FSR.
IDCXM01	IDCXM01	XM11	dump	If data set is modified during testing.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
		XM12	dump	Error obtaining TSO qualifiers.
		XM13	dump	Error obtaining volume information.
		XM14	dump	Error obtaining Format-1 DSCB.
		XM15	dump	OPEN indicator is on at OPEN.
		XM16	dump	OPEN failure.
		XM17	dump	Not key-sequenced data set (KSDS) cluster.
		XM18	dump	In create mode at OPEN.
		XM19	dump	Index component was modified during testing.
IDCXM02	IDCXM02	XM20	dump	After UGPOOL, prior to index test selection.
	SEQIXTST	XM21	dump	After UGPOOL, at entry to procedure.
	MULIXTST	XM22	dump	After UGPOOL, at entry to procedure.
		XM23	dump	After UGPOOL, at sequence set testing code portion.
		XM24	dump	After UGPOOL, at intermediate level testing code portion.
	GETIXCI	XM25	dump	After UGET, prior to internal control interval evaluation.
IDCXM02	MSGHNDLR	XM26	dump	After UPRINT, prior to exit.
IDCXM03	INDXPROC	XM30	dump	An error was returned following an attempt to read an index sequence set control interval.
		XM31	dump	An error was returned following an attempt to read a data control interval that should contain a software end-of-file.
	VALAMDSB	XM32	dump	An error detected in the sequence set access method data set statistics blcok.
	VALARDB	XM33	dump	An error detected in the sequence set address range definition block.
	GPOOL	XM35	dump	An error was returned following an attempt to obtain storage for a data control interval.

Module or CSECT to Dump Points Cross-Reference

Module or CSECT	Procedure	Dump Point	Type	Situation at Dump Point
	RDDATACI	XM36	dump	An error was returned following an attempt to read a data control interval.
IDCXP01	IDCXP01	XPFN	dump	End of EXPORT FSR, before data sets are closed and space freed.
		XP01	dump	Start of EXPORT FSR.
	CTLGPROC	XPZZ	dump	After calling the catalog to locate, alter, or delete an entry.
	DELTPROC	XPZY	dump	Just after the UCATLG macro.

Selected Areas Notes: The following list describes the selected areas printed at the specified dump points. On the printed output, the area title precedes each area dumped.

Dump Point	Area Title	Area Description
1. CCAC	CPLPTR	VSAM catalog parameter list address.
	CPL	VSAM catalog parameter list.
	FVTPTR	VSAM catalog field vector table address.
	FVT	VSAM field vector table.
	DEVTPTR	VSAM field parameter list address (device type FPL).
	DEVTFPL	VSAM field parameter list (device type FPL).
	FSEQPTR	VSAM field parameter list address (file sequence FPL).
	FSEQFPL	VSAM field parameter list (file sequence FPL).
2. CCAV	DEVTPTR	Address of device code list.
	DEVTLIST	Device code list (referenced by FPL).
	VOLPTR	Address of volume serial number list.
	VOLLIST	Volume serial number list (referenced by FVT).
	FILEPTR	Address of file sequence list.
	FILELIST	File sequence number list (referenced by FPL).

Dump Point	Area Title	Area Description
3. CCLV	ENTYPPTR	Address of catalog entry type returned by VSAM catalog.
	ENTYPLST	Catalog entry type returned by VSAM catalog.
	CATVLPTR	Address of catalog volume information returned by VSAM catalog.
	CATVLLST	Catalog volume information returned by VSAM catalog.
4. CCCN	BLKLEVEL	OS/VS catalog level index.
	BLOCKPTR	OS/VS catalog index block pointer.
	BLOCK	OS/VS catalog index block.
5. CCDC	Same as CCAC	
6. CCDV	Same as CCAV	
7. CCLC	CPLPTR	VSAM catalog parameter list address.
	CPL	VSAM catalog parameter list.
	ENTYPPTR	VSAM field parameter list address (entry type FPL).
	ENTYPFPL	VSAM field parameter list (entry type FPL).
	CATVLPTR	VSAM field parameter list address (volume information FPL).
	CATVLFPL	VSAM field parameter list (volume information FPL).
8. CCPD	Same as CCCN	
9. CCVI	Same as CCCN	
10. LCAP	FLAGS	Internal flags and switches used in IDCLC01 (refer to the microfiche listing data area, FLAGS, for mapping).
	ASSOC#	Number of associations of the catalog entry being processed.
	HOLDETYP	Type of catalog entry being processed.
11. LCCD	Same as LCAP	
12. LCOJ	CATNAME	Catalog name.
	OBJCNT	Number of entries being processed. **Note:** Multiple entries can be processed when a generic name is supplied.

Dump Point	Area Title	Area Description
	OBJTYP	Entry type currently being processed.
	OBJNM	Name of entry being processed.
	CACBFLG	Flag indicating that VSAM should return the catalog ACB pointer.
	CTGOPTN2	Second option byte indicator in CTGPL.
	FLAGS	Internal flags and switches (refer to microfiche listing data area, FLAGS, for mapping).
	TYPECNT	Number of entry types specified by the user.
	TYPELIST	Types of entry types specified by the user.
13. LCMG	ERRDARG	Text processor argument list (DARGLIST) used for printing messages.
14. LCTP	DARGLIST	Text processor argument list (DARGLIST) used for printing the catalog data.
15. LCAL	CATRC	VSAM catalog management return code.
	CTGENT	VSAM locate key (either the entry name or the CI number).
	CTGPSWD	User-supplied password.
	CTGPL	VSAM catalog management parameter list.
	CTGFL array	
	FPL(1)	VSAM field parameter list.
		Note: The number of FPLs (nn) varies with the amount of catalog information requested (i.e., NAME, HISTORY, VOL, etc.).
	MULTIFPL	VSAM field parameter list if a special function FPL is required.
16. LCBL	Same as LCAL	
17. LCLO	FLAGS	Internal flags and switches (refer to microfiche listing data area, FLAGS, for mapping).
18. LCWA	CTGWKAPT	Work area address of VSAM returned catalog fields.
	CTGWKA array	
	WKA(1) WKAEND	VSAM returned catalog fields.

Dump Point	**Area Title**	**Area Description**
		Note: This work area is dumped as an array of 256-byte blocks and the last block less than 256 bytes is indicated as WKAEND.
19. LCRA	Same as LCAP	
20. LCRP	Same as LCAP	
21. RPCI	OLDCI#	CI number of backup catalog record to be converted.
	NEWCI#	Converted CI number in the target catalog (that is, OLDCI# converted to NEWCI#).
22. RPIO	DLOUTREC	A record in the high key range of the target catalog which was deleted because it did not exist in the backup catalog.
	FUPOTREC	A record in the low key range of the target catalog which was converted to a free record because it did not exist in the backup catalog.
	INSOTREC	A record inserted into the target catalog because it existed in the backup catalog but not in the target catalog.
	UPOUTREC	A record used to update the target catalog because the same record existed in both the backup and the target catalogs.
	RDCCREC	Catalog control record of the target catalog before it was updated.
	UPCCREC	Catalog control record of the target catalog after it was updated with results of the reload operation.
	RDINPREC	A record from the backup catalog before any action is taken.
	RDOUTREC	A record from the target catalog before any action is taken.
	2ND-HALF	The second half of the record printed just above.
23. RPRV	DSATRPL	If both input and output are catalogs, the CPL used to locate the catalog DSATTR field to determine if it has the recoverable attribute. **Note:** The first printed is for the output catalog; the second is for the input catalog.
	DSATRFPL	The FPL chained off the CPL described for DSATRPL.
24. RPTU	SORSTABL	A table which maps the extents of the high key range of the backup catalog. **Note:** Each entry maps one extent and contains: Word 1—High RBA of the extent

Dump Point	Area Title	Area Description
		Word 2—Number of CIs in the extent The table is used to convert a CI number in the backup catalog to the appropriate CI number for the target catalog (see 'RPCI' above).
	TARGTABL	Same as SORSTABL for the target catalog.
25. RCAC	WKAREA	Catalog work area.

The TSO TEST Command

If you invoked access method services interactively with TSO, you can use the TSO TEST command to diagnose a problem. For a complete description of the TSO TEST command, see *TSO Command Language Reference.* In order to execute access method services with the TEST command, enter the following after logging on at your terminal:

```
TEST 'SYS1.CMDLIB

(Access method services command name)'CP
```

TSO responds with:

```
ENTER COMMAND FOR CP
```

Now, enter:

```
Access method services command with its parameters
```

TSO responds with:

```
TEST
```

Now, access method services is running under the TSO TEST command, and you can enter any TEST subcommand.

How to Use the TSO TEST Command

You must use the microfiche listings with the TSO TEST command.

If a problem occurs and you have no idea which modules are involved, run the command under TSO TEST and stop execution at the beginning of each module. Each access method services module begins with a PROL macro which calls module IDCSA03 at entry point IDCSAPR. Using the microfiche listings for IDCSA03, find the offset from the beginning of IDCSA03 to IDCSAPR. Enter:

```
AT IDCSA03.IDCSA03.+offset-to-IDCSAPRD
```

When execution stops at the beginning of each module, you can display the trace tables with the LIST subcommand. From the inter-module trace table you should be able to tell the modules involved.

If you suspect which modules are involved, rerun the access method services command under TSO TEST and stop execution at the beginning of each module involved by using an AT subcommand. Each time execution stops you can examine the trace tables and storage to further isolate the problem.

Once you know the areas within a module that have caused the problem, select a dump or trace point where you want to stop execution. Using the microfiche listing for the module, find the offset from the beginning of the module to the dump or trace point. Enter an AT subcommand for each point where you want to stop execution.

Even though you don't have a dump, refer to the section on "Reading a Dump" to find data areas in storage.

ABORT Codes

Whenever an unrecoverable error is detected by the processor, the routine that detects the error issues a UABORT macro. The system adapter then issues a write-to-programmer message (IDC4999I) with an ABORT code indicating the error and produces a snap dump if the //AMSDUMP DD card is specified. If you invoked access method services with a batched job, you supply the //AMSDUMP DD card in the JCL used to invoke access method services. If you invoked access method services interactively with TSO, you must supply the //AMSDUMP DD card in the procedure you specified in the PROC parameter of the LOGON command. For more information on the LOGON command, see *TSO Command Language Reference*. A snap dump produced through a UABORT macro has an ID = 000.

Module IDCSA01, procedure PRNTERR, detects the ABORT condition. One of the following ABORT codes is issued along with message IDC4999I.

ABORT Code	Module or CSECT	Procedure	Situation that Caused ABORT
24(18)	IDCTP01	IDCTP01	The pointer to the print control table in the GDT is not set.
	IDCTP04	RESETCON	The pointer to the print control table in the GDT is not set.
28(1C)	IDCIO01	IDCIOIT	Storage was not available for the I/O adapter historical area and message area.
	IDCIO02	BLDOCMSG	A message that sufficient storage was not available could not be issued because the SYSPRINT data set is not open.
	IDCIO04	IDCIO04	PUTLINE returned a return code greater than 4.

ABORT Code	Module or CSECT	Procedure	Situation that Caused ABORT
	IDCSA01	GETCORE	Storage was not available for the automatic storage required for IDCSA02, IDCSA03, IDCSA06, IDCSA07, IDCSA08, IDCSA09, IDCSA10, IDCIO01, IDCIO05, or IDCTP01.
	IDCSA02	IDCSA02	Storage was not available for a dynamic volume serial list or a dynamic ddname list.
	IDCSA03	GETCORE	Storage was not available for a GETMAIN request.
	IDCTP01	LINEPRT	Storage unavailable for page header line.
	IDCTP01	ERROR	Storage unavailable for saving conversion/print argument list.
	IDCTP05	IDCTP05	Storage unavailable for loading static text formatting structure.
	IDCTP04	ESTSCONT	Storage unavailable for line stack buffer.
	IDCTP04	PCTSETUP	Storage unavailable for print chain translate table.
	IDCTP04	PCTSETUP	Storage unavailable for main title line.
	IDCTP04	PCTSETUP	Storage unavailable for subtitle line.
	IDCTP04	PCTSETUP	Storage unavailable for page footing lines.
	IDCTP04	INITPCT	Storage unavailable for print control table.
32(20)	IDCIO01	IDCIOGT	The pointer to the IOCSTR is zero, or the open flag in the IOCSTR is not set, indicating that the data set to be accessed has not been opened successfully.
		IDCIOPT	The pointer to the IOCSTR is zero, or the open flag in the IOCSTR is not set, indicating that the data set to be accessed has not been opened successfully.
	IDCIO03	IDCIO03	The pointer to the IOCSTR is zero, or the open flag in the IOCSTR is not set, indicating that the data set to be accessed has not been opened successfully.
36(24)	IDCIO02	BLDOCMSG	The SYSPRINT data set could not be opened, or the SYSPRINT data set has already been closed and a message cannot be issued.
	IDCTP01	STACKPUT	Unable to transmit data to output file.
40(28)	IDCIO01	IDCIOCL	The length of the UCLOSE argument list is invalid. The length must be greater than 1 and less than 6.

ABORT Code	Module or CSECT	Procedure	Situation that Caused ABORT
		IDCIOOP	The length of the UOPEN argument list is invalid. The length must be greater than 1 and less than 6.
		IDCIOPT	The length of the UPUT argument list is invalid. The length must be greater than 1 and less than 4.
		IDCIOSI	The length of the UIOINFO argument list is invalid. The length must be greater than 3 and less than 6.
	IDCSA02	IDCSA02	The argument list of a UGSPACE, UGPOOL, UFPOOL, UALLOC, or UCIR macro is invalid.
	IDCSA05	IDCSA05	The argument list for the UTIME macro is invalid.
	IDCSA08	IDCSA08	The argument list of a USYSINFO, URACHECK, USCRATCH, or UWTO macro is invalid.
44(2C)	IDCSA02	IDCSA02	The SNAP macro was not successful. Either the DCB for the dump data set was not open or was invalid, or not enough storage was available for the dump.
56(38)	IDCSA02	IDCSA02	The BLDL macro failed to find a required access method services module.
72(48)	IDCRS05	CKERR	An internal RESETCAT error occurred. This situation should not occur in a debugged program.

You can find UABORT macros by examining the microfiche listings. The expansion of a UABORT macro for an ABORT code of 28 looks like this:

```
RESPECIFY(REG13,REG14,REG15) RSTD;
REG15 = 28;
REG14 = GDTABT;
REG13 = GDTABH;
GEN(BR    REG14);
RESPECIFY(REG13,REG14,REG15)UNRSTD;
```

Reading a Dump

This section describes how to find modules and data areas belonging to the processor in a full region dump, either a snap dump or an ABEND dump. If you are debugging under the TSO TEST command, you can use this section with the microfiche listings to find data in storage.

Snap dumps are produced by the processor on two different occasions. If the test option is on and the FULL keyword is specified, the processor produces as many snap dumps as requested, at the points requested. The IDs of these dumps start with ID = 001. If an ABORT condition occurs, the processor produces a snap dump with an ID = 000.

All executable modules and certain data areas belonging to the processor are preceded by an EBCDIC character string to identify the module or data area. Modules are preceded by the full module name, for example, IDCTP01b. (The date of compilation, in character form, follows the module name.) Data areas are preceded by a 4-byte identifier, either specific to the data area, or for the storage area in which it is built. For example, the GDT is preceded by the characters GDTȸ. The FDT is built in storage owned by the executive, and it is found in the storage areas preceded by the characters EX00.

How to Find the Module and Registers

The best way to determine which module caused the dump and to find the registers of that module varies according to the type of dump you have.

In an ABEND dump, standard methods explained in *Debugging Handbook* should be used.

In a snap dump caused by an ABORT condition, the last entry in the inter-module trace table identifies the module that issued the UABORT macro. Message IDC4999I identifies the ABORT code set in the UABORT macro. Once you know the ABORT code and the module that issued the UABORT macro, you can use the previous list of ABORT codes to determine the internal procedure that issued the UABORT macro and the situation that caused the procedure to issue the macro. The last entry in the intra-module trace table may be a trace point within the module that issued the UABORT macro.

The registers at the time that the UABORT macro was issued are not saved by the processor and cannot be found in a dump.

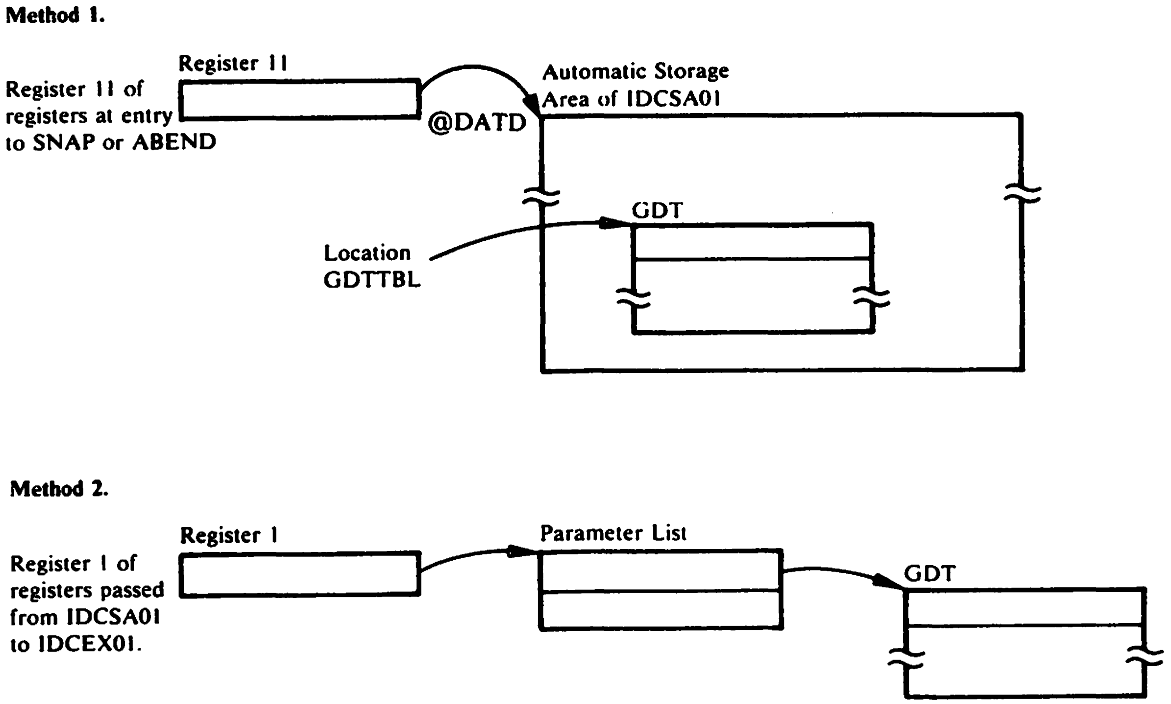
If you have a snap dump produced at a dump point, the trace tables printed along with the dump tell you at what point the dump occurred. The next to the last ID in the inter-module trace table identifies the module that issued the UDUMP macro; the last ID in the intra-module trace table identifies the exact dump point at which the dump was produced. You can use the trace tables printed along with the dump to trace the flow of control before the dump point. These trace tables are better to use for this purpose than the trace tables in the dump because the printed trace tables do not contain all the trace points encountered while producing the dump. The trace tables in the dump have been filled with dump-related trace points.

You can find the registers at the time the UDUMP macro was issued in the save area where IDCDB01 saved the caller's registers. The first word of this save area contains the characters DB01.

Figure 19, Part 1, illustrates how to find the module that caused the dump and its registers in a snap dump produced through the test option. In this example, module IDCSA02 called for a dump at the dump point 'ZZCA'. Module IDCDB01 saved the registers of module IDCSA02 in the latter's save area.

How to Find the GDT

The global data table (GDT) is preceded by the identifier GDTƀ (see Figure 19, Part 4) so you may be able to find it by scanning down the right side of the dump. A more systematic way of finding the GDT depends upon the type of dump you have. Figure 15 shows the two methods of finding the GDT and is referred to in the following paragraphs.



Method 1.

Register 11
Register 11 of registers at entry to SNAP or ABEND
@DATD
Automatic Storage Area of IDCSA01
GDT
Location GDTTBL

Method 2.

Register 1
Register 1 of registers passed from IDCSA01 to IDCEX01.
Parameter List
GDT

Figure 15. How to Find the GDT

In a snap dump produced as the result of an ABORT condition, you must use method 1 shown in Figure 15. The GDT is contained in the system adapter's (IDCSA01) automatic storage area. Register 11 of the registers at entry to SNAP points to the automatic storage area of IDCSA01. The GDT is located at offset GDTTBL in the storage area; you must examine the microfiche listing for IDCSA01 to find the offset from the start of the automatic storage area at @DATD to location GDTTBL. Add the offset of location GDTTBL to the contents of register 11 to obtain the address of the GDT.

In an ABEND dump, if the ABEND occurred after the call to IDCSA01 but before IDCSA01 calls IDCEX01, then you must again use method 1. Add the contents of register 11 of the registers at entry to ABEND to the offset of GDTTBL, to find the address of the GDT.

If the ABEND occurred after IDCSA01 called IDCEX01, use method 2 shown in Figure 15. The address of the GDT was passed as a parameter from IDCSA01 to IDCEX01. You must find the save area where IDCEX01 saved the registers belonging to IDCSA01. Register 1 in this save area contains the address of a parameter list. The first word in the parameter list contains the address of the GDT.

In a snap dump produced as a result of the test option, you can most easily find the GDT using Method 2. Find the save area where IDCEX01 saved the registers belonging to IDCSA01. Register 1 in this save area contains the address of a parameter list. The first word in the parameter list contains the address of the GDT.

The GDT is the "anchor" for all areas of the processor. In the GDT are found pointers to the trace tables, to the historical areas, and to the entry points of the system adapter, the I/O adapter, and the test processor.

Figure 19 shows the GDT as it appears in a dump. Part 1 of Figure 19 shows the registers belonging to IDCSA01 and saved by IDCEX01. Register 1 points to the parameter list. Part 4 of Figure 19 shows the parameter list and the GDT.

How to Find Save Areas

The first word of the standard save area for processor modules contains the ID of the module that saved its caller's registers in that save area. (The module ID is the last 4 characters of the module name.) For example, if the first word of the save area contains DE01, then you would know that IDCDE01 saved its caller's registers in this area. The remainder of the save area is set up following standard register saving conventions.

The save area chain normally appears in the formatted area of a dump. In addition, the start of the access method services save area chain can be found in GDTABH in the global data table (GDT). You must examine the microfiche listing of IDCSA01 to find the offset of location GDTABH.

Figure 19 shows the save areas at the start of a dump.

How to Find the Trace Tables

The trace tables can easily be found once you have found the GDT. The third word of the GDT (including the GDT identifier) points to the inter-module trace table; the fourth word of the GDT points to the intra-module trace table.

Several areas in a dump may look as if they contain the trace tables; however, these areas may simply be areas used in constructing the trace tables.

Figure 19, Part 4, shows how the trace tables appear in a dump. Note that the last (20th) trace point in the intra-module trace table is SASN; IO01 is not part of the trace table.

Note: If, in the inter-module trace table, the sequence SA02 SA02 occurs, the second SA02 is really the ID for module IDCIO02.

How to Find the FDT

You can find the function data table (FDT) for an FSR after the FSR has received control by finding the save area in which the FSR saved the registers belonging to IDCEX01. The first word of this save area contains the module ID of the FSR, for example, PR01 for the PRINT FSR. The preceding save area in the save area trace contains EX01 in the first word. Register 1 in the save area where the FSR saved registers contains the address of a parameter list. The second word of that parameter list contains the address of the FDT.

All FDTs are built by the reader/interpreter in a UGPOOL storage area obtained by the executive; the UGPOOL area has an ID of EX00. The first two words of the FDT contain the name of the command.

Figure 19 shows how an FDT looks in a dump. Part 1 of Figure 19 shows the registers belonging to IDCEX01 and saved by IDCDL01. Register 1 points to the parameter list. Part 3 of Figure 19 shows the parameter list and the FDT.

How to Find Automatic Storage Areas

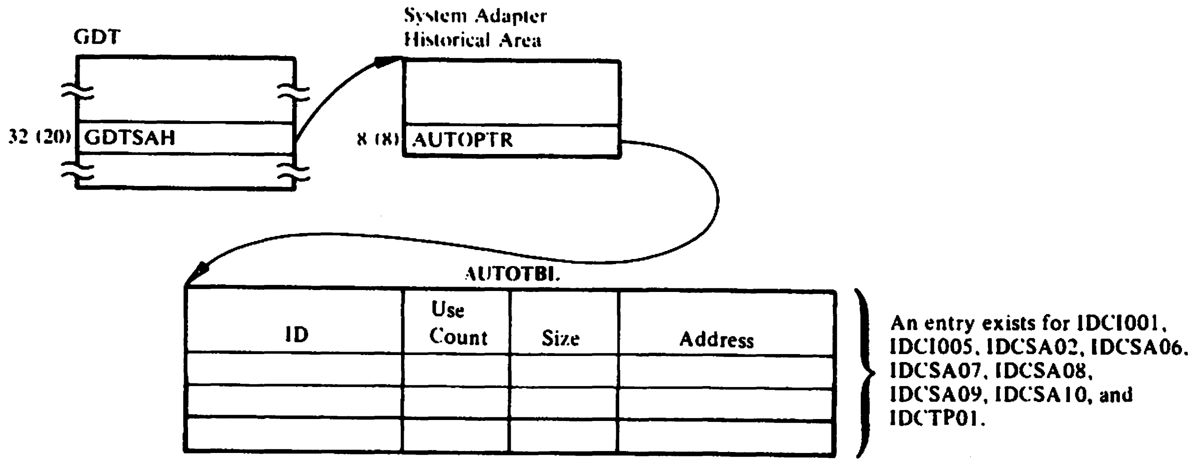
The automatic storage area for a module is that storage area obtained for the module whenever the module is entered; dynamic storage areas, on the other hand, are those storage areas obtained by the module as it is executing. All automatic storage areas, as well as dynamic storage areas, are obtained by the system adapter.

The automatic storage area for most processor modules is preceded by an 8-byte header. The first 4 bytes contain the number of bytes in the automatic storage area (including the 8-byte header), and the last 4 bytes contain the module ID. However, for commonly called modules, namely, IDCIO01, IDCSA02, IDCSA03, and IDCTP01, no header precedes the storage area, unless the module has been called recursively. On recursive calls (that is, the module has been called again within the original call), the storage area that is obtained is preceded by an 8-byte header.

The best way to find the automatic storage area for a module depends upon the module.

The address of the automatic storage area for module IDCSA03 is kept in the GDT.

The addresses of the automatic storage areas for modules IDCIO01, IDCIO05, IDCSA02, IDCSA06, IDCSA07, IDCSA08, IDCSA09, IDCSA10, and IDCTP01 are kept by the system adapter in the AUTOTBL. Figure 16 shows the format of the AUTOTBL and how to find it. However, if any of these modules have been called recursively, indicated by a use count in the AUTOTBL greater than one, another automatic storage area has been obtained. You must find the second and third storage areas using the module's data register or save area register as explained in the next paragraphs.


GDT
System Adapter
Historical Area
32 (20)
GDTSAH
8 (8)
AUTOPTR
AUTOTBL
ID
Use Count
Size
Address
An entry exists for IDCI001, IDCI005, IDCSA02, IDCSA06, IDCSA07, IDCSA08, IDCSA09, IDCSA10, and IDCTP01.

Field	*Contents*
ID	Module ID, 'IO01', 'IO05', 'SA02', 'SA06', 'SA07', 'SA08', 'SA09', 'SA10' or 'TP01'
Use Count	Number of automatic storage areas obtained for the module: 0 – no storage area being used 1 – the storage area whose address is in this table is the only storage area being used >1 – another storage area has been obtained for the module
Size	Number of bytes in automatic storage area
Address	Address of automatic storage area

Figure 16. Format of AUTOTBL

Part 4 of Figure 19 on page 350, shows how the system adapter historical area and AUTOTBL appear in a dump.

To find the automatic storage area for any module, you can examine the microfiche listings to find which register has been used by the compiler as the data register. This register, usually register 11, points to the automatic storage area.

For all processor modules, the first item in the automatic storage area is the save area. Thus, you can also use register 13, which contains the address of the save area, to find the automatic storage area belonging to that module.

Figure 17 shows the automatic storage area for module IDCEX01. Module IDCEX01 has called IDCDL01; therefore, module IDCDL01 has saved the registers belonging to IDCEX01 in the save area.

Part 2 of Figure 19 on page 350, shows an automatic storage area as seen in a dump.

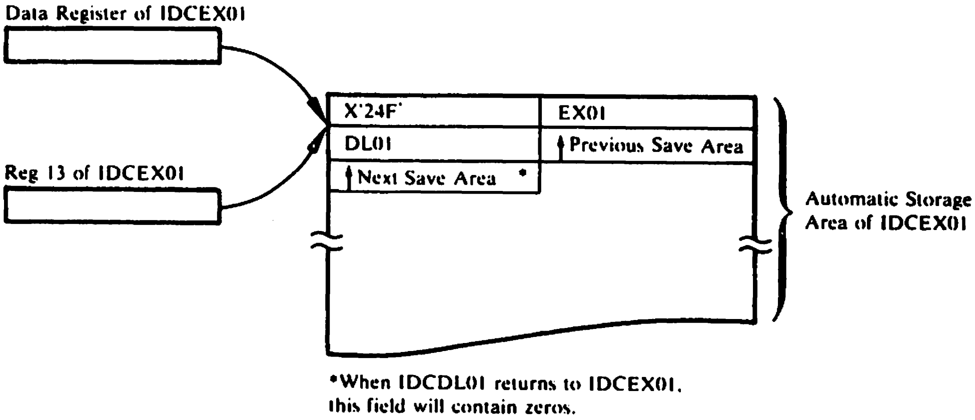

Data Register of IDCEX01
Reg 13 of IDCEX01
X'24F'
EX01
DL01
Previous Save Area
Next Save Area *
Automatic Storage Area of IDCEX01
*When IDCDL01 returns to IDCEX01, this field will contain zeros.

Figure 17. Example of an Automatic Storage Area

How to Find Dynamic Storage Areas

The dynamic storage area is that area obtained by the module as it is necessary; the automatic storage area, on the other hand, is that storage area obtained for the module whenever the module is entered. All dynamic storage areas, as well as all automatic storage areas, are obtained by the system adapter. A module obtains storage areas dynamically by issuing either a UGSPACE or a UGPOOL macro.

To find a storage area obtained via a UGSPACE macro, you must examine the microfiche listings to see where the module has saved the address of that particular storage area. To find a storage area obtained via a UGPOOL macro, you can again examine the microfiche listings or you can follow the UGPOOL storage chain maintained by the system adapter.

Figure 18 shows how to find the chain of UGPOOL areas from the system adapter's historical area.

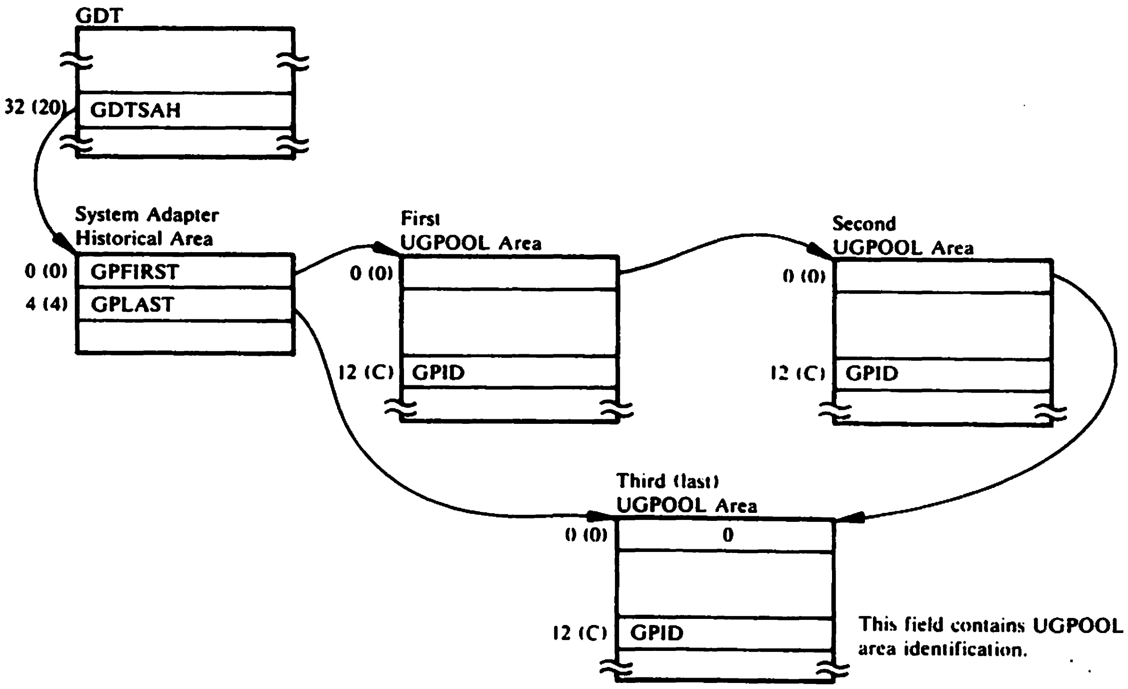


Figure 18. UGPOOL Area Chain

Contents of UGPOOL Areas

The following list contains the UGPOOL IDs used by different modules when they obtain storage. The list also contains the name of the internal procedure that issues the UGPOOL macro, and the contents stored in the UGPOOL area.

Figure 19 shows the UGPOOL chain as it appears in a dump. Part 4 of Figure 19 on page 350 shows the start of the chain in the GDT. Part 3 shows the chain.

Module	UGPOOL ID	Procedure	Contents of UGPOOL Area
IDCAL01	AL00	ALTERPRC	One of the following: PASSWALL field or volume list.
		IDCAL01	CTGPL, CTGFV, and CTGFLs. Catalog entry names returned from a generic locate. UGPOOL area is obtained by IDCSA02(UCIR).
		INDEXPRC	CTGPL, CTGFV, CTGFL, and IDAAMDSB field.
		LOCATPRC	Catalog work area for locate requests.

Module	UGPOOL ID	Procedure	Contents of UGPOOL Area
		DALCPROC	Volume list and ALLAGL parameter list.
IDCBI01	BI01	JCPROC	Area obtained by UIOINFO to contain sort work file data set name and volume serial list; passed back to JCPROC.
	BIPG	INITPROC	One 2048 byte buffer, followed by area for define CTGPL, CTGFVs, and CTGFLs followed by alternate index record output buffer; area starts on page boundary.
	BIPG	INITPROC	Record sort area followed by table which controls the sort.
IDCCC01	CC00	CATALOG	A new VSAM catalog management work area if VSAM return code indicates insufficient space.
		CNVTINIT	OS/VS SYSCTLG index blocks for 22 index levels.
		INITPROC	VSAM parameter lists, data addressed by the VSAM parameter lists, VSAM catalog management work area.
IDCCC02	CC00	ASSOCTNS	Associations save area.
		CTLGPROC	A new VSAM catalog management work area if VSAM return code indicates insufficient space.
		DALCPROC	Volume serial list.
		DELSPACE	Volume serial list.
		LVLRPROC	Volume list and device type list.
		RANGPROC	Keyrange list.
		XASSOCNS	Associations save area.
IDCCK01	CK00	BUILDTAB	Area where table of checkpoint IDs is built.
IDCDE01	DE00	DALCPROC	List of volumes to be mounted and ALLAGL parameter list.
		IDCDE01	CTGPL, and CTGFVs.
IDCDE02	DE00	ALLCPROC	One of the following: volume list, file sequence list, device type list, or CTGFLs (DSATTR, LRECL, BUFSIZE, SPACPARM, DEVTYP, FILESEQ).
		KEYPROC	One or both of the following: AMDSBCAT CTGFL and IDAAMDSB field, or key range list.

Module	UGPOOL ID	Procedure	Contents of UGPOOL Area
		MODELPRC	One of the following: CTGPL and CTGFLs used to locate a model object, or catalog locate work area.
		NAMEPROC	Creation, expiration date, and exception exit CTGFLs and GDG attributes.
		PROTPROC	PASSWALL CTGFL, OWNERID CTGFL, PASSWALL field, user authorization record, RGATTR CTGFL and RGATTR field.
IDCDL01	DL00	DCDL01	Catalog entry names returned from a generic locate. UGPOOL area is obtained by IDCSA02(UCIR).
	DL01	MORESP	Larger VSAM catalog management services work area if necessary.
IDCIO01	IO00	IDCIOCR	Areas for the cryptographic macros parameter list, header area for enciphered data set, installation data buffer, cipher buffer, cipher length array, and cipher relative number array. The UGPOOL ID is supplied by the caller.
		IDCIOIT	I/O adapter historical area and VSAM message area.
	IO*nn*	PUTREP	Work area where VSAM moves records during GET. The UGPOOL ID is the same as the ID for the associated IOCSTR.
IDCIO02	10*nn*	BUILDACB	ACB, RPL, EXLST for a VSAM data set. The UGPOOL ID is the same ID as the associated IOCSTR.
IDCIO02	IO*nn*	BUILDDBK	JFCB and exit list if OPENJ processing; DCB; DECB if BSAM update processing. The UGPOOL ID is the same as the ID for the associated IOCSTR.
		BUILDRPL	Work area where VSAM moves records during GET. The UGPOOL ID is the same as the ID for the associated IOCSTR.
		CKNONOP	Work area used to assemble an ISAM data set with fixed records and nonembedded keys. The UGPOOL ID is the same as the ID for the associated IOCSTR.
		OPENRTN	IOCS prefix, IOCSTR, IOCSEX, and data set name. Each data set that is opened is assigned a unique UGPOOL ID, starting with IO01; the next data set that is opened is assigned an ID of IO02. All areas associated with a data set have the save UGPOOL IDs.

Module	UGPOOL ID	Procedure	Contents of UGPOOL Area
IDCIO03		DSINFO	Area in which data set name, volume serial numbers, device type, and/or format-4 time stamp is returned to the caller if an area is not supplied by the caller. The UGPOOL ID is supplied by the caller.
IDCLC01	LC00	INITPROC	Main CTGPL used for all locate requests except when locating the entry names of associated entries. This area also contains a save area for the CTGPL.
	LC01	INITPROC	All CTGFLs, followed by the CTGFL save area.
	LC02	INITPROC	Catalog work area referenced by the main CTGPL.
	LC03	INITPROC	CTGPL used to locate entry names of associated entries. This area also contains a save area for the CTGPL.
	LC04	INITPROC	Catalog work area referenced by the CTGPL used to locate entry names of associated entries.
	LC05	INITPROC	String of control interval numbers and types of associated entries for cluster, AIX, GDG and pagespace.
	LC06	INITPROC	Text processor argument list.
	LC07	INITPROC	Abbreviations used in catalog listing, loaded from static text module.
	LC10	ENTPROC	Catalog entry names returned from a generic locate. UGPOOL area is obtained by IDCSA02 (UCIR).
	LC11	INITPROC	String of control interval numbers and types of associated entries for data, index, path, and non-VSAM.
IDCLC02	LC08	LOCPROC	Larger catalog work area. UGPOOL LC02 is released.
	LC09	CDIPROC	Larger area for string of control interval numbers and types of associated entries. UGPOOL LC05 or LC11 is released.
IDCMP01	MP01	BFPLPROC	CTGFL.
		BPASPROC	PASSWALL CTGFL.
		CLUSPROC	Buffer to read data records from the portable data set.
		CTLGPROC	Larger catalog work area.
		DALCPROC	List of volumes to be mounted.

Module	UGPOOL ID	Procedure	Contents of UGPOOL Area
		DELTPROC	Larger VSAM catalog management services work area if necessary.
		LVLRPROC	One of the following: volume list for define, or DEVTYPE CTGFL.
		RANGPROC	Range list.
		FVTPROC	FVT and pointers to FPLs.
IDCRC02	RC02	IDCRC02	Output buffer for header record.
		CLUSPROC	Output buffer for control record.
		LOCPROC	Buffer for FPLs and catalog work area.
		CTLGPROC	Buffer for larger catalog work area.
		NVSMPROC	Output buffer for control record.
		SAVEPROC	Buffer for catalog control records.
		GDGPROC	Output buffer for control record.
		ASOCPROC	Output buffer for control record.
IDCRI01	EX00	GETSPACE	FDT—data substructures.
		MORSPACE	FDT—data list substructures.
		SCANCMD	FDT—secondary pointer vectors.
	RI*nn*	INREPEAT	FDT—temporary space for secondary pointer vectors. *nn* is the ID of the parameter associated with the secondary pointer vector.
IDCRI02	EX00	IDCRI02	One of the following: reader/interpreter tables or FDT.
	RI*nn*	IDCRI02	FDT—temporary space for secondary pointer vectors. *nn* is the ID of the parameter associated with the secondary pointer vector.
IDCRI04	EX00	GETSPACE	FDT data substructures.
		REPLIST	FDT secondary address vectors.
		RISETUP	FDT primary address vector and secondary count vectors.

Module	UGPOOL ID	Procedure	Contents of UGPOOL Area
		SETDFLT	Default parameter work area.
		SUBSCAN	Work area for input command.
	RI04	REPLIST	Additional space for tables used only by the reader/interpreter.
		RISETUP	Tables used only by the reader/interpreter.
IDCRM01	RM01	ALISPROC	Catalog data record buffer.
		BFPLPROC	Obtain one or two FPLs.
		BPASPROC	Contain PASSWALL field information.
		CLUSPROC	Buffer area for data record containing catalog locate area. Also volume list.
		CPLPROC	Catalog parameter list.
		CTLGPROC	Larger catalog work area.
		DALCPROC	List of volumes to be mounted.
		DELTPROC	Larger catalog work area.
		FVTPROC	FVT and pointers to FPLs.
		GDGPROC	Storage for data record.
		LVLRPROC	Volume serial list. DEVTYP FPL and associated device type lists. List of FILESEQUENCE numbers and associated FPL.
		NFVTPROC	FVT and total number of FPLs.
		NVSMPROC	Buffer for data record.
		RANGPROC	Storage for range list.
		UCATPROC	Storage for data record.
IDCRP01	RP00	MERGEICF COPYICF	Catalog work areas.
	RP01	IDCRP01	SHIP keynames list.
	RP03	GETNAM	UCIR work area.

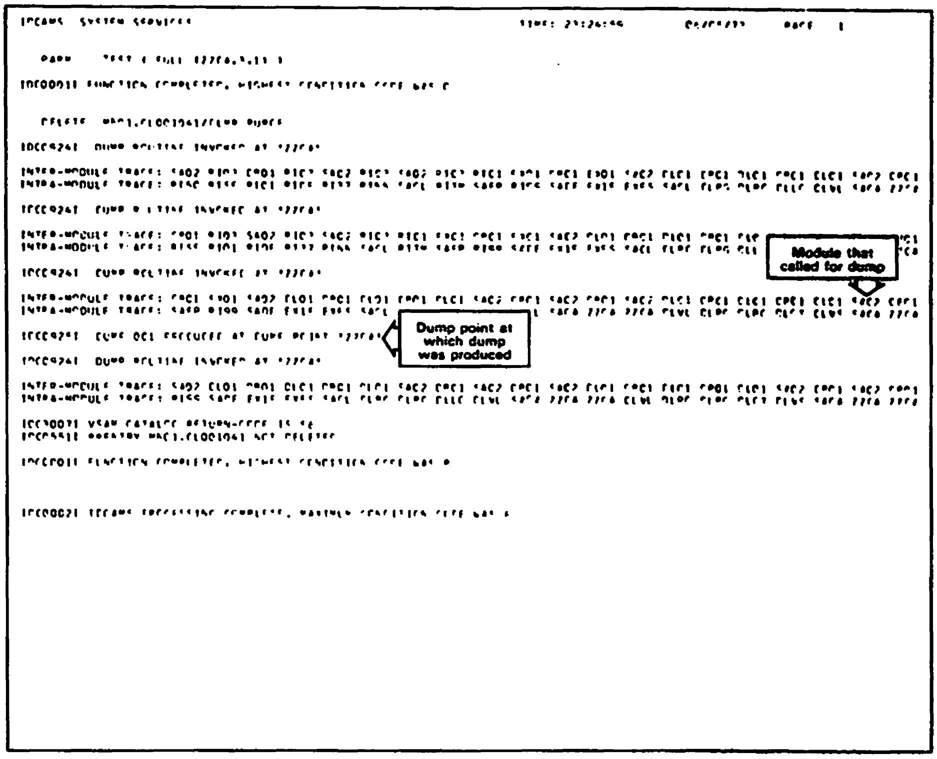
Module	UGPOOL ID	Procedure	Contents of UGPOOL Area
IDCRS01	RS01	IDCRS01	Automatic storage for modules IDCV IDCRS01 through IDCRS07.
	RS01	INIT	Work area used for Umacro parameter lists, record access blocks, and control interval translate table.
	RS03	INIT	Area obtained by UIOINFO for catalog data set information.
IDCRS03	RS10	GETTAB	Tables obtained as needed for association checking.
	RS11	PROCVOL	Work areas used for bit maps.
	RS12	VERB	Work area used for GDG association checking.
IDCRS04	RS04	NINIT	Work area used for FIND processing.
	RS04	NXPND	Extension to FIND work area.
IDCRS05	RS01	BLDRLST	RESVOL table.
	RS02	BLDVLST	VOLSERTB table.
IDCRS06	RS03	WFDEF	Work area used for UCATLG parameter list to define work file, area obtained by UIOINFO for work file data set information.
IDCRS07	RS03	RENMSETV	CAMLST for RENAME.
	RS03	SCRATCHP	Volume list for SCRATCH.
IDCTP01	TP03	LINEPRT	Header line.
IDCTP04	TP01	INITPROC	Secondary print control table.
		PCTSETUP	One of the following: print control table, subtitle lines, or footing lines.
IDCTP05	TP01	IDCTP05	Entry from a static text format structure.
IDCXP01	XP01	ALTRPROC	CTGFV and CTGFLs for catalog alter request.
		CLUSPROC	Output buffer for control records.
		CTLGPROC	Larger catalog work area.
		DELTPROC	CTGPL for catalog delete request.
		LOCPROC	One of the following: CTGPL and CTGFLs for catalog locate request, or catalog work area for locate request.

Sample Dump

The dump displayed in Figure 19 was obtained through the test option at the ZZCA dump point. The PARM command was specified as follows:

```
PARM TEST( FULL( ZZCA,3,1 ) )
```

Various fields within the dump are marked; these fields are discussed more fully in this chapter.


Module that called for dump
Dump point at which dump was produced

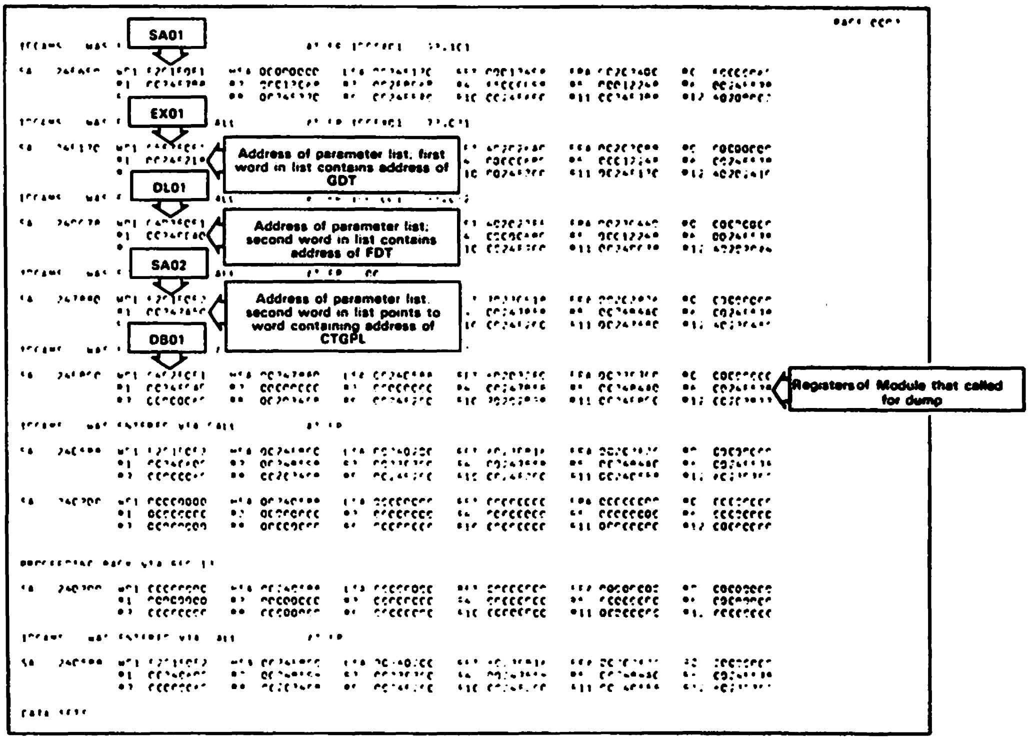
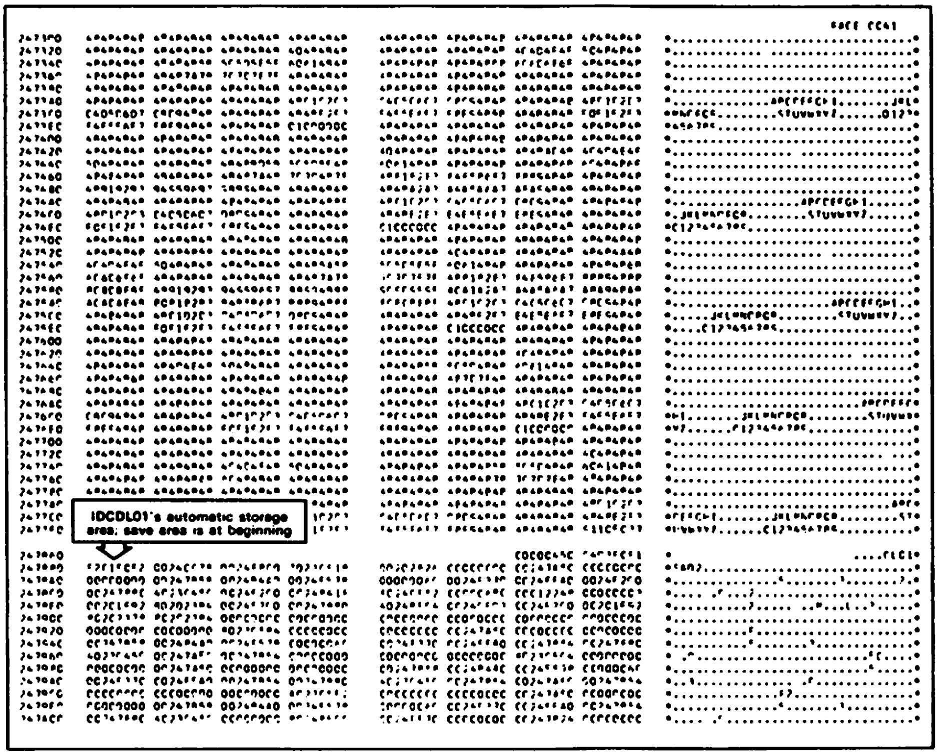
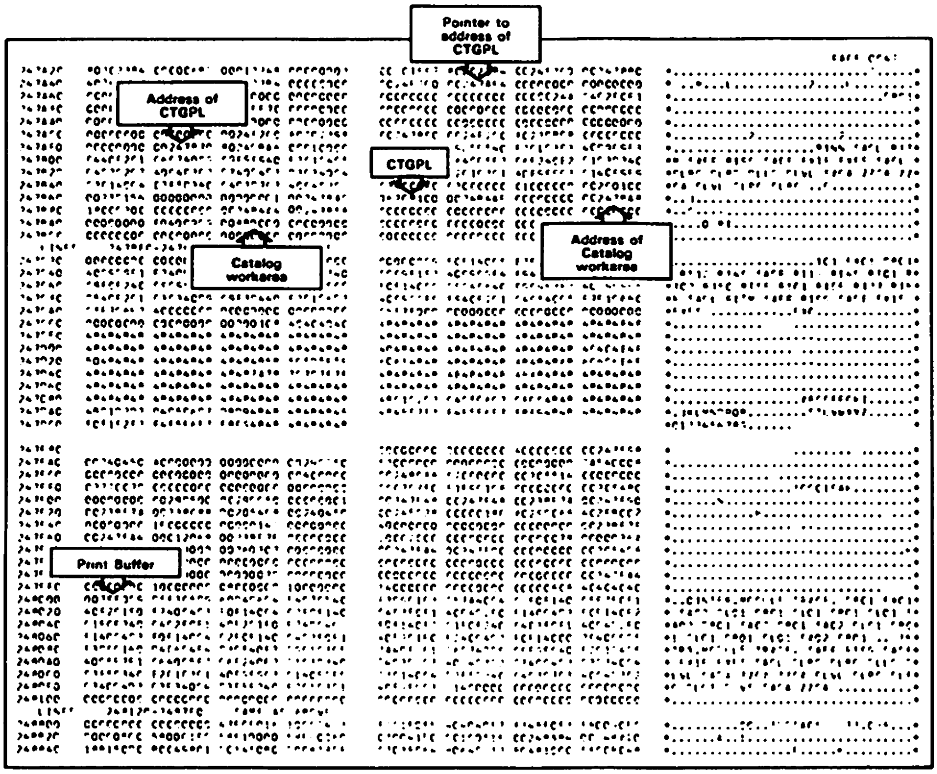

SA01
EX01
Address of parameter list, first word in list contains address of GDT
DL01
Address of parameter list; second word in list contains address of FDT
SA02
Address of parameter list, second word in list points to word containing address of CTGPL
DB01
Registers of Module that called for dump

Figure 19 (Part 1 of 4). Sample Dump



IDCDL01's automatic storage area; save area is at beginning



Pointer to address of CTGPL

Address of CTGPL

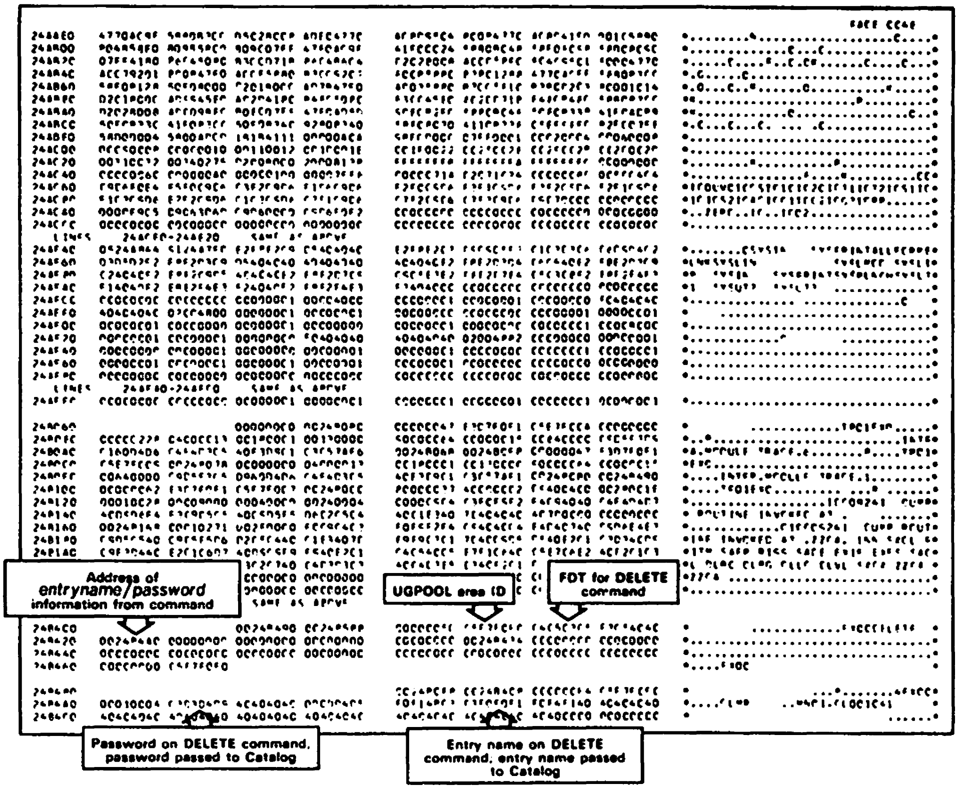
CTGPL

Catalog workarea

Address of Catalog workarea

Print Buffer

Figure 19 (Part 2 of 4). Sample Dump


Address of entryname/password information from command
UGPOOL area ID
FDT for DELETE command
Password on DELETE command, password passed to Catalog
Entry name on DELETE command; entry name passed to Catalog

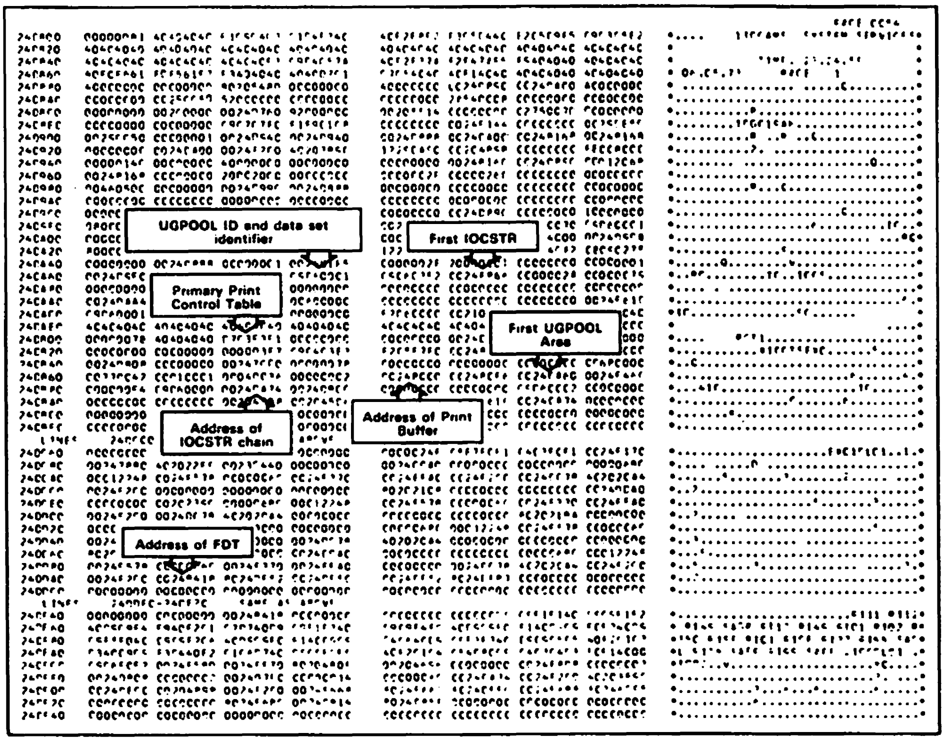

UGPOOL ID and data set identifier
First IOCSTR
Primary Print Control Table
First UGPOOL Area
Address of Print Buffer
Address of IOCSTR chain
Address of FDT

Figure 19 (Part 3 of 4). Sample Dump

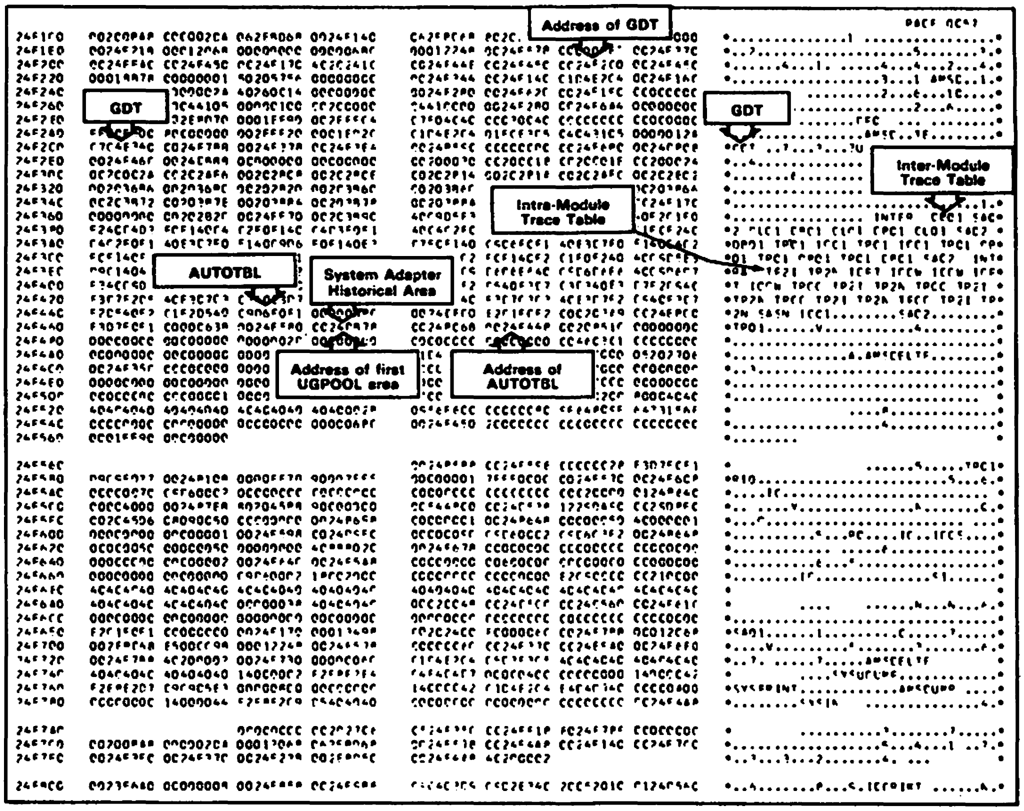

Address of GDT
GDT
GDT
Inter-Module Trace Table
Intra-Module Trace Table
AUTOTBL
System Adapter Historical Area
Address of first UGPOOL area
Address of AUTOTBL

Figure 19 (Part 4 of 4). Sample Dump

Debugging a Catalog Problem

There may be a problem within catalog management routines or within access method services routines that invoke catalog management if one of the following situations occurs: an ABEND occurs within catalog management routines, the return code from the catalog indicates a nonuser error, or the printed output from the catalog is incorrect. To determine whether the problem exists in access method services or in catalog management, you must examine the argument lists passed between the processor and Catalog Management. If the catalog in question is an ICF catalog, you may wish to use the DIAGNOSE command to look for catalog or related VSAM volume data set (VVDS) errors (see "Using DIAGNOSE for BCS or VVDS Problems" in this chapter).

This section explains how to obtain a dump that contains the catalog management argument lists and how to find the argument lists within the dump.

To determine whether the argument lists passed between the processor and catalog management are correct, see "Method of Operation" and *VSAM Logic*. "Method of Operation" explains what argument lists are passed to catalog management by each FSR. *VSAM Logic* explains the contents of the argument lists and also explains the arguments that are returned by catalog management.

Obtaining a Dump for a Catalog Problem

If you do not have an ABEND dump within catalog management, you can use the Test option to obtain a dump within access method services before and after the call to catalog management.

The "Module or CSECT to Dump Points Cross-Reference" list contains all the dump points within the processor; you can specify these dump points on the FULL option of the TEST keyword to obtain a full region dump. Each FSR that issues a UCATLG macro to call catalog management has dump points before and after each macro. In addition, the system adapter module that issues the catalog management SVC, SVC 26, has a dump point before and after the SVC.

Some FSRs have unique dump points around different types of calls to catalog management. For example, IDCDL01 has dump points DLVL around the call to locate the entry type and dump points DLVS around the call to delete the entry. Some FSRs have the same dump point around all calls to catalog management, for example, IDCMP01. Some FSRs have dump points at which you can obtain selected fields in addition to a full region dump, for example, dump points LCBL and LCAL in IDCLC01.

The system adapter dump point ZZCA can always be used, for any FSR, to obtain dumps before and after a call to catalog management.

To determine at which iterations of a dump point you wish a full region dump, you must determine how many calls to catalog management have been made by the FSR before the call that caused the problem. You can either refer to "Sequence of Calls Made by FSR" or rerun the job with the AREAS option.

"Sequence of Calls Made by FSR" summarizes the sequence of calls each FSR makes to catalog management. Using this summary, assume that the LISTCAT FSR, IDCLC01, while listing all the information for an index cluster entry, listed the cluster name under the index entry incorrectly. You would know that the call to the catalog that retrieved that name was the seventh call the LISTCAT FSR made to catalog management.

Instead of using this summary, you can rerun the job with the AREAS option of the TEST keyword to determine which iteration of a dump point you need to use.

For example, if you wish to use dump point ZZCA to obtain a dump, rerun the job with the following test option:

```
  PARM TEST( AREAS( ZZ ) )
```

From the trace output you can see how many times dump point ZZCA was encountered before the problem occurred.

How to Find Catalog Management Argument Lists

The catalog parameter list (CTGPL) is the one argument list always passed between access method services and catalog management. The CTGPL may point to a catalog work area, a CTGFV, or one or more CTGFLs. Thus, once you find the CTGPL, you can find all the catalog management argument lists.

The best way to find the CTGPL in a dump depends upon the type of dump you have: an ABEND dump within catalog management, a snap dump taken at a dump point within an FSR, or a snap dump taken at the ZZCA dump point in the system adapter.

In an ABEND dump within catalog management, register 1 of the registers saved when SVC 26 was entered contains the address of the CTGPL.

In a snap dump taken at a dump point within an FSR, the address of the CTGPL is stored at location CTGPLPTR in the FSR's automatic storage area. You must examine the microfiche listings to determine the offset of location CTGPLPTR in the automatic storage area.

In a snap dump taken at dump point ZZCA within the system adapter, the address of the CTGPL is again stored at location CTGPLPTR in the FSR's automatic storage area. However, the address of the CTGPL is also passed as an argument from the FSR to IDCSA02 when the UCATLG macro is issued.

Figure 20 shows how to find the address of the CTGPL using register 1 at entry to IDCSA02. Register 1 contains the address of a parameter list. The second word of the parameter list points to a full word that contains the address of the CTGPL.

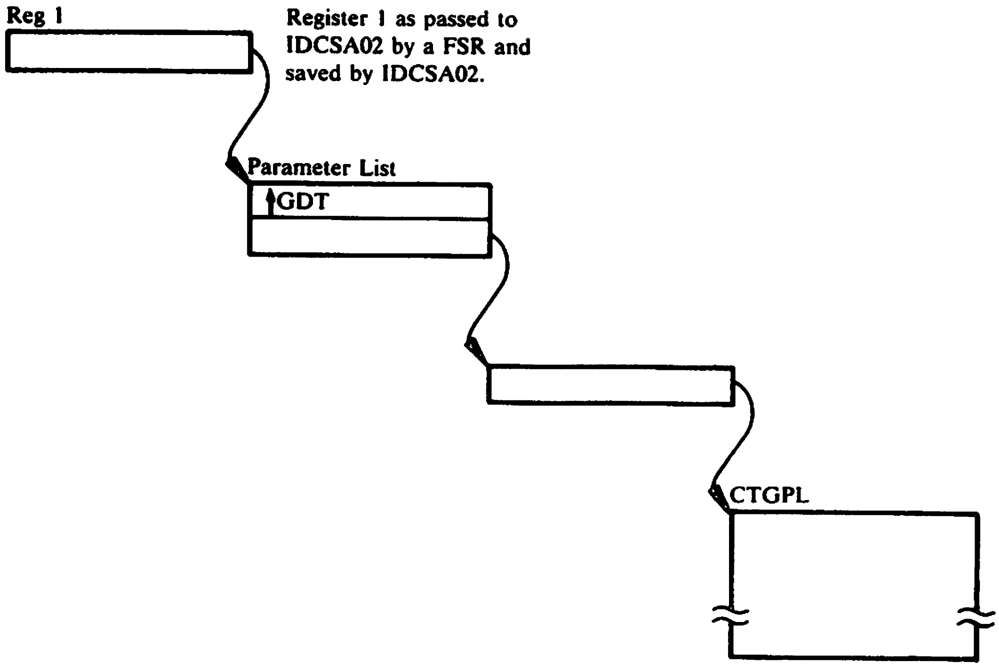

Reg 1
Register 1 as passed to IDCSA02 by a FSR and saved by IDCSA02.
Parameter List
↑GDT
CTGPL

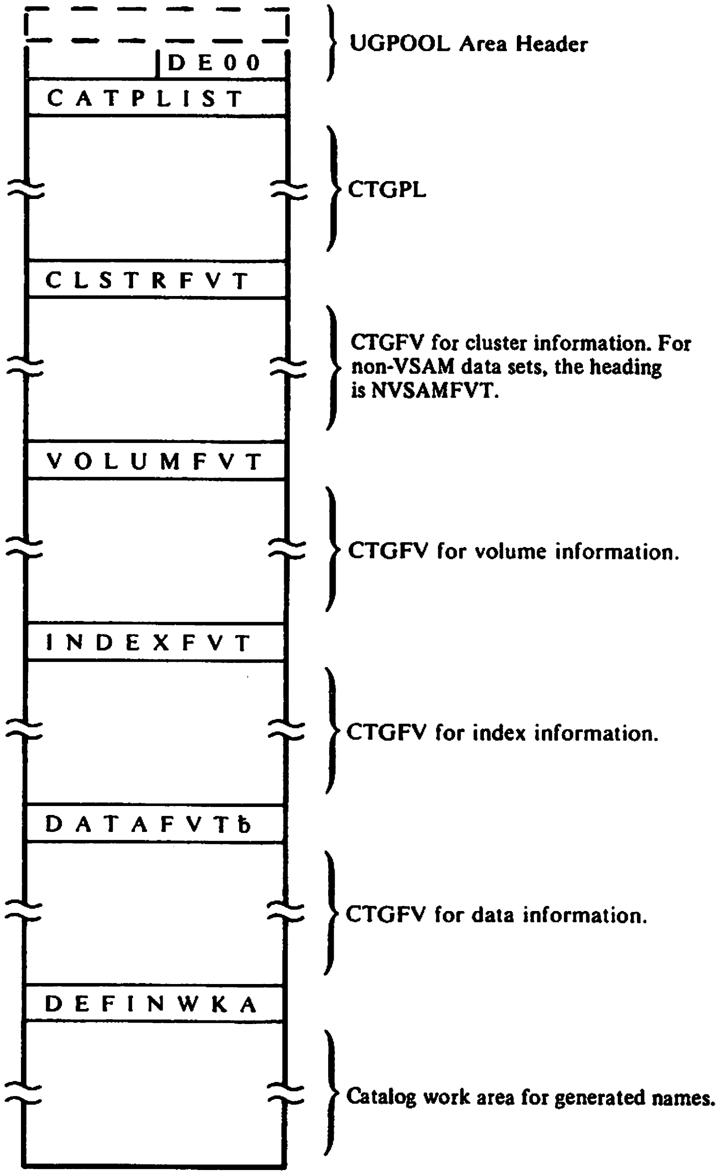
Figure 20. How to Find the CTGPL

In addition to the CTGPL, catalog management returns to the processor a code in register 15 that indicates the result of the catalog request. The best way to find the return code in a dump again depends upon the type of dump you have: a snap dump taken at a dump point within an FSR, or a snap dump taken at dump point ZZCA.

In a snap dump taken at a dump point within an FSR, you must examine the microfiche listings to determine where the FSR has stored the return code. However, any nonzero return code is always printed by the FSR in a subsequent message.

In a snap dump taken at a dump point within the system adapter, the catalog return code is stored at location TESTRC in IDCSA02's automatic storage area. You must examine the microfiche listings to determine the offset of TESTRC in the automatic storage area.

Some FSRs have headings before the storage areas that contain the catalog management argument lists. These headings may help you find the catalog management argument lists in a dump. Figure 21 shows the DEFINE FSR's storage area that contains the argument lists set up for a define request.


DE00
UGPOOL Area Header
CATPLIST
CTGPL
CLSTRFVT
CTGFV for cluster information. For non-VSAM data sets, the heading is NVSAMFVT.
VOLUMFVT
CTGFV for volume information.
INDEXFVT
CTGFV for index information.
DATAFVTƀ
CTGFV for data information.
DEFINWKA
Catalog work area for generated names.

If any of the above CTGFVs are not set up for a define request, the heading and CTGFV area contains zeros.

Figure 21. Catalog Argument Lists in Storage Area of DEFINE FSR

Sequence of Catalog Calls Made by FSR

The following table summarizes the sequence of calls each FSR makes to catalog management. **Sequence of Catalog Calls Made by FSRs**

FSR **Sequence of Calls to Catalog Management**

IDCAL01

1. A call to locate catalog fields if one of the following fields is being nullified or altered: MASTERPW, CONTROLPW, UPDATEPW, READPW, CODE, ATTEMPTS, AUTHORIZATION, ERASE | NOERASE, SHAREOPTIONS, FREESPACE, WRITECHECK | NOWRITECHECK, UNINHIBIT | INHIBIT, UPGRADE, UNIQUEKEY, NONUNIQUEKEY, KEYS, or RECORDSIZE, SCRATCH | NOSCRATCH, EMPTY | NOEMPTY, ADDVOLUMES | REMOVEVOLUMES, STAGE | BIND | CYLINDERFAULT, DESTAGEWAIT | NO.

If UPGRADE was supplied:

1. A call to locate the associated data component of the alternate index to verify that it is empty.

2. A call to alter the alternate index entry.

If RECORDSIZE was supplied for the data object:

1. A call to locate the cluster or alternate index associated with the data object.

2. A call to locate the index associated with the cluster or alternate index related to the data object.

3. A call to alter the data entry.

If RECORDSIZE was supplied for the cluster or alternate index object:

1. A call to locate the associated data object.

2. A call to locate the associated index object.

3. A call to alter the data entry.

If RECORDSIZE was supplied for the path object:

1. A call to locate the data object of the related alternate index or cluster.

2. A call to locate the index object of the related alternate index cluster, or cluster.

3. A call to alter the data entry.

If KEYS was supplied for the data object:

1. A call to locate the cluster or alternate index associated with the data object.

2. A call to locate the index associated with the cluster or alternate index related to the data object.

3. A call to locate the alternate index's base cluster, if the data object is associated with an alternate index.

4. A call to locate the data object of the base cluster.

5. A call to alter the data entry.

6. A call to alter the related index object key values.

If KEYS was supplied for the cluster object:

1. A call to locate the associated data object.

2. A call to locate the associated index object.

3. A call to alter the data entry.

4. A call to alter the related index object key values.

If KEYS was supplied for the alternate index object:

1. A call to locate the associated data object.

2. A call to locate the associated index object.

3. A call to locate the base cluster object.

4. A call to locate the base cluster's data object.

5. A call to alter the data entry.

6. A call to alter the related index object key values.

If KEYS was supplied for the path object:

1. A call to locate the data object of the related alternate index or cluster.

2. A call to locate the index object of the related alternate index or cluster.

3. A call to locate the base cluster's data object, if the path is related to an alternate index.

4. A call to alter the related entry's data object.

5. A call to alter the related index object's key values.

IDCBI01

1. A call to locate the catalog ACB, entry type, and associations of the name specified for the base cluster—may be the base cluster itself or a path over the base cluster.
2. A call to locate the AMDSB of the base cluster's data component.
3. A call to locate the entry type and associations of the name specified for the alternate index—may be the alternate index itself or a path over the alternate index.
4. If locate 3 returned a path over the alternate index, a call to locate the entry type and associations of the alternate index.
5. A call to locate the AMDSB of the alternate index's data component.

If an external sort is performed:

1. Two calls to define each sort work file.
2. Two calls to delete each sort work file.

IDCCC01

For each OS/VS catalog entry that is to be moved to the VSAM or ICF catalog a call to:

1. Define the data set or alias.
2. Locate information if the name of the OS/VS data set already exists in the target catalog, and the OS/VS data set name does not begin with 'SYS1.'
3. Alter the VSAM catalog entry, if necessary, to point to the OS/VS data set if the duplicate name in the VSAM catalog is not a VSAM data set, and the conditions under 2. are satisfied.

For each VSAM catalog entry that is to be moved to the ICF catalog, a call to:

1. Locate the base sphere entry, which could be non-VSAM, cluster, GDG base, or user catalog connector.
2. Locate the data entry, only for the cluster base object.
3. Locate the index entry, only for the KSDS cluster base object.
4. Define VVDS on each volume containing a cluster.
5. Delete data space from each volume containing a cluster.

6. Define base sphere into the target catalog.

7. Locate each association to base object.

8. Define each association into the target catalog.

9. Delete base sphere from the target catalog if an error occurred in processing its association.

IDCDE01

1. Define the entire entry.

IDCDE02

1. Locate each object that is modeled, as follows: three calls if the MODEL keyword is specified in the cluster parameter list for a KSDS cluster or in the user catalog parameter list; two calls if the MODEL keyword is specified in the cluster parameter list for an ESDS cluster or in both the data and index parameter lists; one call if the MODEL keyword is specified in a data parameter list or an index parameter list only.

IDCDL01 For each entry:

1. Locate the entry type, if the type was not specified on the command.

2. Delete the entire entry.

3. An iterative series of calls to delete any remaining parts of a structure as necessary.

IDCLC01 For each alias entry:

1. Locate the alias entry.

2. Locate the true entry.

For each cluster entry:

1. A call to locate the cluster entry.

2. *A call to locate the name of the data entry associated with the cluster entry.

3. *A call to locate the name of the index entry associated with the cluster entry, only for KSDS clusters.

4. *Repetitive calls to locate the names of the alternate indexes and paths associated with the cluster entry (if any exist).

5. A call to locate the data entry.

6. *A call to locate the name of the cluster entry associated with the data entry.

7. A call to locate the index entry, only for KSDS clusters.

8. *A call to locate the name of the cluster entry associated with the index entry.

9. Repetitive calls to locate the path entries (if any exist).

10. *Repetitive calls to locate the cluster, data, and index (for key-sequenced files) associated with the path entries.

For each data entry:

1. Locate the data entry.

2. *Locate the name of the cluster or alternate index entry associated with the data entry.

For each generation data group entry:

1. Locate the generation data group entry.

2. *Locate the non-VSAM generation index data set names.

3. Locate each generation index data set.

4. *Locate the generation data group names associated with the non-VSAM entry.

*Applies to VSAM catalogs only.

For each index entry:

1. Locate the index entry.

2. *Locate the name of the cluster or alternate index entry associated with the index entry.

IDCLCO1 For each alternate index entry:

1. *A call to locate the alternate index entry.

2. *A call to locate the name of the data entry associated with the alternate index entry.

3. *A call to locate the name of the index entry associated with the alternate index entry.

4. *A call to locate the name of the cluster entry associated with the alternate index entry.

5. *Repetitive calls to locate the names of the paths associated with the alternate index entry (if any exist).

6. A call to locate the data entry.

7. *A call to locate the name of the alternate index entry associated with the data entry.

8. A call to locate the index entry.

9. *A call to locate the name of the alternate index entry associated with the index entry.

10. Repetitive calls to locate the path entries (if any exist).

11. *Repetitive calls to locate the alternate index, data and index (of alternate index), and data and index (of cluster) name associated with the path entries.

For each path entry:

1. A call to locate the path entry.

2. *For a path over a cluster, a call to locate the name of the cluster, and data and index (of cluster) associated with the path entry.

3. *For a path over an alternate index, a call to locate the name of the alternate index, data and index (of alternate index), and data and index (of cluster) associated with the path entry.

* Applies to VSAM catalogs only.

For each non-VSAM entry:

1. Locate the non-VSAM entry.

2. *Locate the name of any associations for each generation data group or alias.

For each user catalog entry:

1. Locate the user catalog entry.

2. *Locate the name of any associations for each alias.

For each space entry:

1. *Locate the space entry.

2. *Locate each data set name in a space entry, for example, three calls if three data sets are defined in the data space.

IDCMP01

1. Define the cluster, alternate index, or ICF user catalog.

2. Locate the cluster, alternate index, or ICF user catalog entry if the previous define failed because of a duplicate entry in the catalog.

3. Locate the data entry.

4. Locate the index entry, only for a duplicate KSDS cluster, alternate index, or ICF user catalog entry and if the temporary export flag is not set in the data entry.

5. Delete the entry, if there is a duplicate nonempty entry.

6. An iterative series of calls to delete any remaining parts of the structure.

7. Define the cluster again, if there was a duplicate entry.

8. Delete the defined entry, if an error occurred copying data into the defined entry.

9. An iterative series of calls to delete any remaining parts of the structure.

10. Alter the data entry, if the INHIBITTARGET keyword was specified at export time.

11. Alter the index entry, if the INHIBITTARGET keyword was specified at export time for a KSDS cluster.

* Applies to VSAM catalogs only.

IDCRC01 For VSAM clusters:

1. A call to locate the cluster entry.
2. A call to locate the data entry.
3. A call to locate the index entry for a KSDS or AIX.
4. Repetitive call to locate path entries, if they exist for a VSAM cluster.

For GDGs:

1. A call to locate the GDG entry.
2. A call to locate any non-VSAM association to the GDG.
3. Repetitive calls to locate alias entries for the non-VSAM association.

For non-VSAM entries:

1. A call to locate the non-VSAM entry.
2. Repetitive calls to locate alias entries if present.

IDCRM01

1. A call to define the object.
2. A call to delete the object if a duplicate name is indicated following the first call to catalog.
3. A series of calls to catalog to delete the remainder of the structure.
4. A call to define the object if a duplicate name was found.
5. A call to alter the name of the object if it is a VSAM entry and OUTFILE was specified to the dummy name specified on the OUTFILE ddcard.
6. A call to alter the name of the object back to its original name if the previous call was executed.
7. A call to delete the object defined if import fails after the define.
8. A series of calls to catalog to delete the remainder of the structure.

IDCRP01

1. Locate the INFILE data set type.
2. Locate the OUTFILE data set type.

IDCRS01

1. A call to locate the catalog data set name.
2. A call to locate the catalog volume serial number and time stamp.

IDCRS06

1. A call to define the work file.
2. A call to delete the work file.

IDCXP01

1. Locate to validate master catalog password if EXAMINE or ICF catalog.
2. Locate OPEN indicator and AMDSB.
3. Locate the cluster, alternate index, or ICF catalog entry.
4. Locate the ICF catalog entry if a STEPCAT or JOBCAT is not present.
5. Locate the data entry.
6. Locate the index entry for a KSDS cluster, an alternate index, or ICF catalog.
7. A call to locate the related base cluster name if the object being exported is an alternate index.
8. A series of iterative calls to locate catalog information about the path objects associated with the object.
9. A call to alter the data entry, if TEMPORARY, INHIBITSOURCE, or INHIBITTARGET was specified on the command.
10. A call to alter the index entry, if TEMPORARY, INHIBITSOURCE, or INHIBITTARGET was specified on the command, and the object is a KSDS cluster or an alternate index.
11. A call to delete the entry if PERMANENT was specified on the command.

12. A series of iterative calls to delete any remaining parts of the structure.

Using DIAGNOSE for BCS or VVDS Problems

The ICF catalog consists of two entities: the basic catalog structure (BCS) and associated VVDSs. When a VSAM data set is cataloged in an ICF catalog, an entry is placed in the BCS. Entries are also placed in each VVDS for the volumes upon which the data or index components reside. See Figure 22 on page 368 for an example of catalog information for a VSAM data set cataloged in an ICF catalog.

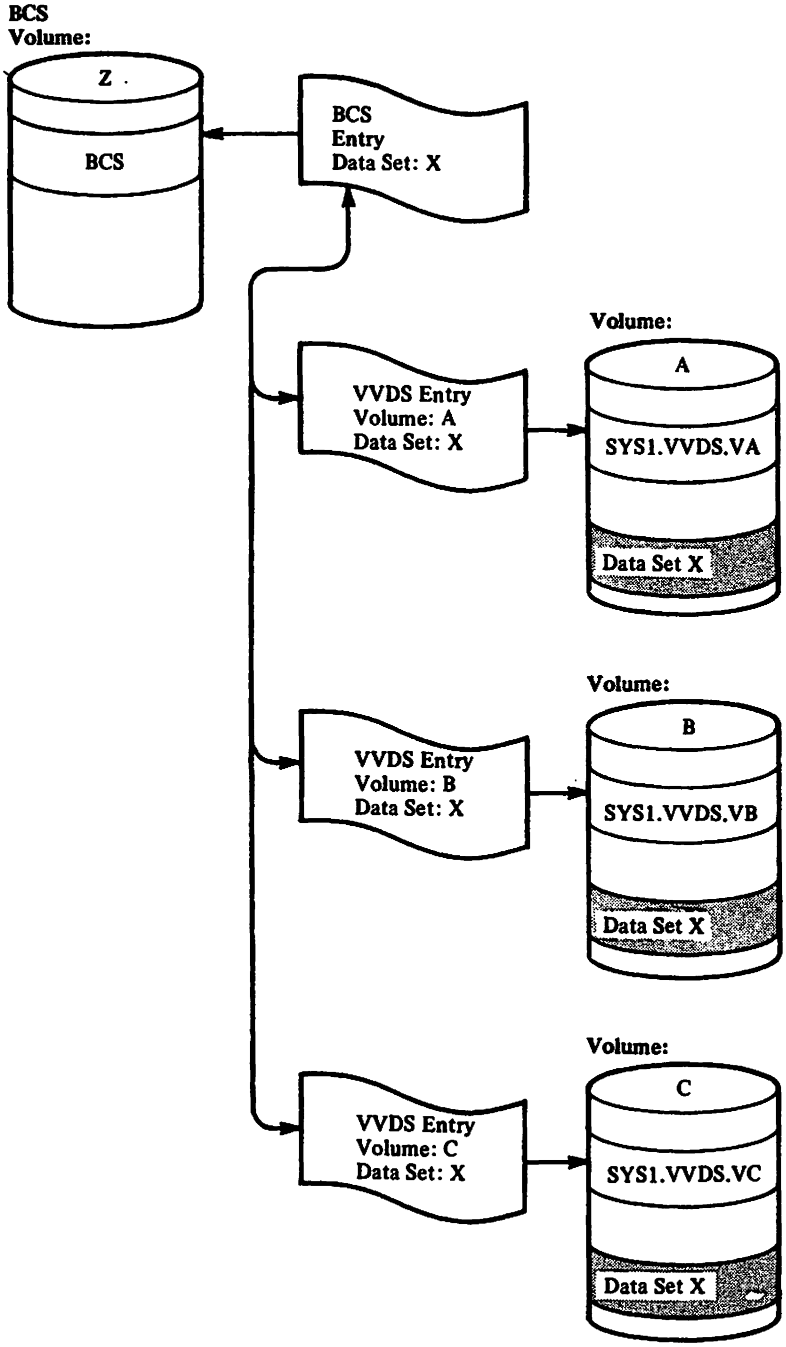
The DIAGNOSE command may be used to scan either the BCS or the VVDSs, looking for:

- Format errors in entries
- Errors between associated records in the BCS or VVDS
- Errors occurring between information in the BCS and any VVDSs.

  It also produces a listing with the above information.

When a problem with VVDS or BCS is encountered, consider running DIAGNOSE to expose those problems caused by damage to the data sets. If damage has occurred in either of these data sets, be sure to look for damage in the other data set. *Access Method Services Reference* gives the specifics of the DIAGNOSE command. Information about DIAGNOSE output may be found in *Catalog Administration Guide*. This manual also describes recovery procedures. DIAGNOSE messages are contained in *System Messages*. BCS and VVDS records are described in *Catalog Diagnosis Reference*.

NonVSAM and GDG data sets do not have VVDS entries, only BCS entries.



VSAM data set *x*, which exists on volume A, B, and C, is cataloged in the BCS on volume Z. The catalog information for VSAM data set *x* consists of the data in the BCS entry and the data in the three VVDS entries in this example.

Figure 22. Example of Catalog Information for a VSAM Data Set Cataloged in an ICF Catalog

Debugging a Formatting Problem

If data is misformatted, the problem may be in the parameters given to the UPRINT macro. The UPRINT parameters are: (1) the address of the GDT; (2) the address of an alternate IOCSTR or zero; (3) the address of a DARGLIST data area in storage; and (4) the address of a FMTLIST data area, if it is in storage. If the FMTLIST is in a static text module, the fourth parameter is zero and the DARGLIST contains information to find the FMTLIST. The DARGLIST and the FMTLIST control the formatting of the data. The DARGLIST in general contains information about the input data within the FMTLIST. The FMTLIST controls the order of formatting by the placement of the substructures. Refer to the "Data Areas" chapter for a detailed description of the GDT, IOCSTR, DARGLIST, and FMTLIST.

Problems are most likely to occur between the DARGLIST and the FMTLIST. The examples show how the text processor uses the DARGLIST and FMTLIST to format the data. With each example is a flowchart with blocks keyed to the FMTLIST substructure.

Example I

A module wants to space one line then print data starting in column 10 as shown in Figure 23 on page 372. The data is in the module's storage rather than in a static text module.

The output is:

```
70 characters of data starting in column 10
```

In the module's storage are:

- The data to be printed
- A DARGLIST
- A FMTLIST

The data is:

Offset	Name	Contents	Comments
0	any, INFO for example	70 characters of EBCDIC data	

The DARGLIST is:

Offset	Name	Contents	Comments
0	DARGDBP	INFO	Address of the block of data to be printed.
4	DARGRETP	0	The line is to be printed rather than just formatted and returned to the module without printing.
8	DARGSTID	0	No static text module is used—the FMTLIST and data are in the module's storage.
12	DARGILP	70	Number of characters to print.
14	DARGCNT	0	No insert or replication substructures occur in the FMTLIST.
16	DARGRETL	0	Since no data is returned, the length of the return area whose address is in DARGRETP.
18	DARGIND	0	Indicates printing is to start in the column indicated in FMTLIST. No DARGARY is defined because no insert or replication substructures are used in the FMTLIST.

The FMTLIST is:

Offset	Name	Contents	Comments
0	FMTFLGS	X'40'	Identifies these 6 bytes as a spacing substructure.
1	none	0	Unused.

Offset	Name	Contents	Comments
2	FMTSPF	1	Space one line.
4	FMTSPT	C'R'	Space the number of lines in FMTSPF relative to the last line printed.
5	none	0	Unused.
6	FMTFLGS	X'90'	Identifies these 12 bytes as a block substructure and the end of the FMTLIST.
7	none	0	Unused.
8	FMTILEN	70 or 0	If 70 is specified, it is used as the length of the data. If 0 is specified, the length of the converted data is used as the length to print. Because no conversion is being done in this example, the result is the same if 70 or 0 is specified.
10	FMTIOFF	0	Get the data starting with the first byte.
12	FMTOCOL	10	Place the data in output column 10.
14	FMTOLEN	70	Number of bytes to print. 0 would give the same result because no conversion is being done.
16	FMTCNVF	0	No conversion is being done on the data addressed by DARGDBP.

Discussion: The spacing substructure causes one line to be spaced.

The next substructure is identified as a block data substructure. The address of the block of data is in DARGDBP. No conversion is to be done on the data. The text processor moves the 70 bytes of data to the 10th byte in the next line.

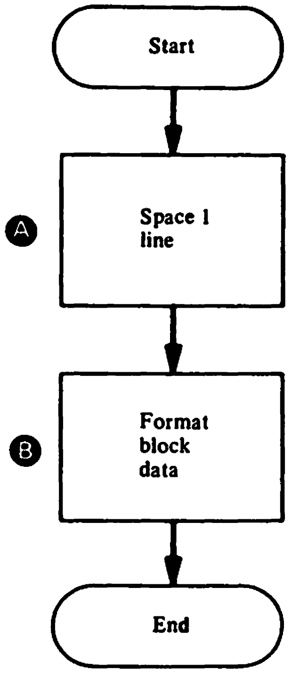


Figure 23. Formatting Example I

Example II

A module wants to space two lines, print a header, space two more lines, and print all of a block of data no matter how many lines the block of data takes with single spacing between subsequent lines (see Figure 24 on page 373). The header is in static text module IDCTSAL0 at entry X'03'. The block of data is in the module. Also, if there is no record number for the header, the module wants to print the word UNKNOWN.

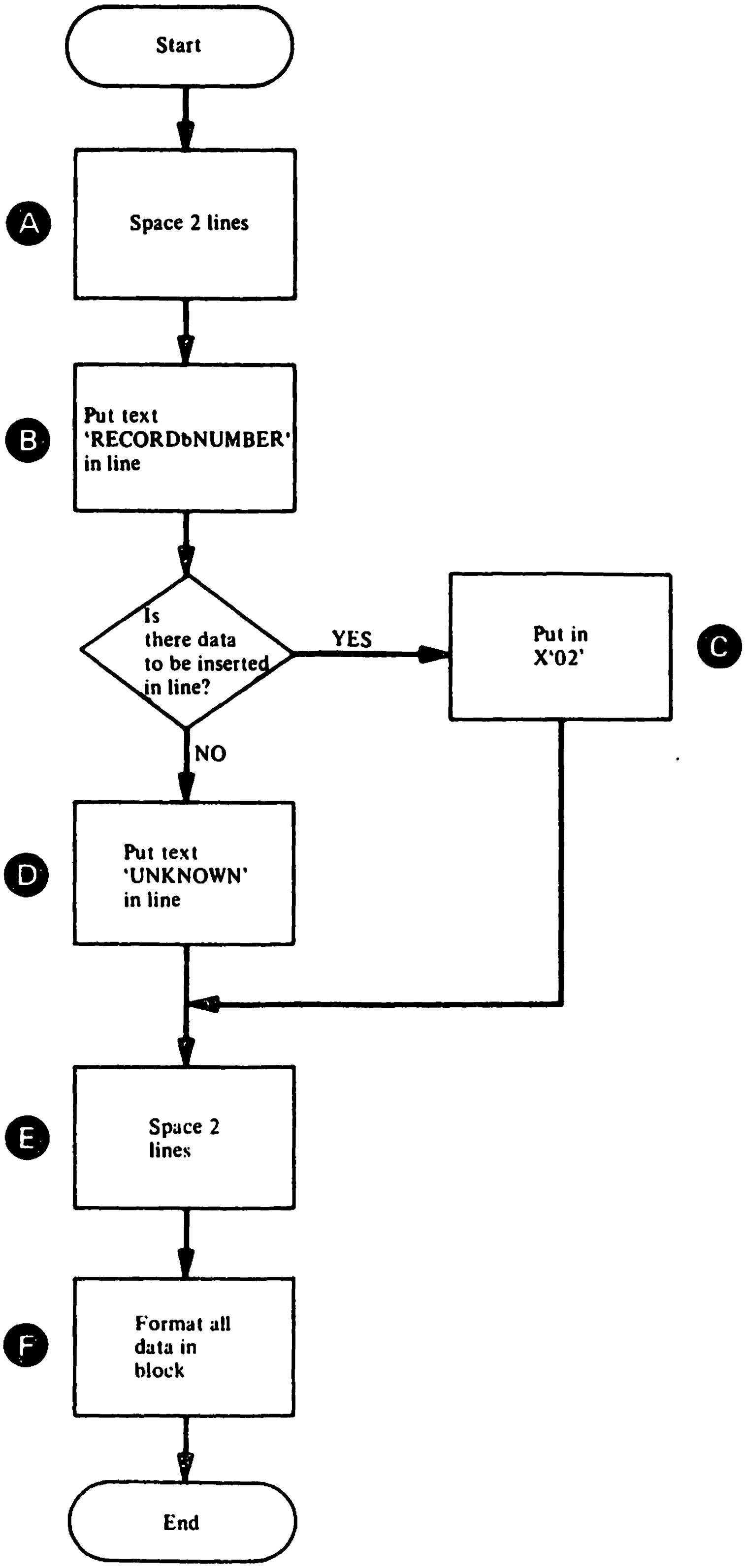


Figure 24. Formatting Example II

The output is:

```
(1 blank line)
RECORD NUMBER 002
(1 blank line)
xxxxxxx converted data for as many lines as necessary
```

The module has in its storage:

- The data for the record number in the header, in this example X'02'
- The block of data to convert and print
- A DARGLIST

Already existing in a static text module are:

- A FMTLIST
- Text for the header, in this example, the characters 'RECORDb̸NUMBER'

The data is:

Offset	Name	Contents	Comments
0	any, RECNUM for example	1 byte with the value X'02'	
1	any, DUMPIT for example	2000 bytes of binary data	The binary data will be converted to printable hexadecimal.

The DARGLIST is:

Offset	Name	Contents	Comments
0	DARGDBP	DUMPIT	Address of the block of data to convert.
4	DARGRETP	0	The lines are to be printed rather than just formatted and returned to the module without printing.
8	DARGSTID	C'AL0',X'03'	Static text identification to locate the FMTLIST—the FMTLIST IDCTSAL0 at entry 3.
12	DARGILP	2000	The length of DUMPIT.

Offset	Name	Contents	Comments
14	DARGCNT	1	One insert data appears in DARGARY.
16	DARGRETL	0	The length of the converted data is used as the number of bytes to print.
18	DARGIND	0	Printing starts in the column indicated in FMTLIST.
19	none	0	Unused.
20	DARGARY	none	DARGARY is the name of the rest of DARGLIST.
20	DARGINS	4	This number is matched with an insert substructure in FMTLIST.
22	DARGINL	1	The number X'02' occupies 1 byte.
24	DARGDTM	RECNUM	Address of the number X'02' in the module.

At entry X'03' in static text module IDCTSAL0 is:

Offset	Name	Contents	Comments
0	TXT	71	Length of the FMTLIST and the data that follows the FMTLIST.
2	FLG	0	This static text entry is for data not a message or header.
4	FMTFLGS	X'40'	Identifies these 6 bytes as a spacing substructure.
5	none	0	Unused.
6	FMTSPF	2	Space two lines.
8	FMTSPT	C'R'	Space the lines relative to the last printed line.
9	none	0	Unused.

Offset	Name	Contents	Comments
10	FMTFLGS	X'04'	Identifies these 10 bytes as a static text substructure—the data is immediately after the FMTLIST.
11	none	0	Unused.
12	FMTSTL	13	Number of bytes in C'RECORDℬNUMBER'.
14	FMTSTO	54	Number of bytes the data C'RECORDℬNUMBER' is from the first substructure in FMTLIST.
16	FMTOCOL	1	The data C'RECORDℬNUMBER' is to be printed in column 1.
18	FMTOLEN	0	0 indicates the output length is the same as the input length for this data.
20	FMTFLG	X'20'	Identifies these 12 bytes as an insert substructure.
21	none	0	Unused.
22	FMTRFNO	4	This number is matched with the number in DARGINS in order to get the address of the data X'02'.
24	none	0	Unused.
26	FMTOCOL	15	The data X'02' is printed in column 15.
28	FMTOLEN	3	The converted data is to take up three columns.
30	FMTCNVF	X'1000'	The data X'02' is to be converted from byte to zoned decimal
32	FMTFLGS	X'02'	Identifies these 8 bytes as a default text substructure.
33	none	0	Unused.

Offset	Name	Contents	Comments
34	FMTILEN	7	Number of bytes in the data C'UNKNOWN'.
36	FMTIOFF	67	Number of bytes the data C'UNKNOWN' is from the first substructure in FMTLIST.
38	FMTOCOL	15	The data C'UNKNOWN' is printed in column 15.
40	FMTFLGS	X'40'	Identifies these 6 bytes as a spacing substructure.
41	none	0	Unused.
42	FMTSPF	2	Space two lines.
44	FMTSPT	C'R'	The two lines are spaced relative to the last printed line.
45	none	0	Unused.
46	FMTFLGS	X'90'	Identifies these 12 bytes as a block data substructure and the last substructure in FMTLIST.
47	none	0	Unused.
48	FMTILEN	0	Zero means use the length of the block data in DARGILP.
50	FMTIOFF	0	Start at the first byte of the block data.
52	FMTOCOL	1	Start the block of data in output column 1.
54	FMTOLEN	0	Zero means print the block data until the input is exhausted no matter how many lines it takes.
56	FMTCNVF	X'8000'	Convert the block of data from binary to printable hexadecimal.
58	any	C'RECORD ɓNUMBER'	Data for the second substructure.

Offset	Name	Contents	Comments
71	any	C'UNKNOWN'	Data for the default text substructure.

Discussion: The first spacing substructure causes two lines to be spaced.

The static text 'RECORD₿NUMBER' is put in the next line.

The insert number in the insert substructure is matched with the insert number in DARGLIST. The number X'02' from the module is converted to zoned decimal and placed in column 15.

The next spacing substructure causes two more lines to be spaced.

The block data substructure causes the data addressed by DARGDBP to be converted to printable hexadecimal until all the bytes in DARGILP have been converted and printed. If the module wants to print the same lines again but with a different record number and different block data, only DARGDBP, and DARGDTM need to be changed. If there had not been a reference number 4 in DARGLIST, the data 'UNKNOWN' would have been printed instead of the record number '002' This allows more freedom for the module to vary the output just by changing insert reference numbers in the DARGLIST.

Example III

A module wants to space three lines then print repeating fields on different lines so the output would appear as (see Figure 25 on page 385):

```
(2 blank lines)
field A    field B    X'field C1'   field D1   field E1
                      X'field C2'   field D2   field E2
```

The module has in storage:

- All the data to be printed
- A DARGLIST
- A FMTLIST

The data is:

Offset	Name	Contents	Comments
0	A	4 bytes of EBCDIC data	
4	B	4 bytes of packed decimal data	
8	C1	2 bytes of binary data	
10	D1	2 bytes of binary data	
12	E1	1 byte of EBCDIC data	
13	C2	2 bytes of binary data	
15	D2	2 bytes of binary data	
17	E2	1 byte of EBCDIC data	

The DARGLIST is:

Offset	Name	Contents	Comments
0	DARGDBP	A	
4	DARGRETP	0	The lines are to be printed rather than just formatted and returned to the module.
8	DARGSTID		No static text module is used.
12	DARGILP	18	Number of bytes from field A through field E2.
14	DARGCNT	1	There is one repetition substructure in the FMTLIST.
16	DARGRETL	0	The length of the converted data is used as the number of bytes to print.
18	DARGIND	0	Printing starts in the column indicated in FMTLIST.
19	none	0	Unused.
20	DARGREP	7	Number that is matched with a repetition substructure in FMTLIST.
22	DARGPCT	2	The group of fields identified by repetition substructure 7 in FMTLIST is to be printed twice.

The FMTLIST is:

Offset	Name	Contents	Comments
0	FMTFLGS	X'40'	Identifies these 6 bytes as a spacing substructure.
1	none	0	Unused.
2	FMTSPF	3	Space three lines.
4	FMTSPT	C'R'	Space the lines relative to the last printed line.

Offset	Name	Contents	Comments
5	none	0	Unused.
6	FMTFLGS	X'10'	Identifies these 12 bytes as a block data substructure.
7	none	0	Unused.
8	FMTILEN	4	Number of bytes in field A.
10	FMTIOFF	0	Field A begins zero bytes from the block of data whose address is in DARGDBP.
12	FMTOCOL	1	Print field A starting in column 1.
14	FMTOLEN	4	Number of bytes the converted field A occupies in the printed line.
16	FMTCNVF	0	No conversion is done on field A.
18	FMTFLGS	X'10'	Identifies these 12 bytes as a block data substructure.
19	none	0	Unused.
20	FMTILEN	4	Number of bytes of storage field B occupies.
22	FMTIOFF	4	Field B starts 4 bytes from the block of data whose address is in DARGDBP.
24	FMTOCOL	10	Print field B starting in column 10.
26	FMTOLEN	10	Number of bytes the converted field B occupies in the printed line.
28	FMTCNVF	X'0880'	Convert field B from packed decimal to unpacked decimal with zero suppression.
30	FMTFLGS	X'08'	Identifies these 8 bytes as a replication substructure.

Offset	Name	Contents	Comments
31	none	0	Unused.
32	FMTRENO	7	Matched with a number in DARGLIST to find the number of iterations.
34	FMTRBC	3	The data identified in the next three substructures is to be repeated.
36	FMTRIO	5	The number of bytes from field C1 to field C2 in storage. This number is added to the address of the first field each time the field is repeated.
38	FMTFLGS	X'10'	Identifies these 12 bytes as a block data substructure for fields C1 and C2.
39	none	0	Unused.
40	FMTILEN	2	Number of bytes fields C1 and C2 each occupy in storage.
42	FMTIOFF	8	Number of bytes from field A to field C1.
44	FMTOCOL	22	Print fields C1 and C2 starting in column 22.
46	FMTOLEN	7	Number of bytes the converted fields C1 and C2 each occupy in the printed line.
48	FMTCNVF	X'4000'	Convert fields C1 and C2 from binary to printable hexadecimal enclosed in X'data'.
50	FMTFLGS	X'10'	Identifies these 12 bytes as a block data substructure for fields D1 and D2.
51	none	0	Unused.
52	FMTILEN	2	Number of bytes fields D1 and D2 each occupy in storage.

Offset	Name	Contents	Comments
54	FMTIOFF	10	Number of bytes from field A to field D1.
56	FMTOCOL	31	Print fields D1 and D2 starting in column 31.
58	FMTOLEN	6	Number of bytes the converted fields D1 and D2 each occupy in the printed line.
60	FMTCNVF	X'1000'	Convert fields D1 and D2 from binary to printable decimal.
62	FMTFLGS	X'90'	Identifies these 12 bytes as a block data substructure for fields E1 and E2 and the last substructure in the FMTLIST.
63	none	0	Unused.
64	FMTILEN	1	Number of bytes fields E1 and E2 each occupy in storage.
66	FMTIOFF	12	Number of bytes from field A to field E1.
68	FMTOCOL	39	Print fields E1 and E2 each starting in column 39.
70	FMTOLEN	1	Number of bytes the converted fields E1 and E2 each occupy in the printed line.
72	FMTCNVF	X'0000'	No conversion is done on fields E1 and E2.

Discussion: The first spacing substructure causes three lines to be spaced.

The block data substructures for fields A and B describe the location of A and B within the block addressed in DARGDBP. Field A is not converted. Field B is converted from packed decimal to zoned decimal and leading zeros are replaced with blanks.

The replication substructure number is matched with an identification number in DARGREP. When a match is found, the DARGPCT immediately after DARGREP tells how many times to repeat the substructures. If the module wants to use the same FMTLIST and print another group of fields C, D, and E, only DARGPCT needs to be changed. The replication substructure tells how many substructures to repeat and an offset that is used to find the group of fields being repeated. On the first repetition the offset is not used, on the second it is added once; on the third repetition it is added twice.

The next substructure describes C1 and C2. On the first repetition the value in FMTIOFF is added to the value in DARGDBP to find field C1. To find field C2, FMTIOFF and FMTRIO in the repetition substructure are added to DARGDBP. Each time a group of substructures is repeated a new line is printed because the output columns for each substructure do not change. For example, in order to print both C1 and C2 in column 22, a new line must be printed. Both C1 and C2 are converted to printable hexadecimal preceded by X and followed by a single quote.

Fields D1 and D2 are described by the next substructure. D1 and D2 are converted to printable decimal.

The substructure for fields E1 and E2 is also the end of FMTLIST. E1 and E2 are not converted.

After E1 is formatted, the three substructures following the repetition substructure are repeated. A new line is started because FMTOCOL keeps the output and the columns as they were each time a field is printed. Fields C2, D2, and E2 are put in the next line. The FMTLIST is finished after E2 is printed.

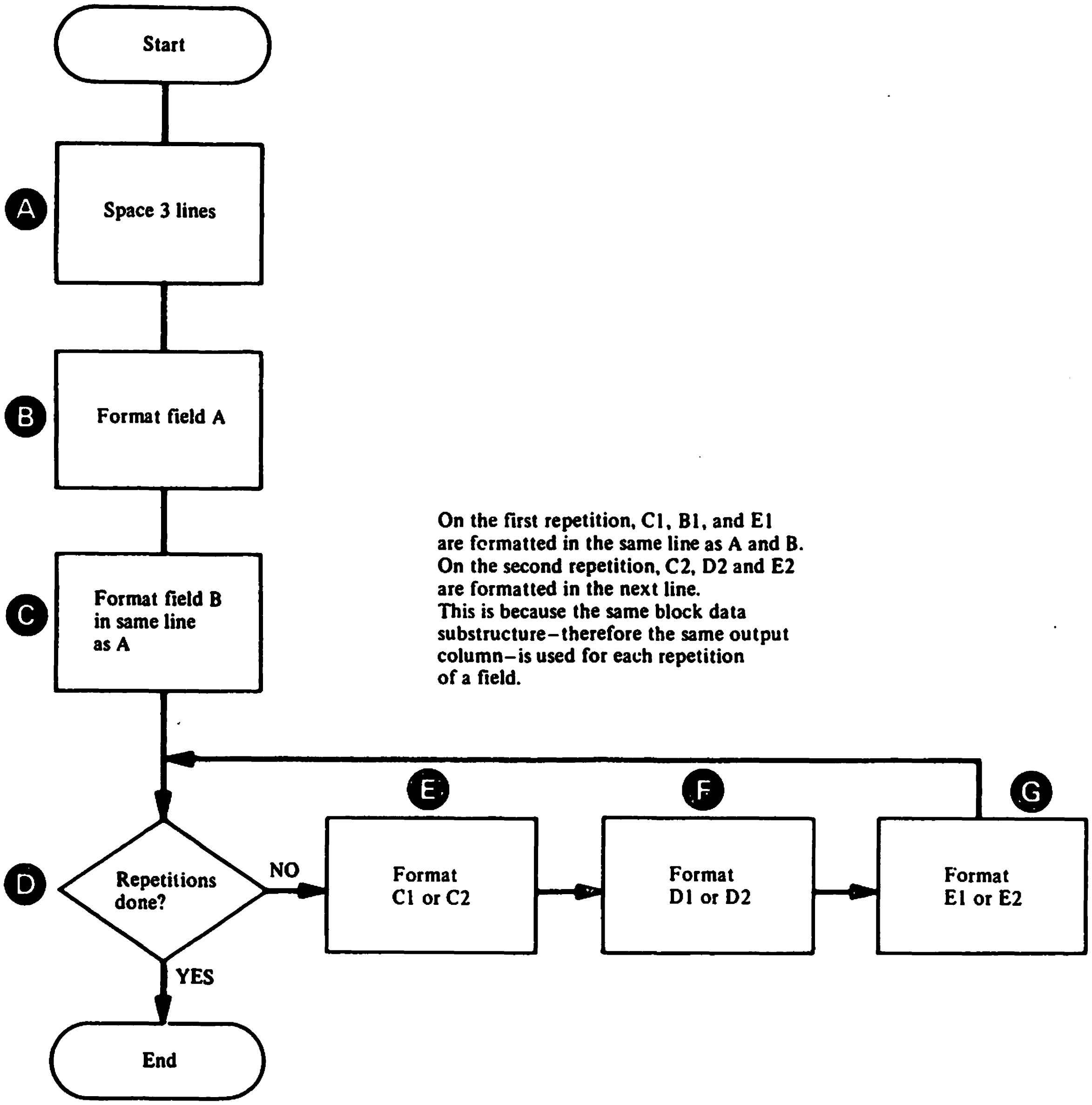

Start
A
Space 3 lines
B
Format field A
C
Format field B in same line as A
On the first repetition, C1, B1, and E1 are fcrmatted in the same line as A and B. On the second repetition, C2, D2 and E2 are formatted in the next line. This is because the same block data substructure–therefore the same output column–is used for each repetition of a field.
D
Repetitions done?
NO
YES
E
F
G
Format C1 or C2
Format D1 or D2
Format E1 or E2
End

Figure 25. Formatting Example III

Obtaining a Dump for a Text Processor Problem

If you do not have an ABEND dump within the text processor routines or an ABORT snap dump within the text processor, you can use the test option to obtain a dump. You may want to obtain a dump within the routine that invoked the text processor or within the text processor itself.

The "Module or CSECT to Dump Points Cross-Reference" list contains all the dump points within the processor; you can specify these dump points on the FULL option of the TEST keyword to obtain a full region dump.

The text processor has dump points before and after it converts data to printable form. You should use these dump points if there is an error in converting the data.

How to Find Text Processor Argument Lists

If you suspect a problem within the text processor, the two structures you should locate in a dump are the print control table (PCT) and the dynamic data argument list (DARGLIST). The PCT and the DARGLIST are described in the chapter "Data Areas." The eighth word of the GDT contains the address of the PCT. The address of the DARGLIST is the third parameter passed to IDCTP01 for a UPRINT request.

Two other structures that you may find helpful to locate in a dump are the queue of format structures and the print buffer.

Note: In the listings the print buffer is called the stack buffer.

Figure 26 on page 387 shows the queue of format structures maintained by the text processor. There is an entry in the queue for each text structure entry used for the current function. Each entry in the queue contains the 4-byte static text identifier specified in the DARGLIST. The first 3 bytes contain the last 3 characters of the text-structure module name; the fourth byte contains the entry number of the format structure within the module.

Figure 27 on page 388 shows the print buffer maintained by the text processor. It contains the records, other than messages, that have not been printed. The records to be printed are kept in the print buffer until the buffer becomes full or a message must be printed. The primary and secondary PCTs contain the address of the first record in the buffer and the address of the next empty space in the buffer. If both addresses are equal, the buffer is empty.

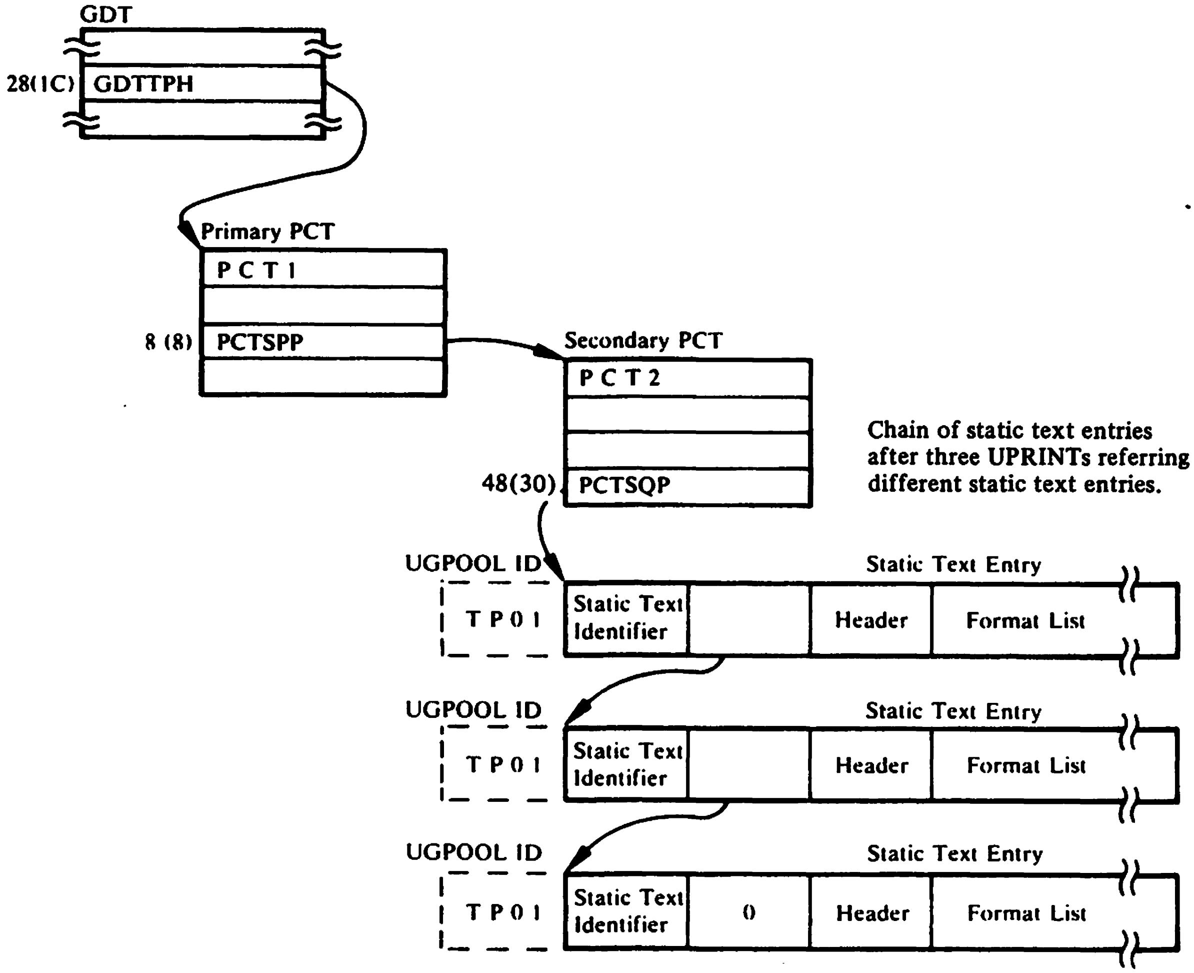
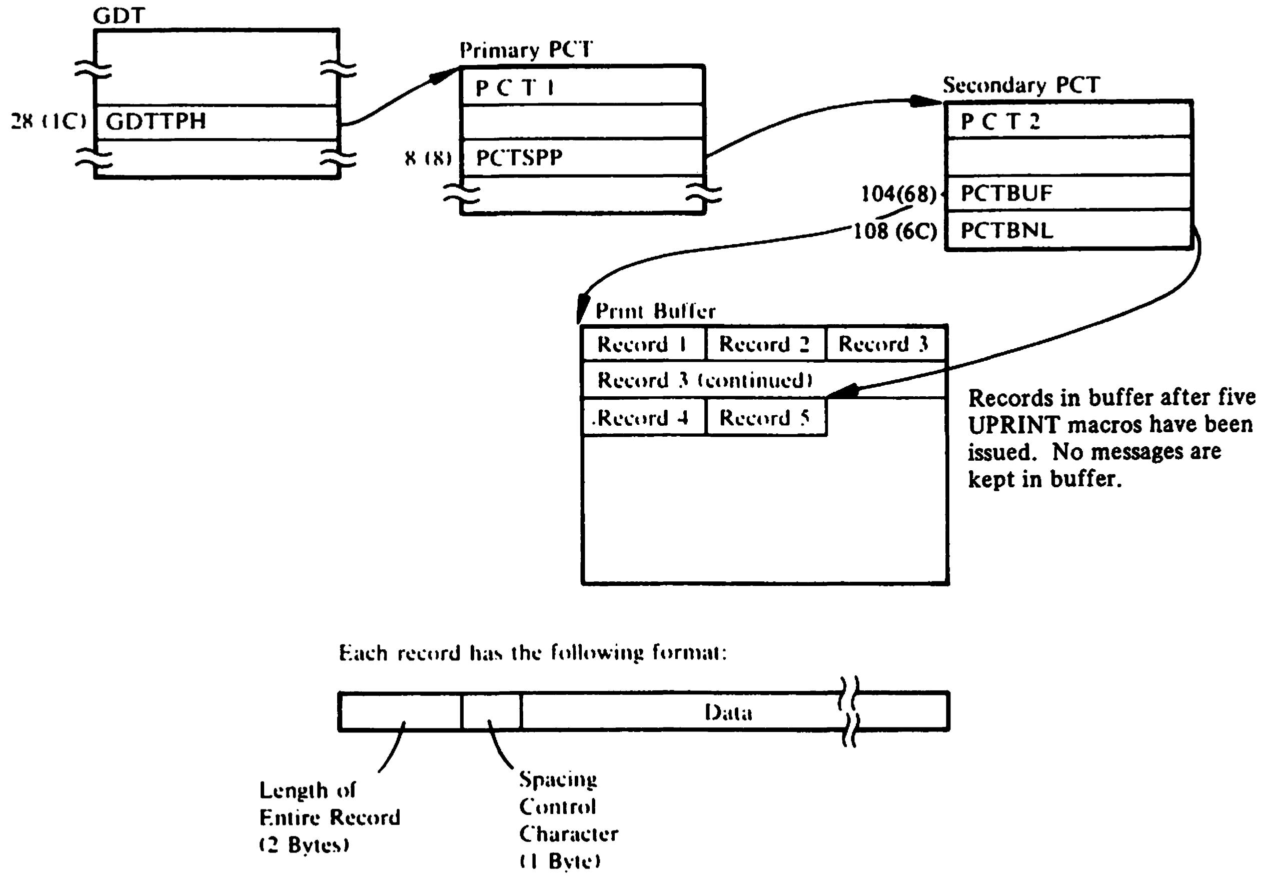


Figure 26. Text Processor Format Structure Queue



Records in buffer after five UPRINT macros have been issued. No messages are kept in buffer.

Each record has the following format:

Figure 27. Text Processor Print Buffer

Debugging an I/O Problem

There may be an I/O problem within system I/O routines, or within access method services if an ABORT condition occurs in the I/O adapter or if an ABEND occurs within the system I/O routines. To determine whether the problem exists in the routines that invoke the I/O adapter, in the I/O adapter itself, or in the system I/O routines, you must examine the argument lists passed between the I/O adapter and the invoking routines, and the I/O adapter and the system I/O routines.

This section explains how to obtain a dump that contains the I/O argument lists and how to find the argument lists in a dump.

Obtaining a Dump for an I/O Problem

If you do not have an ABEND dump within system I/O routines or an ABORT snap dump within the I/O adapter, you can use the test option to obtain a dump. You may want to obtain a dump within the routine that invoked the I/O adapter or within the I/O adapter itself.

The "Module or CSECT to Dump Points Cross-Reference" list contains all the dump points within the processor; you can specify these dump points on the FULL option of the TEST keyword to obtain a full region dump.

The I/O adapter has dump points before and after it issues the OPEN SVC (dump points IO1O and IO20) and before it issues the CLOSE SVC (dump point IO1C). You should use these dump points if there is an error opening or closing data sets. The I/O adapter has a dump point (IOVR) after issuing a VSAM I/O request which returns a nonzero return code. You should use this dump point if you wish to obtain a dump in a VSAM I/O error situation.

How to Find I/O Argument Lists

The input/output communications structure (IOCSTR), which is constructed for each data set that has been opened, contains pointers to most of the control blocks used by the system I/O routines. The IOCSTR is also the argument list that is passed between the I/O adapter and the routines that invoke the I/O adapter, except for the initial open request. Thus, once you find the IOCSTR, you can find most of the other arguments passed between the I/O adapter and other routines. The chapter "Data Areas" explains the format of the IOCSTR.

Figure 28 on page 390 shows the chain of IOCSTRs constructed for all opened data sets; however, the data sets may not have been opened successfully. The I/O adapter historical area contains a pointer to the start of the chain.

You can find the address of the IOCSTR for a particular I/O request by finding the parameter list passed to IDCIO01 by the invoking routine. Register 1 of the registers saved by IDCIO01 contains the address of a parameter list. The second word of the parameter list contains the address of the IOCSTR. The third, fourth, and fifth words may contain addresses of additional IOCSTRs.

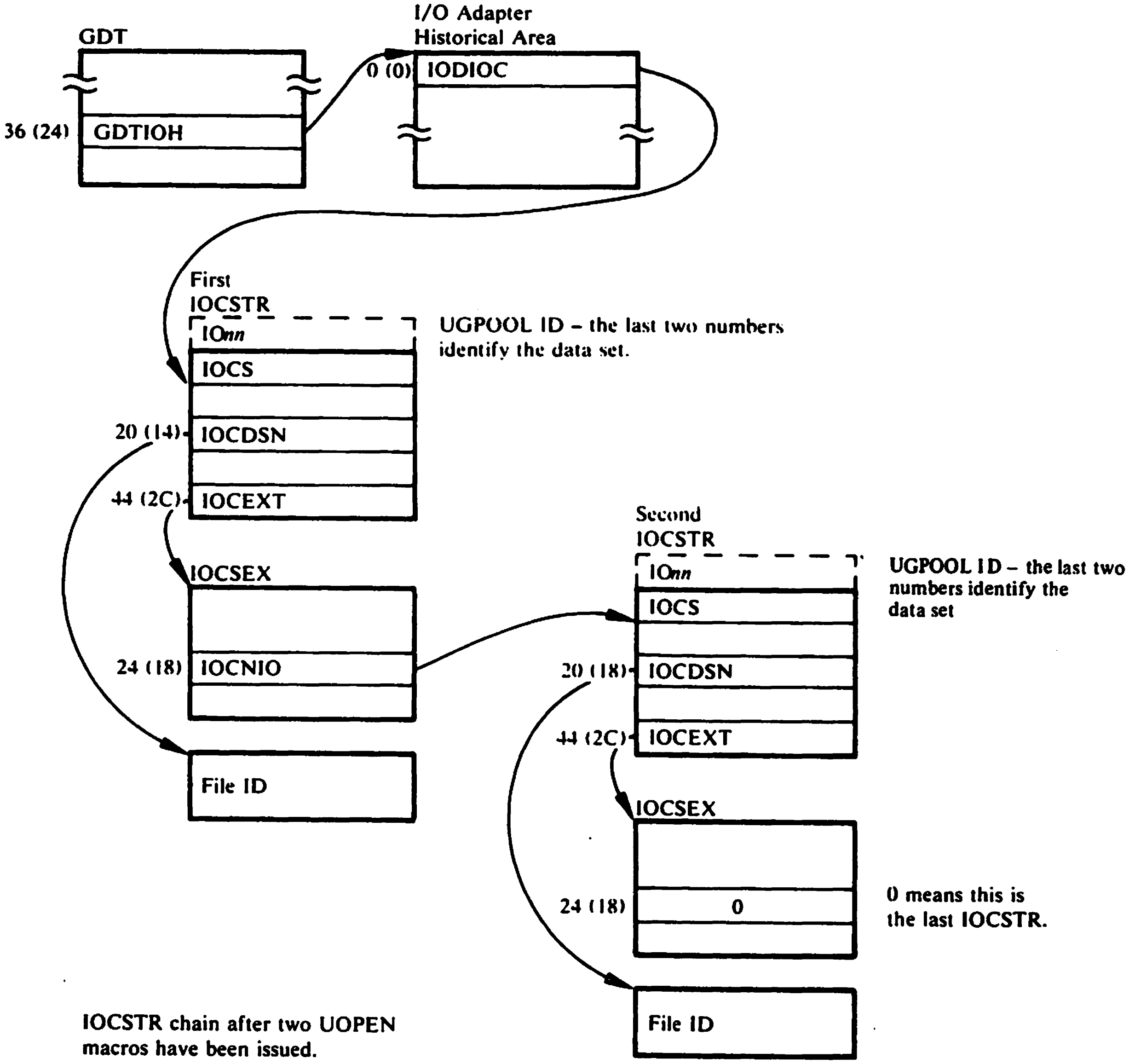
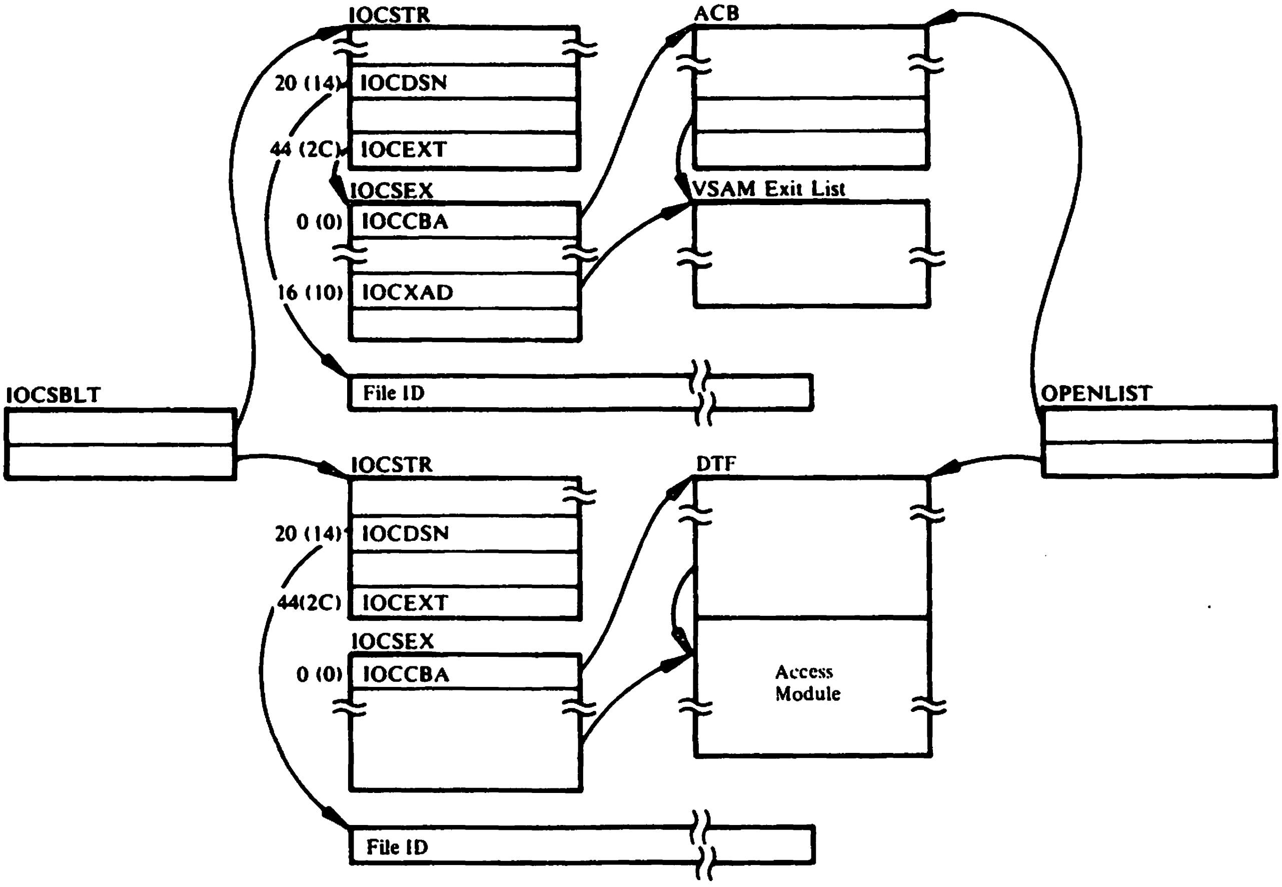

GDT
36 (24)
GDTIOH
I/O Adapter
Historical Area
0 (0)
IODIOC
First
IOCSTR
IOnn
UGPOOL ID - the last two numbers identify the data set.
IOCS
20 (14)
IOCDSN
44 (2C)
IOCEXT
IOCSEX
24 (18)
IOCNIO
File ID
Second
IOCSTR
IOnn
UGPOOL ID - the last two numbers identify the data set
IOCS
20 (18)
IOCDSN
44 (2C)
IOCEXT
IOCSEX
24 (18)
0
0 means this is the last IOCSTR.
File ID
IOCSTR chain after two UOPEN macros have been issued.

Figure 28. IOCSTR Chain

Open Argument Lists

Figure 29 on page 391 shows how the I/O control blocks are connected before the OPEN SVC is issued. The IOCSTR addresses can be found from the IOCSTR chain as shown in Figure 28. The IOCSBLT table, which contains pointers to the IOCSTRs for the data sets being opened, can be found at location IOCSBLT in IDCIO01's automatic storage area. The OPENLIST table, which contains pointers to the DCBs and ACBs for the data sets being opened, can be found at location OPENLIST in IDCIO01's automatic storage area. In a system dump within the OPEN SVC, register 1 of the registers saved at entry to the SVC contains the address of the OPENLIST table.



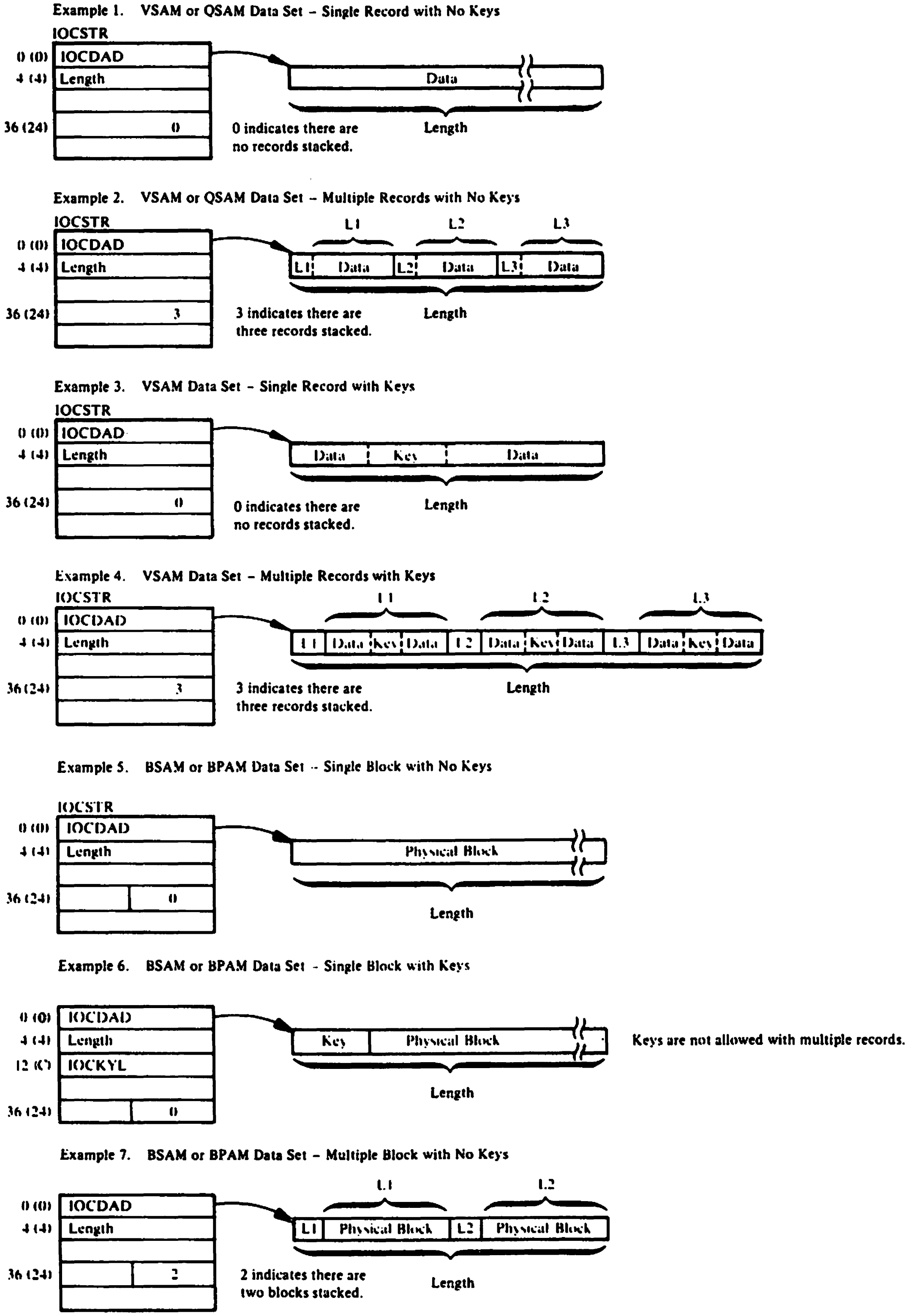
Two data sets are to be opened, one
VSAM and one non-VSAM data set.

Figure 29. I/O Control Blocks before OPEN

UGET and UPUT Argument Lists

This section contains some examples of input and output from the UGET and UPUT macros. These examples may be helpful in determining whether the IOCSTR and records for a UPUT request have been passed correctly to the I/O adapter, and whether the IOCSTR and records for a UGET request have been returned correctly by the I/O adapter.

Figure 30 on page 392 shows the IOCSTRs and records passed to the I/O adapter via a UPUT macro. The I/O adapter adds record descriptor words (RDWs) for QSAM output before passing the records to QSAM.



Example 1. VSAM or QSAM Data Set - Single Record with No Keys

Example 2. VSAM or QSAM Data Set - Multiple Records with No Keys

Example 3. VSAM Data Set - Single Record with Keys

Example 4. VSAM Data Set - Multiple Records with Keys

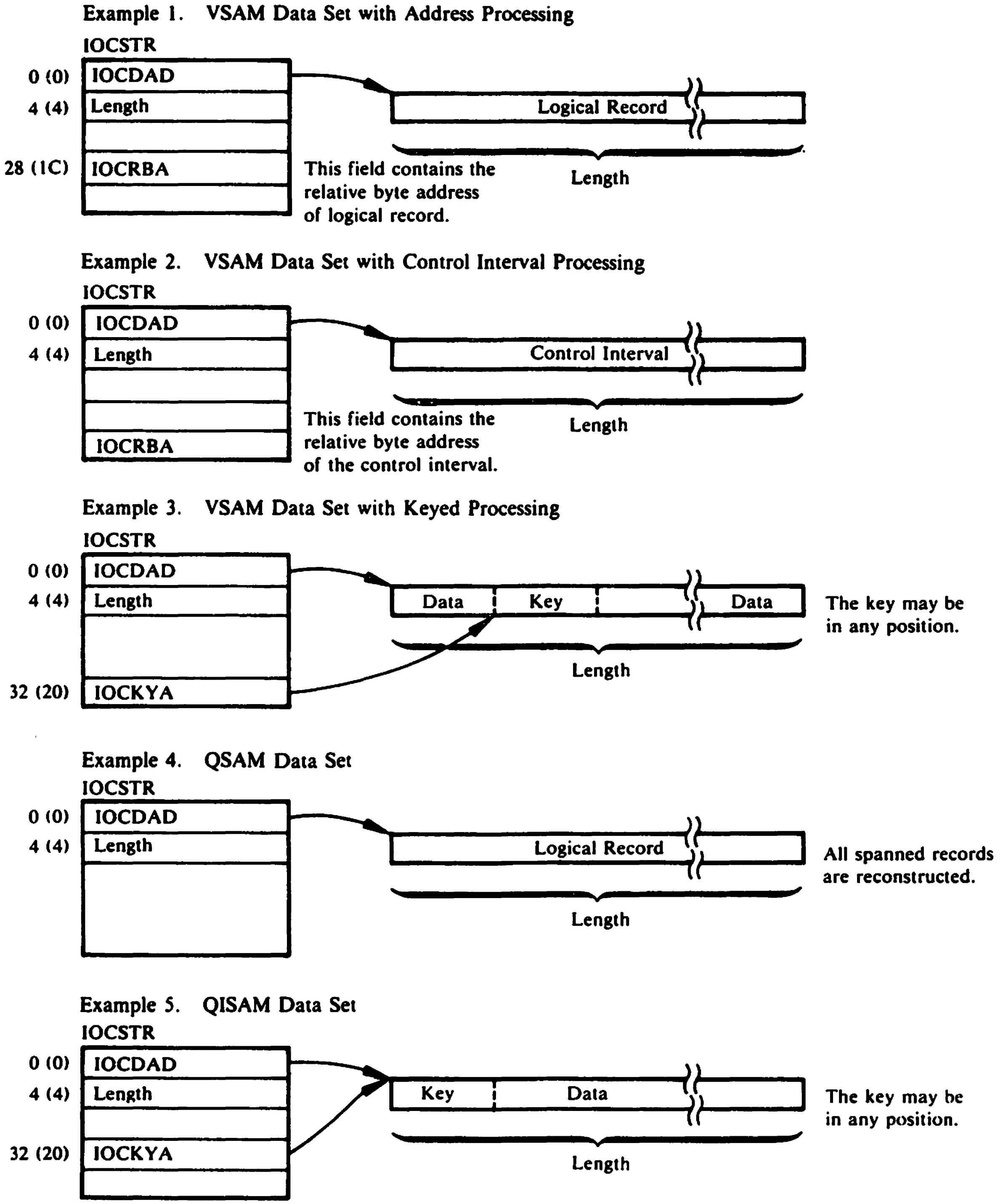
Example 5. BSAM or BPAM Data Set - Single Block with No Keys

Example 6. BSAM or BPAM Data Set - Single Block with Keys

Example 7. BSAM or BPAM Data Set - Multiple Block with No Keys

Figure 30. Input to UPUT Macro

Figure 31 shows the IOCSTRs and data returned by the I/O adapter after a UGET macro is processed.



Example 1. VSAM Data Set with Address Processing

Example 2. VSAM Data Set with Control Interval Processing

Example 3. VSAM Data Set with Keyed Processing

Example 4. QSAM Data Set

Example 5. QISAM Data Set

Figure 31. Output from UGET Macro

Processor Messages

The following lists all the messages printed by the processor. For each message, the following information is listed: the last three characters of the text structure module that contains the message followed by the number of the message within the module; the module(s) that causes the message to be printed; the procedure (within that module) which detects the situation that causes the message to be printed; and the situation that causes the message to be printed. For message IDC4999I there is no text structure module because the message is written when a UABORT macro is issued. The table lists the UABORT code instead of the text structure module for message IDC4999I.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC0001I	UV0-1 UV0-9	IDCAL01	IDCAL01	Function was completed without a severe error.
		IDCCC01	IDCCC01	Function was completed without a severe error. All or part of the OS/VS catalog entry was converted to VSAM or ICF catalog entries.
		IDCBI01	TERMPROC	Function was completed without an error or without a severe error in processing the base cluster.
		IDCDE01	IDCDE01	Function was completed without a severe error.
		IDCDL01	IDCDL01	Function was completed without a severe error.
		IDCLC01	IDCLC01	Function was completed without a severe error. All or part of the desired catalog listing was generated.
		IDCLR01	CLEANUP	Function was completed without a severe error.
		IDCMP01	IDCMP01	Function was completed without a severe error.
		IDCPM01	IDCPM01	Function was completed without a severe error.
		IDCPR01	IDCPR01	Function was completed without error, or (1) an end-of-file was reached in the input data set before the ending delimiter specified by the user, or (2) a recoverable I/O error occurred while retrieving or printing a record, or (3) an error occurred closing data sets.
		IDCRC01	EXITTHE	Function was completed without a severe error.
		IDCRM01	IDCRM01	Function was completed without a severe error.

Message	STID	Module	Procedure	Situation That Caused Message
		IDCRP01	IDCRP01	Function was completed without error, or (1) an end-of-file was reached in the input data set before the ending delimiter specified by the user, or (2) a recoverable I/O error occurred while copying a record, or (3) an error occurred closing data sets.
		IDCRS01	WRAPUP	Function was completed without a severe error.
		IDCVY01	IDCVY01	Function was completed without a severe error.
		IDCXP01	IDCXP01	Function was completed without a severe error.
IDC0002I	UV0-2	IDCEX03	IDCEX03	Access method services completed processing.
IDC0005I	UV0-5	IDCPR01	IDCPR01	Printing of records is completed.
		IDCRP01	IDCRP01	Copying of records is completed.
IDC0014I	UV0-15	IDCCC01	IDCCC01	In a TSO environment, a nonzero return code condition was encountered.
		IDCVY01	IDCVY01	In a TSO environment, a nonzero return code condition was encountered.
		IDCLC01	IDCLC01	In a TSO environment, a nonzero return code condition was encountered.
		IDCBI01	TERMPROC	In a TSO environment, a nonzero return code condition was encountered.
		IDCDL01	IDCDL01	In a TSO environment, a nonzero return code condition was encountered.
		IDCXP01	IDCXP01	In a TSO environment, a nonzero return code condition was encountered.
		IDCMP01	IDCMP01	In a TSO environment, a nonzero return code condition was encountered.
		IDCRS05	CKERR	In a TSO environment, a nonzero return code condition was encountered.
		IDCRM01	IDCRM01	In a TSO environment, a nonzero return code condition was encountered.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
		IDCAL01	IDCAL01	In a TSO environment, a nonzero return code condition was encountered.
		IDCDE01	IDCDE01	In a TSO environment, a nonzero return code condition was encountered.
IDC01407I	CC0-23	IDCCC01	VSAMCNVT	The sphere entry and any of its associations were successfully backed out of the target catalog.
IDC01408I	CC0-24	IDCCC02	DELSPACE	The data spaces are removed from the specified volume.
IDC0204I	RI0-5	IDCRI03	IDCRI03	The preceding command was scanned for syntax-checking purposes only.
IDC0206I	RI0-7	IDCRI01	SCANSEP	An extra comma was found between parameters.
IDC0222I	RI0-23	IDCRI01	NXTFIELD	A semicolon was found within a quoted constant.
IDC0233I	RI0-34	IDCRI01	SCANCMD	Too many closing parentheses were found at the end of a command or subparameter list.
IDC0234I	RI0-35	IDCRI01	INREPEAT	Too few parentheses were found at the end of a command.
			SCANCMD	Too few parentheses were found at the end of a command.
IDC0339I	IO0-55	IDCIO01	CRYPTWH	Informational message providing file keyname and enciphered data key.
IDC0342I	IO0-58	IDCIO01	CRYPTWH	Informational message providing generated private data key.
IDC0361I	SA7-9	IDCSA07	IDCSALC	The VSAM LOCATE request was unsuccessful. Refer to the message issued immediately prior to this one for the reason.
IDC0362I	SA7-10	IDCSA07	VSAMUCT	An error occurred during a VSAM DELETE request. Refer to a message issued prior to this one for the reason why the DELETE failed. As a result of the error, the data set named in this message was not scratched.
IDC0363I	SA7-11	IDCSA07	VSAMUCT	An error occurred during a VSAM DELETE request. Refer to a message issued prior to this one for the reason why the DELETE failed. As a result of the error, the data set named in this message was not uncataloged.

Message	STID	Module	Procedure	Situation That Caused Message
IDC0394I	SA6-14	IDCSA06	DEMNTVOL	IDCSA06 failed to demount the requested volume. (This is an informational message for the operator.)
		IDCSA10	RECOVERY	An attempt to demount a volume in STAE/ESTAE exit routine failed. (This is an informational message for the operator.)
IDC0396I	SA7-1	IDCSA07	GETENT	The data set named was not recataloged.
			UPDATENT	A previous message has the reason.
			TESTENT	A previous message has the reason.
IDC0397I	SA7-2	IDCSA07	UPDATENT	The data set identified was located but could not be recataloged. The return code from recataloging was 8. This situation should occur only when the volume was originally located in a VSAM or ICF catalog—the LOCATE macro supports a VSAM or ICF catalog, but the CATLG macro to recatalog does not. Because all data sets in the VSAM catalog that owns the volume have been recataloged, the data set must have been located in a VSAM catalog that did not own the volume.
IDC0398I	SA7-5	IDCSA07	TESTENT	The data set identified resides on more than 20 volumes and apparently has not been recataloged.
			SCANCMD	Too few parentheses were found at the end of a command.
IDC0508I	DE0-9	IDCDE01	IDCDE01	Define of the data set is completed.
		IDCMP01	CLUSPROC	Define of the data set being imported is completed.
		IDCRM01	CLUSPROC	Define of the data set is completed.
IDC0509I	DE0-10	IDCDE01	IDCDE01	Define of the data set is completed.
		IDCMP01	CLUSPROC	Define of the data set being imported is completed.
		IDCRM01	CLUSPROC	Define of the data set is completed.
IDC0510I	DE0-11	IDCDE01	IDCDE01	Define of the VSAM or ICF catalog is completed.
IDC0511I	DE0-12	IDCDE01	IDCDE01	Define of the data space is completed.
IDC0512I	DE0-13	IDCDE01	IDCDE01	Define of the data set is completed.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC0520I	DE0-21	IDCDE01	IDCDE01	The message identifies the recovery volume serial number.
		IDCMP01	CLUSPROC	The message identifies the recovery volume serial number.
		IDCRM01	CLUSPROC	The message identifies the recovery volume serial number.
IDC0526I	AL0-1	IDCAL01	IDCAL01	Alter of the volume is completed.
IDC0531I	AL0-7	IDCAL01	IDCAL01	All altered entry names are listed.
IDC0532I	AL0-8	IDCAL01	IDCAL01	All entry names not altered are listed.
IDC0534I	AL0-10	IDCAL01	MEMRENAM	A member name not renamed is listed.
IDC0535I	AL0-11	IDCAL01	MEMRENAM	The listed member name has been renamed.
IDC0548I	DL0-10	IDCDL01	MEMDELET	The listed member name has not been deleted.
IDC0549I	DL0-11	IDCDL01	MEMDELET	The listed member name has been deleted.
IDC0550I	DL0-1	IDCDL01	CATCALL	The catalog returned the name and type of a successfully deleted entry in the catalog work area.
		IDCMP01	DELTPROC	The object with the same name as the object being imported was deleted successfully from the catalog.
			DELTPROC	The object being imported was deleted successfully from the catalog after an error occurred copying data into the object.
		IDCRM01	DELTPROC	The object with the same name as the object being imported was deleted successfully from the catalog.
			DELTPROC	The object being imported was deleted successfully from the catalog after an error occurred copying data into the object.
		IDCXP01	DELTPROC	The object being exported was deleted successfully from the catalog.
IDC0551I	DL0-8	IDCDL01	DELTPROC	A catalog object was not deleted because of a catalog locate error, a command parameter error, or a catalog delete error.

Message	STID	Module	Procedure	Situation That Caused Message
		IDCXP01	DELTPROC	The object being exported could not be deleted from the catalog. The catalog return code indicates the reason.
IDC0555I	DL0-5	IDCDL01	CATCALL	The volume entry was not deleted although empty space on the volume was deleted successfully. The catalog return code was 160.
IDC0571I	PR0-19	IDCRP01	IDCRP01	Reloading of a catalog was initiated.
IDC0594I	XP0-5	IDCXP01	CLUSPROC	The portable data set was created successfully.
IDC0603I	MP0-11	IDCMP01	CLUSPROC	The user catalog was connected successfully.
IDC0604I	MP0-12	IDCMP01	CLUSPROC	The first record of the portable data set contained the timestamp written at the time of export.
		IDCRM01	IDCRM01	The first record of each group of associated items on the portable data set contained the time and date of export.
IDC0622I	MP0-22	IDCRM01	UCATPROC	A user catalog has been disconnected successfully.
IDC0626I	MP0-26	IDCRM01	CLUSPROC UCATPROC NVSMPROC GDGPROC	IMPORTRA succeeded for the object named in the message.
IDC0634I	CC0-7	IDCCC01	CONVERT	CONVERT prints the number of OS/VS catalog entries that were converted to VSAM or ICF catalog entries.
IDC0635I	CC0-9	IDCCC01	CATALOG	The OS/VS catalog entry was not converted to a VSAM or ICF catalog entry.
IDC0636I	CC0-13	IDCCC01	CONVERT	CONVERT prints the number of non-VSAM entries in the VSAM catalog whose volume information was updated.
IDC0637I	CC0-11	IDCCC01	CONVERT	CONVERT prints the name of a VSAM catalog entry that was not updated to match an OS/VS catalog entry.
IDC0639I	CC0-15	IDCCC01	VSAMCNVT	Conversion of the specified sphere has started.
IDC0652I	BI0-13	IDCBI01	FINPROC	The alternate was built with no errors.
IDC0665I	LR1-16	IDCLR01	CLENCRA	Informational message stating the number of entries that did not compare.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC0669I	RC0-14	IDCRC01	IDCRC01	Informational message stating the CRA from which the entries are processed.
IDC0670I	RC0-15	IDCRC01	EXPORTDR	Informational message stating that data set is on portability data set.
IDC0672I	RC0-17	IDCRC01	CKCATNM	Informational message stating the catalog name for which CRAs are being processed.
IDC0674I	RC0-20	IDCRC01	EXPORTDR	Secondary message containing the object name for which the export driver was called.
			SYNCH	Object named was invalid in the CRA in comparison with the data set.
			DUPNAMCK	Object name appeared twice in the CRA.
IDC0676I	RC0-5	IDCRC01	TERM	Informational message stating that the portability data set was created successfully.
IDC0861I	CK0-6	IDCCK01	IDCCK01	No type 1 DSDR records for a tape data set were found.
IDC0862I	CK0-7	IDCCK01	BUILDTAB	The checkid identified was selected by the user more than once.
IDC0863I	CK0-8	IDCCK01	CHRPROC	A duplicate entry was found for a selected checkid.
IDC0874I	LR1-5	IDCLR01	INTSORT	Space could not be obtained for the sort table. The objects are printed first in, first out.
IDC0877I	LR1-8	IDCLR01	CLENCRA	Informational message stating the number of objects that did not compare.
IDC0888I	RC0-23	IDCRC01	EXPORTDR	Informational message stating that the exported entry contained no data.
IDC0896I	DL0-19	IDCDL01	CATCALL	HSM has deleted the entry.
IDC0922I	EX0-5	IDCDB02	ITEMDUMP	An invalid dump item was specified in the dump argument list.
IDC0923I	EX0-6	IDCDB02	ARRAYHDR	Invalid array header parameters were specified in the dump argument list.
IDCO924I	EX0-7	IDCDB01	IDCDB01	The dump routine was invoked through a UDUMP macro.

Message	STID	Module	Procedure	Situation That Caused Message
IDC0925I	EX0-8	IDCDB01	IDCDB01	A dump was requested through a UDUMP macro.
IDC0970I	VS0-22	IDCVS01	VTOCPUT	The number of tracks and the cylinder and head address could not be restored in the VTOC.
IDC1069I	CM0-57	IDCSA09	CHKCODE	High-order bit in reason code returned by SVC 126 was on, indicating an error updating inventory.
IDC1252I	RI1-16	IDCRI04	RISETUP	MAXID indicates no parameters are defined for this command, but ECTNOPD indicates that parameters were coded.
IDC1502I	DE0-5	IDCDE02	MODELPRC	Security information was suppressed when a model object was retrieved from the catalog.
IDC1543I	AL0-18	IDCAL01	CHECKPRC	New KEY/RECORDSIZE values equal to old default values.
IDC1544I	AL0-19	IDCAL01	CHECKPRC	New KEY/RECORDSIZE values equal to old nondefault values.
IDC1561I	LC1-2	IDCLC02	ANSVPROC	The UGPOOL request for a larger catalog work area failed. More space was required to process the associations.
			LOCPROC	The UGPOOL request for a larger catalog work area failed. A catalog entry required more space.
IDC1562I	LC1-3	IDCLC01	ENTPROC	Only space entries were requested; however, an entry in the entry list is greater than 6 characters.
IDC1564I	LC1-5	IDCLC01	RTEPROC	An entry retrieved from the catalog is not a type that can be listed.
IDC1565I	LC1-6	IDCLC01	ENTPROC	An entry retrieved from the catalog and specified in the user's entry list is not one of the types requested by the user.
IDC1566I	LC1-8	IDCLC01	ENTPROC	Either (1) the correct password was not supplied for a cluster or AIX entry and so the data and index and path association information could not be processed, or (2) the correct password was not supplied for an entry and the user requested more information than merely entry names, or (3) another type of catalog locate error occurred.

Message	STID	Module	Procedure	Situation That Caused Message
			GNXTPROC	Either the correct password was not supplied for an entry and the user requested more information than merely entry names, or another type of catalog locate error occurred.
			RTEPROC	Either (1) the correct password was not supplied for a cluster or AIX entry, and, even though the user requested only entry names, the names of the data and index or path association were not returned by the catalog, or (2) the correct password was not supplied for a data or index or path entry associated with a cluster or AIX entry, and field information other than entry names was not returned by the catalog, or (3) a nonsupported entry type was returned from the catalog.
IDC1567I	LC1-9	IDCLC01	RTEPROC	Retrieval of a data or index or path entry associated with a cluster or AIX entry was attempted, using the control interval number of the associated entry contained in the cluster or AIX entry. However, the entry could not be found in the catalog.
		IDCLC02	CDIPROC	Retrieval of a data or index or path entry associated with a cluster of AIX entry was attempted, using the control interval number of the associated entry contained in the cluster or AIX entry. However, the entry could not be found in the catalog.
			VPROC	Retrieval of the data set names associated with a data space was attempted using the control interval number of the associated entry contained in the data space. However, the entry could not be found in the catalog.
IDC1569I	LC1-12	IDCLC01	INITPROC	The EXPIRATION LISTCAT option was specified with entry type(s) that contain either no expiration date field or an expiration date field that is never initialized.
IDC1574I	PR0-22	IDCRP01	CATCOMP	More than 100 true name entries failed a comparison test during catalog reload. Processing continues but comparison does not.
IDC1575I	PR0-23	IDCRP01	CATCOMP	A true name record existed on a backup or target catalog without a corresponding record on the backup or target catalog.
IDC1595I	XP0-6	IDCXP01	CLUSPROC	Passwords were suppressed when the object to be exported was retrieved from the catalog.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC1597I	XP0-8	IDCXP01	CLUSPROC	The specified parameter is invalid for an ICF catalog.
IDC1631I	CC0-4	IDCCC01	CATALOG	VSAM catalog management gave a return code of 8 after trying to define the OS/VS catalog entry. The duplicate data set name started with 'SYS1.'.
IDC1632I	CC0-2	IDCCC01	CVPEPROC	The CVOLEQUATES parameter was missing or the volume serial number in the control volume pointer entry did not match a volume serial number in the CVOLEQUATES parameter.
IDC1638I	CC0-14	IDCCC01	GIPEPROC	A generation index pointer entry has been found and the "alias processing" flag is on.
IDC1644I	BI0-5 BI0-17	IDCBI01	SORTPROC	The base cluster record identified in the message was too short to contain the entire alternate key.
IDC1645I	BI0-6 BI0-18	IDCBI01	BLDPROC	Multiple occurrences of the same alternate key have been encountered in building an alternate index defined with the UNIQUEKEY attribute.
IDC1646I	BI0-7	IDCBI01	BLDPROC	The alternate index record identified in the message was too short to contain all the base cluster pointers.
IDC1653I	BI0-14	IDCBI01	FINPROC	The alternate index was built but nonterminating errors were encountered.
IDC1661I	RC0-6	IDCRC01	EXPORTDR	Informational message stating that the data set exported was out-of-synch.
IDC1662I	RC0-7	IDCRC01	EXPORTDR	Informational message stating that the data set was not exported and was out-of-synch.
IDC1663I	RC0-8	IDCRC02	CLUSPROC	Catalog field could not be located for a path to a VSAM cluster or an alias association to a non-VSAM data set or a non-VSAM association to a GDG.
IDC1664I	RC0-9	IDCRC02	NVSMPROC	Invalid or no association to a GDG for a non-VSAM data set.
IDC1667I	RC0-12	IDCRC01	OBJVOLCK	Volumes are out-of-synch because data set is not on both volumes.
IDC1678I	RC0-2	IDCRC01	EXPORTDR	An error occurred while processing an association for an object being exported.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC1679I	RC0-4	IDCRC01		The timestamps or CI of a multivolume data set were not equal.
IDC18601	CK0-5	IDCCK01	IDCCK01	The checkid selected by the user wasn't found in the checkpoint data set.
IDC1864I	CK0-9	IDCCK01	IDCCK01	No CHR records were found in the checkpoint data set.
IDC1865I	CK0-10	IDCCK01	DSDRVOLS	A type 2 DSDR record was expected but not found.
IDC1866I	CK0-11	IDCCK01	GETNEXT	End-of-data occurred in the checkpoint data set while DSDR records were being processed.
IDC1867I	CK0-12	IDCCK01	DSDRPROC	Volume sequence number for a tape data set exceeded the number volume of the volumes.
IDC1870I	LR1-1	IDCLR01	GETPRT	An I/O error occurred while reading the CRA.
		IDCLR02	IDCLR02	An I/O error occurred while reading the CRA.
IDC1871I	LR1-2	IDCLR01	GETPRT	An I/O error occurred while reading the catalog.
		IDCLR02	IDCLR02	An I/O error occurred while reading the catalog.
IDC1875I	LR1-15	IDCLR01	TCICTCR	The CI from the catalog record could not be found in the CTT table; therefore it could not be translated.
IDC1878I	LR1-9	IDCLR01	CATOPEN	IDCRC04 encountered an error while searching for the catalog name in the cluster record of the catalog.
			CKEYRNG	IDCRC04 encountered an error while searching for the high key value in a given CRA record.
			CRAOPEN	IDCRC04 encountered an error while searching for either the owning catalog name or the volume serial number in the CRA record.
			CTTBLD	IDCRC04 encountered an error while searching for the entry type of the catalog CI in the CRA record.
			GETPRT	IDCRC04 encountered an error while searching for the entry type or the entry name in the CRA record.
			INTASOC	IDCRC04 encountered an error while searching for the associated entry type or entry name fields in the CRA records.

Message	STID	Module	Procedure	Situation That Caused Message
			INTSORT	IDCRC04 encountered an error while searching for the name in a given CRA record.
			INTVEXT	IDCRC04 encountered an error while searching for the extension pointer in a given CRA record.
			PRTCMP	IDCRC04 encountered an error while searching for the used length field in a given CRA record.
			PRTDMP	IDCRC04 encountered an error while searching for the used length field in a given CRA record.
			PRTOJVL	IDCRC04 encountered an error while searching for the volume information or high key value in a given CRA record.
			PRTVOL	IDCRC04 encountered an error while searching for the volume timestamp information in a given catalog or CRA record.
IDC1880I	LR1-11	IDCLR01	PRTVOL	Timestamp for the format-4 record could not be read for the CRA volume.
IDC1881I	LR1-12	IDCLR01	INITLZE	Alternate output data set OPEN failed.
IDC1885I	LR1-17	IDCLR01	PRTMCWD	IDCRC04 encountered an error while searching for mismatched fields in a given CRA record. The CRA record had previously been read and had indicated that mismatches existed.
IDC1887I	RC0-22	IDCRC01	SCANCRA	I/O error encountered on a CRA record.
			TIMESTAMP	Volume timestamp could not be obtained.
IDC1890I	DL0-12	IDCDL01	CATCALL	Catalog management indicated that the RACF profile of the data set being deleted could not itself be deleted because it is ineligible for deletion.
IDC1891I	DL0-13	IDCDL01	CATCALL	When deleting a RACF indicated data set, the RACF profile could not be found.
IDC1927I	EX0-12	IDCPM01	MARGPARM	Margin values specified are invalid.
IDC1968I	VS0-11	IDCVS01	VTOCPUT	An error occurred during reading or updating of the VSAM time stamp.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
	VS0-19	IDCVS01	VTOCPUT	An error occurred during reading or updating of the VSAM time stamp.
	VS0-20	IDCVS01	VTOCPUT	An error occurred during reading or updating of the VTOC VSAM time stamp.
ICD1969I	VS0-12	IDCVS01	VTOCPUT	An error occurred during reading or updating of the VTOC and prevented the alternate track information from being restored after the copy of data.
IDC2011I	UV0-12	IDCSA08	IDCSA08	A function was requested that required data sets to be scratched, but not enough virtual storage was available to build the parameter list for the SCRATCH macro.
		IDCSA09	IDCSA09	There was insufficient storage for the ECB and the message area that are required for SVC 126.
		IDCDL01	MORESP	Insufficient main storage was available for a work area for VSAM catalog management.
		IDCVS03	GETBLK	Insufficient storage for volume VTOC processing.
IDC2035I	TP6-3	IDCTP06	IDCTP06	Invalid ERCNVTAB passed to UERROR.
IDC2100I	SA7-3	IDCSA07	GETENT	The return code for an OS locate request was neither 0 nor 8.
IDC2101I	SA7-4	IDCSA07	UPDATENT	The return code from an OS recatalog request was neither 0 nor 8.
IDC2360I	SA7-8	IDCSA07	IDCSALC	The ULOCATE function, after locating a data set name in the catalog, determined that either the data set resides on a different device type than the catalog indicates or a duplicate data set name exists.
IDC2364I	SA7-12	IDCSA07	VSAMLOC	The ULOCATE function, after locating a data set name in the catalog, determined it was not a non-VSAM data set. The situation was probably caused by a duplicate data set name.
IDC2370I	IO5-1	IDCIO05	READJFCB	The READJFCB routine was unable to read the JFCB entry.
IDC2371I	IO5-2	IDCIO05	BLDBLK	The storage required to perform I/O processing was unavailable.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC2372I	IO5-3	IDCIO05	CKOPN	The OPEN SVC encountered an error that prevented the opening of the DCB and further I/O processing.
IDC2373I	IO5-6	IDCIO05	RETRY	Invalid password.
IDC2374I	IO5-7	IDCIO05	CLOSESTD	The CLOSE SVC encountered an error that prevented the closing of the DCB and further I/O processing.
IDC2375I	IO5-8	IDCIO05	SYNAD	A SYNAD error occurred during EXCP I/O processing; the message indicates the data set or volume label.
	IO5-9	IDCIO05	SYNAD	A SYNAD error occurred during EXCP I/O processing; the message indicates the data set or volume label.
IDC2376I	IO5-7	IDCIO05	CLOSESTD	CLOSE SVC encountered an error which resulted in the CLOSE ABEND exit being entered.
IDC2381I	SA6-1	IDCSA06	UCBPOST	The UCB could not be updated because updating would cause duplicate label in the system.
IDC2382I	SA6-2	IDCSA06	UCBCHECK	The volume was not mounted for exclusive use because it was mounted when dynamic allocation was invoked.
IDC2386I	SA6-6	IDCSA06	MOUNTCTL	The previous volume could not be demounted; therefore, the assigned unit could not be used to mount needed volumes.
IDC2387I	SA6-7	IDCSA06	MOUNTCTL	The specified volume could not be mounted.
IDC2388I	SA6-8	IDCSA06	DEMNTVOL	The specified volume could not be demounted.
IDC2389I	SA6-9	IDCSA06	DEMNTVOL	The specified volume was demounted but data could not be destaged.
IDC2390I	SA6-10	IDCSA06	ENQDEQ	The volume serial number is shared; therefore, the specified volume could not be enqueued for exclusive use.
IDC2391I	SA6-11	IDCSA10	SETSTAE	The ESTAE macro returned a nonzero return code that prevented recovery protection in case of an abnormal termination.
IDC2399I	SA7-7	IDCSA07	IDCSALC	The ULOCATE function, after locating a data set name in the catalog, determined that either the data set resides on a different volume than the catalog indicates or a duplicate data set name exists.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC2533I	AL0-9	IDCAL01	MEMRENAM	A member name is specified with a generic name.
IDC2552I	DL0-2	IDCDL01	PARAMCHK	The type of the entry to be deleted was retrieved from the catalog, but the type is not one the user is allowed to delete.
IDC2553I	DL0-3	IDCDL01	PARAMCHK	The type of the entry to be deleted was retrieved from the catalog, but the type conflicts with the erase option.
IDC2557I	DL0-7	IDCDL01	PARAMCHK	The scratch option was specified for an object of an invalid type.
IDC2559I	DL0-9	IDCDL01	MEMDLETE	A generic name was specified when attempting to delete a partitioned data set member.
IDC2563I	LC1-4	IDCLC01	INITPROC	Either the allocation request conflicts with the type specification of cluster, PGSPC, AIX, PATH, ALIAS, or GDG, space, non-VSAM, or user catalog, or the volume request conflicts with the type specification of cluster, PGSPC, AIX, PATH, ALIAS, or GDG.
		IDCLC02	AUPROC	The allocation request conflicts with a non-VSAM, ALIAS, or GDG or user catalog entry specified in the entry list or volume request conflicts with an alias entry specified in the entry list.
			VPROC	The allocation request conflicts with a space (volume) entry specified in the entry list.
IDC2616I	MP0-16	IDCMP01	CLUSPROC	A path import operation failed.
		IDCRM01	CLUSPROC	A path import operation failed.
IDC2618I	MP0-18	IDCMP01	CLUSPROC	An invalid OBJECTS subparameter was found.
IDC2620I	MP0-20	IDCMP01	CLUSPROC	The object to be imported is a type not supported on this system.
		IDCCC01	VSAMCNVT	The catalog being converted contained an object not supported on this system.
		IDCRM01	CLUSPROC	The named object is a type not supported on this system.

Message	STID	Module	Procedure	Situation That Caused Message
IDC2621I	MP0-21	IDCRM01	CLUSPROC NVSAMPROC GDGPROC UCATPROC ALISPROC	The object named could not be imported.
IDC2630I	CC0-3	IDCCC01 IDCCC02	CATALOG DEFNVSM CVNTGDG DEFCLUS CNVTUCAT CNVTALIS CNVTPATH	VSAM catalog management gave a return code of 8 after trying to define the OS/VS catalog entry. A locate on the VSAM or ICF catalog indicated that the duplicate data set is a VSAM data set.
IDC2640I	BI0-1	IDCBI01	LOCPROC	The file identified via OUTFILE or OUTDATASET is not an alternate index.
IDC2642I	BI0-3	IDCBI01	LOCPROC	The alternate index identified in the message is not related to the base cluster identified via INFILE or INDATASET.
IDC2647I	BI0-8	IDCBI01	INITPROC	Storage was not available to obtain buffers and work areas.
IDC2648I	BI0-9	IDCBI01	JCPROC FINPROC	DD statements for sort work files are either missing or in error.
IDC2649I	BI0-10	IDCBI01	DEFPROC	A sort work area was obtained smaller than that required and job control for sort work files was missing or in error.
IDC2650I	BI0-11	IDCBI01	DEFPROC	An internal sort could not be completed and job control for sort work files was missing or in error.
IDC2651I	BIO-12	IDCBI01	DEFPROC	Define of sort work files failed.
IDC2654I	BI0-15	IDCBI01	FINPROC	The alternate index was not built due to severe errors.
IDC2655I	BI0-16	IDCBI01	CATPROC	Catalog information was not returned for a locate request.
IDC2656I	BI0-19	IDCBI01	CATPROC	A VSAM catalog locate failed with a nonzero return code.
IDC2660I	RC0-3	IDCRC01	CKNAMES	The object named is not a VSAM cluster or valid non-VSAM data set.
		IDCRC02	CLUSPROC	The object named is not a VSAM cluster.
			GDGPROC	The object named is not a GDG or has invalid associations to a GDG.

Message	STID	Module	Procedure	Situation That Caused Message
			NVSMPROC	The object named was not a non-VSAM data set or a user catalog.
IDC2666I	RC0-11	IDCRC01	SYNCH	The selected entry was not found in the selected CRA.
IDC2668I	RC0-13	IDCRC01	OBJVOLCK	A required volume was not supplied in the CRA keyword.
IDC2671I	RC0-16	IDCRC01	CKCATNM	The CRA has a different name than the others being processed.
IDC2673I	RC0-19	IDCRC01	BUILDCRV	Required information about the volume could not be obtained.
IDC2675I	RC0-21	IDCRC01	CKNAMES	The same name was found in more than one CRA.
IDC2677I	RC0-1	IDCRC01	EXPORTDR	The data set was not exported because of the error indicated in previous messages.
IDC2872I	LR1-3	IDCLR01	CRAOPEN	The catalog specified in the input for compare was not the owning catalog found in the CRA.
IDC2873I	LR1-4	IDCLR01	CATOPEN	Catalog could not be opened, therefore the compare option was ignored.
			CRAOPEN	The CRA opened belongs to a catalog other than the one specified in the compare.
IDC2876I	LR1-6	IDCLR01	CRAOPEN	A verify was issued after opening a CRA and it failed.
IDC2879I	LR1-10	IDCLR01	CATOPEN	IDCRC04 could not find the catalog name from the cluster record or the volume serial number of the catalog so it could not lock out all other usage of the CRA while it is being listed.
IDC2882I	LR1-13	IDCLR01	CTTBLD	LISTCRA encountered an error reading the catalog control record.
IDC2884I	LR1-7	IDCLR01	CATOPEN	A verify was issued after opening a catalog and it failed.
IDC2886I	RC0-18	IDCRC01	ERRCK	CRA cannot be opened because of some errors encountered.
IDC2895I	DL0-18	IDCDL01	CATCALL	All required volumes not included in DD statement specified in file parameter

Message	STID	Module	Procedure	Situation That Caused Message
IDC2908I	IO0-36	IDCIO03	DSINFO	No data set or volume is allocated under that indicated ddname, or no DD statement corresponding to the given ddname could be found.
		IDCSA08	IDCSA08	No volume is allocated under the indicated ddname, or no DD statement corresponding to the given ddname could be found.
IDC2909I	SA0-1	IDCSA08	IDCSA08	An error occurred while an attempt was being made to scratch the indicated data set.
IDC2910I	SA0-2	IDCSA08	IDCSA08	None of the volumes specified for the data set is mounted.
IDC2912I	SA0-4	IDCSA08	IDCSA08	The data set to be scratched is password-protected, and the operator did not supply the proper password.
IDC2913I	SA0-5	IDCSA08	IDCSA08	The OVERRIDE option wasn't specified, and the data set's retention period hasn't expired.
IDC2914I	SA0-6	IDCSA08	IDCSA08	The VTOC couldn't be read because of an I/O error.
IDC2915I	SA0-7	IDCSA08	IDCSA08	A required unit was not available for mounting.
IDC2916I	SA0-8	IDCSA08	IDCSA08	The data set to be scratched was in use.
IDC2917I	SA0-9	IDCSA08	IDCSA08	No RACF profile could be found for the specified data set.
IDC2918I	SA0-10	IDCSA08	IDCSA08	The user does not have the proper RACF authorization for the specified data set.
IDC2919I	SA0-11	IDCSA08	IDCSA08	The parameter list passed to URACHECK macro is invalid.
IDC2930I	SA0-12	IDCSA08	IDCSA08	The data set to be scratched is RACF-protected and the caller doesn't have the proper authorization.
IDC2950I	TP1-1	IDCTP01	IDCTP01	Either (1) no format list or static text identification was passed as input, or (2) no valid bits in FMTFLGS were turned on, or (3) the input or output length specified was less than 1.
IDC2951I	TP1-2	IDCTP01	IDCTP01	The output column specified is not within the print line.
IDC2952I	TP1-3	IDCTP01	BDCONV	For binary to decimal conversions, the input data length was more than 4 or the converted length was more than 16.

Message	STID	Module	Procedure	Situation That Caused Message
			PUPCONV	For packed to unpacked conversions, the converted length was more than 15, or the input data length was more than 8.
IDC2953I	TP1-4	IDCTP01	REDO	A REDO structure is nested.
IDC2954I	TP1-6	IDCTP05	IDCTP05	The requested static text entry was not in the specified module.
IDC2955I	TP1-7	IDCTP01	PUPCONV	An invalid packed decimal field was passed by the caller.
IDC2960I	VS0-2	IDCVS01	NVSAMCHK	A password-protected data set was found on the VTOC but no ddname with that data set name was specified in the PASSWORDFILE keyword.
IDC2961I	VS0-3	IDCVS01	FMT1CHK	The caller indicated that no non-VSAM data sets could reside on the volume. However, a format-1 DSCB was found for a non-VSAM data set.
	VS0-15	IDCVS01	FMT1CHK	The caller indicated that no non-VSAM data sets could reside on the volume. However, a format-1 DSCB was found for a non-VSAM data set.
	VS0-16	IDCVS01	FMT1CHK	The caller indicated that no non-VSAM data sets could reside on the volume. However, a format-1 DSCB was found for a non-VSAM data set.
IDC2962I	VS0-4	IDCVS01	CATCHK	The volume entry could not be found in the VSAM catalog.
IDC2963I	VS0-5	IDCVS01	FMT4READ	The volume services module positioned to the first record in the VTOC, but it was not a format-4 DSCB.
	VS0-17	IDCVS01	FMT4READ	The volume services module positioned to the first record in the VTOC, but it was not a format-4 DSCB.
	VS0-18	IDCVS01	FMT4READ	The volume services module positioned to the first record in the VTOC, but it was not a format-4 DSCB.
IDC2964I	VS0-6	IDCVS01	SCRATCH	An error occurred that prevented the scratching of data sets. The volume is still flagged as a VSAM volume.
IDC2965I	VS0-7	IDCVS01	SCRATCH	An error occurred reading or updating the VTOC that prevented the scratching of any more data sets.
IDC2966I	VS0-8	IDCVS01	RECATLOG	An I/O error occurred that prevented any recataloging of non-VSAM data sets.

Message	STID	Module	Procedure	Situation That Caused Message
IDC2967I	VS0-10	IDCVS01	RECATLOG	An I/O error occurred that terminated the recataloging of data sets.
IDC2971I	VS0-21	IDCVS01	OPENVTOC	A reserve for a volume with the HAVE option was unsuccessful.
IDC2972I	VS0-23	IDCVS03	USAGE	A nonzero return code was received from LSPACE.
IDC3003I	UV0-3	IDCAL01	IDCAL01	The VSAM or ICF catalog could not be opened, or another severe error occurred.
		IDCBI01	TERMPROC	Either (1) a severe error was encountered in processing the base cluster, or (2) the EXTERNALSORT parameter was specified but the job control for sort files was missing or in error.
		IDCCC01	IDCCC01	A severe error occurred. Conversion of the OS catalog was not attempted or terminated if begun.
		IDCDE01	IDCDE01	The VSAM catalog to contain the defined object could not be opened, or another severe error occurred.
			MODELPRC	The VSAM catalog containing the model object could not be opened.
		IDCLC01	IDCLC01	A severe error occurred. Listing of the catalog was not attempted or terminated if begun.
		IDCMP01	IDCMP01	A severe error occurred.
		IDCPR01	IDCPR01	Either (1) an error occurred opening the input or alternate output data sets, or (2) an unrecoverable error occurred while retrieving or printing a record, or (3) more than three I/O errors occurred while retrieving records.
			TEXTPSET	The static text subtitle line could not be retrieved.
			DELIMSET	An incompatible use of delimiters was found during a data set print operation.
		IDCRC01	EXITTHE	Function was not completed because a severe error was encountered.
		IDCRM01	IDCRM01	A severe error occurred.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
		IDCRP01	IDCRP01	Either (1) an error occurred opening the input or output data sets, or (2) an unrecoverable error occurred while copying the data set, or (3) more than three I/O errors occurred while copying the data set, (4) an error occurred while attempting a catalog reload, or (5) a nonrelative record input data set did not have a nonempty relative record output data set.
			DELIMSET	An incompatible use of delimiters was found during a data set copy operation.
		IDCRS05	CKERR	A severe error occurred which prevented further processing.
		IDCVY01	IDCVY01	The VSAM data set to be verified could not be opened, or the verify was not successful.
		IDCXP01	IDCXP01	A severe error occurred.
IDC3004I	UV0-4	IDCAL01	ALTERPRC	Storage was not available for one of the following: the volume list or the PASSWALL field.
			DALCPROC	Storage was not available for volume list.
			IDCAL01	Storage was not available for the CTGPL, CTGFV, and CTGFLs.
			INDEXPRC	Storage was not available for the index parameter list if KEYS was specified.
			LOCATPRC	Storage was not available for the catalog work area.
		IDCCC01	CNVTINIT	Storage was not available for: 22 levels of OS/VS catalog index blocks, VSAM parameter list, data addressed by the VSAM parameter list, or a VSAM catalog work area.
		IDCCC01	CNVTINIT	Storage was not available for 22 levels of OS/VS catalog index blocks.
		IDCCC02	CTLGPROC	Storage was not available for a larger VSAM catalog work area.
			ASSOCTNS ASSOCNS	Storage was not available for the associations list.
			RANGPROC	Storage was not available for the range list.

Message	STID	Module	Procedure	Situation That Caused Message
			LVLRPROC	Storage was not available for the volume serial list, the device types list, or the file sequence number list.
			DELSPACE	Storage was not available for the data space deleted list.
			DALCPROC	Storage was not available for the volume serial list.
		IDCDE01	IDCDE01	Storage was not available for the CTGPL and CTGFVs.
			DALCPROC	Storage was not available for volume list.
		IDCDE02	ALLCPROC	Storage was not available for one of the following: CTGFLs, the volume list, the file sequence list, or the device type list.
			KEYPROC	Storage was not available for one of the following: the AMDSBCAT CTGFL and the AMDSBCAT field, or the key range list.
			MODELPRC	Storage was not available for the catalog parameter list or the catalog work area.
			NAMEPROC	Storage was not available for the CTGFLs.
			PROTPROC	Storage was not available for the CTGFLs needed to set up the protection attributes.
		IDCIO01	IDCRIOCR	Storage was not available for UCRYPT.
			PUTREP	Storage was not available for the input work area.
		IDCIO02	BUILDACB	Storage was not available for the ACB or the EXLST.
			BUILDDBK	Storage was not available for the required I/O areas.
			BUILDRPL	Storage was not available for the input work area or the RPL.
			CKNONOP	No storage is available for the input work area required to process spanned, non-VSAM records.
			DSDATA	No space available to read the label cylinder.
			OPENRTN	Storage was not available for the IOCSTR.

Message	STID	Module	Procedure	Situation That Caused Message
		IDCLC01	INITPROC	Storage was not available for one of the following: catalog parameter lists, catalog work areas, or the static text used in the catalog listing.
		IDCLR01	ADDASOC	Storage was not available for the association table extension.
			BLDVEXT	Storage was not available for the VEXTTBL extension.
			CTTBLD	Storage was not available for the CI translate table.
			INITLZE	Storage was not available for the initial ASSOCTBL and VEXTTBL.
			INTASOC	Storage was not available for the association table extension.
		IDCMP01	FPLPROC	Storage was not available for CTGFLs.
			BPASPROC	Storage was not available for the PASSWALL field.
			CLUSPROC	Storage was not available for the catalog work area.
			CPLPROC	Storage was not available for the CTGPL.
			CTLPROC	Storage was not available for the catalog work area.
			DELTPROC	Storage was not available for the catalog work area.
			FVTPROC	Storage was not available for the CTGFV.
			LVLPROC	Storage was not available for one of the following: the catalog work area, CTGFLs, or volume serial lists.
		IDCPM01	TESTPARM	Storage was not available for the test option data area.
		IDCRC01	IDCRC01	Storage was not available for one of the tables required by EXPORTRA.
		IDCRC02	CLUSPROC	Storage was not available for the control record output buffer.
			CTLGPROC	Storage was not available for the catalog work area.
			GDGPROC	Storage was not available for the control record output buffer.

Message	STID	Module	Procedure	Situation That Caused Message
			LOCPROC	Storage was not available for the CPL, FPL, and the catalog work area.
			NVSMPROC	Storage was not available for the control record output buffer.
			SAVEPROC	Storage was not available for the input record save area.
		IDCRI01	GETSPACE	Storage was not available for the FDT.
			IDCRI02	Storage was not available for one of the following: work space or the FDT.
			INREPEAT	Storage was not available for the FDT.
			RIINIT	Storage was not available for the historical data area.
			SCANCMD	Storage was not available for the FDT.
		IDCRM01	ALISPROC	Storage was not available for the catalog data record buffer.
			BFPLPROC	Storage was not available for the FPLs.
			BPASPROC	Storage was not available for the PASSWALL information.
			CLUSPROC	Storage was not available for the buffer area or volume list.
			CPLPROC	Storage was not available for the catalog parameter list.
			CTLGPROC	Storage was not available for the catalog work area.
			DELTPROC	Storage was not available for the catalog work area.
			FVTPROC	Storage was not available for the FVT or FPLs.
			GDGPROC	Storage was not available for the catalog data record.
			LVLRPROC	Storage was not available for the volume serial list, the device types list, or the file sequence number list.
			NFVTPROC	Storage was not available for the FVT or FPLs.
			NVSMPROC	Storage was not available for the catalog data record.
			RANGPROC	Storage was not available for the range list.

Message	STID	Module	Procedure	Situation That Caused Message
			UCATPROC	Storage was not available for the catalog data record.
		IDCRP01	CRYPTSET	Storage was not available for the CRYPTAGL ship list.
		IDCRS01	IDCRS01	Storage was not available for automatic storage for modules IDCRS01 through IDCRS07.
			INIT	Storage was not available for any one of the following: record access blocks, Umacro parameter lists, control interval translate table, and UIOINFO return area.
		IDCRS03	GETTAB	Storage was not available for tables needed for association checking.
			PROCVOL	Storage was not available for work areas for bit maps.
			VERB	Storage was not available for GDG association checking.
		IDCRS04	NINUT	Storage was not available for FIND processing.
			NXPND	Storage was not available for the extension to the FIND work area.
		IDCRS05	BLDRLST	Storage was not available for the RESVOL table.
			BLDVLST	Storage was not available for the VOLSERTB table.
		IDCRS06	WFDEF	Storage was not available for the work area used for the UCATLG parameter list to define the work file.
		IDCRS07	RENMSETV	Storage was not available for the CAMLST area for RENAME operations.
			SCRATCHP	Storage was not available for the volume list for the SCRATCH macro.
		IDCXP01	ALTRPROC	Storage was not available for the CTGFV.
			CLUSPROC	Storage was not available for the control record output buffer.
			CTLGPROC	Storage was not available for the second catalog work area obtained when the first work area was too small.
			DELTPROC	Storage was not available for the CTGPL or the catalog work area.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
			LOCPROC	Storage was not available for the CTGPL or the catalog work area.
			MORESP	Storage was not available for the catalog work area.
IDC3006I	UV0-6	IDCPR01	DELIMSET	Beginning positioning failed.
		IDCRP01	DELIMSET	Beginning positioning failed.
IDC3007I	(See note at end of list)	IDCAL01	IDCAL01	The catalog return code was nonzero for an alter request.
			CHECKPRC	The catalog return code was nonzero for a locate request.
			LOCATPRC	The catalog return code was nonzero for a locate request.
		IDCBI01	FINPROC	The catalog return code was nonzero for a locate request against the base cluster or alternate index, or for a define request for external sort work files.
		IDCDE01	IDCDE01	The catalog return code was nonzero for a define request.
		IDCDE02	MODELPRC	The catalog return code was nonzero for a request to locate a model object.
		IDCDL01	CATCALL	The catalog return code was nonzero for a delete request. This message is not issued for a return code of 160, however, because 160 indicates a normal condition.
			FINDTYPE	The catalog return code was nonzero for a locate request.
		IDCLC02	LOCPROC	The catalog return code was nonzero for a locate request.
		IDCMP01	CTLGPROC	The catalog return code was nonzero.
			DELTPROC	The catalog return code was nonzero.
		IDCRC02	CTLGPROC	The catalog return code was nonzero for a locate request.
		IDCRM01	CTLGPROC	The catalog return code was nonzero for a define or alter request.
			DELTPROC	The catalog return code was nonzero for a delete request.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
		IDCXP01	CTLGPROC	The catalog return code was nonzero for a delete, alter, or locate request.
			DELTPROC	The catalog return code was nonzero for a delete request.
		IDCRS01	INIT	The catalog return code was nonzero for a locate request.
		IDCRS06	WFDEF	The catalog return code was nonzero for a define request.
			WFDEL	The catalog return code was nonzero for a delete request.
IDC3008I	UV0-8	IDCSA02	IDCSA02	A UPROMPT or UQUAL macro was issued but access method services was not invoked interactively with TSO.
IDC3009I	(See note at end of list)	IDCAL01	IDCAL01	The catalog return code was nonzero for an alter request.
			CHECKPRC	The catalog return code was nonzero for a locate request.
			LOCATPRC	The catalog return code was nonzero for a locate request.
		IDCBI01	FINPROC	The catalog return code was nonzero for a locate request against the base cluster or alternate index, or for a define request for external sort work files.
		IDCCC01	CATALOG	The VSAM catalog management return code was nonzero for a define, locate, or alter request.
		IDCDE01	IDCDE01	The catalog return code was nonzero for a define request.
		IDCDE02	MODELPRC	The catalog return code was nonzero for a request to locate a model object.
		IDCDL01	CATCALL	The catalog return code was nonzero for a delete request. This message is not issued for a return code of 160, however, because 160 indicates a normal condition.
			FINDTYPE	The catalog return code was nonzero for a locate request.
		IDCLC02	LOCPROC	The catalog return code was nonzero for a locate request.
		IDCMP01	CTLGPROC	The catalog return code was nonzero.
			DELTPROC	The catalog return code was nonzero for a delete request.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
		IDCRC02	CTLGPROC	The catalog return code was nonzero for a locate request.
		IDCRM01	CTLGPROC	The catalog return code was nonzero for a define or alter request.
			DELTPROC	The catalog return code was nonzero for a delete request.
		IDCRS01	INIT	The catalog return code was nonzero for a locate request.
		IDCRS06	WFDEF	The catalog return code was nonzero for a define request.
			WFDEL	The catalog return code was nonzero for a delete request.
		IDCSA02	IDCSA02	During a UCIR macro, the VSAM catalog return code was nonzero for a generic locate.
		IDCXP01	CTLGPROC	The catalog return code was nonzero for a delete, alter, or locate request.
			DELTPROC	The catalog return code was nonzero for a delete request.
IDC3010I	UV0-11	IDCXP01	RECPROC	The file identified in the INFILE parameter does not match that given in the EXPORT command or any paths over it.
		IDCLC02	ERRPROC	The locate function found that the name given for listing isn't in the specified catalog.
		IDCSA02	IDCSA02	The locate function found that the name given for a generic locate isn't in the specified catalog.
IDC3012I	TP6-9	IDCTP06	CATERCNV	Verbalization of catalog return code 8, Locate.
IDC3013I	TP6-10	IDCTP06	CATERCNV	Verbalization of catalog return code 8, Define.
IDC3014I	TP6-11	IDCTP06	CATERCNV	Error during catalog operation.
IDC3016I	TP6-12	IDCTP06	CATERCNV	Verbalization of catalog return code 4.
IDC3017I	TP6-13	IDCTP06	CATERCNV	Verbalization of catalog return code 20.
IDC3018I	TP6-14	IDCTP06	CATERCNV	Verbalization of catalog return code 56.
IDC3019I	TP6-15	IDCTP06	CATERCNV	Verbalization of catalog return code 60.
IDC3020I	TP6-16	IDCTP06	CATERCNV	Verbalization of catalog return code 68.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC3021I	TP6-17	IDCTP06	CATERCNV	Verbalization of catalog return code 72.
IDC3022I	TP6-18	IDCTP06	CATERCNV	Verbalization of catalog return code 80.
IDC3023I	TP6-19	IDCTP06	CATERCNV	Verbalization of catalog return code 84.
IDC3024I	TP6-21	IDCTP06	CATERCNV	Verbalization of catalog return code 148.
IDC3025I	TP6-22	IDCTP06	CATERCNV	Verbalization of catalog return code 156.
IDC3026I	TP6-23	IDCTP06	CATERCNV	Verbalization of catalog return code 172.
IDC3027I	TP6-24	IDCTP06	CATERCNV	Verbalization of catalog return code 176.
IDC3028I	TP6-25	IDCTP06	CATERCNV	Verbalization of catalog return code 184.
IDC3029I	TP6-26	IDCTP06	CATERCNV	Verbalization of catalog return code 192.
IDC3030I	TP6-27	IDCTP06	CATERCNV	Verbalization of catalog return code 196, 200.
IDC3031I	TP6-28	IDCTP06	CATERCNV	Verbalization of catalog return code 204.
IDC3032I	TP6-29	IDCTP06	CATERCNV	Verbalization of catalog return code 208.
IDC3033I	TP6-30	IDCTP06	CATERCNV	Verbalization of catalog return code 248.
IDC3067I	CM0-41	IDCVS01	FMT4CHK	The caller indicated that the volume could not be owned by a VSAM catalog; however, the VSAM ownership bit was set.
	CM0-42	IDCVS01	FMT4CHK	The caller indicated that the volume couldn't be owned by a VSAM catalog; however, the VSAM ownership bit was set.
	CM0-43	IDCVS01	FMT4CHK	The caller indicated that the volume couldn't be owned by a VSAM catalog; however, the VSAM ownership bit was set.
IDC3070I	CM0-36	IDCVS02	PUTLAB	The volume serial number could not be updated on the volume label.
	CM0-37	IDCVS02	PUTLAB	The volume serial number could not be updated on the volume label.
	CM0-38	IDCVS02	PUTLAB	The owner name on the volume label could not be updated.
	CM0-39	IDCVS02	PUTLAB	The owner name on the volume label could not be updated.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
	CM0-52	IDCVS02	PUTLAB	Neither the volume serial number nor the owner name could be updated.
	CM0-53	IDCVS02	PUTLAB	Neither the volume serial number nor the owner name could be updated.
IDC3190I	AL0-24	IDCAL01	PARAMCHK	One of the parameters specified on the command is invalid for the entry type.
IDC3191I	AL0-25	IDCAL01	IDCAL01	A VVDS name or its generic form has been specified for the entryname of an ALTER command.
		IDCAL01	IDCAL01	A VVDS name has been specified for the NEWNAME parameter of an ALTER command.
IDC3200I	RI0-1	IDCRI01	SCANCMD	The number of positional parameters found (PPARMCNT) exceeds the number defined in the descriptor for the current subparameter list (SUBCOUNT).
IDC3201I	RI0-2	IDCRI01	BUILDFDT	The input constant length (UNITINDX) exceeds the maximum length defined by the descriptor.
			CONVERT	The input constant length (UNITINDX) exceeds the maximum length defined by the descriptor.
			NXTFIELD	The input constant length (UNITINDX) exceeds the maximum length that the reader/interpreter can handle (UNITMAX).
			PACKCVB	The input constant length (UNITINDX) exceeds the maximum length defined by the descriptor.
IDC3202I	RI0-3	IDCRI01	ERROR1	The remainder of a command was bypassed due to an error in it.
			ERROR2	The remainder of a command was bypassed due to an error in it.
IDC3203I	RI0-4	IDCRI01	DSIDCHK	A data set name does not have the correct syntax.
IDC3205I	RI0-6	IDCRI01	SCANCMD	The closing parenthesis of a subparameter list was found before any parameters were found in the list, or an opening parenthesis was found before any keyword was found.
IDC3207I	RI0-8	IDCRI01	ERROR1	A severe error occurred. The condition code is set to 16, and the reader/interpreter will terminate processing.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
			ERROR2	A severe error occurred. The condition code is set to 16, and the reader/interpreter will terminate processing.
IDC3208I	RI0-9	IDCRI01	KWDPARM	A keyword parameter, defined as having a subfield, does not have a left parenthesis following the keyword.
IDC3209I	RI0-10	IDCRI01	KWDPARM	A keyword's subfield does not have a closing parenthesis following it.
			POSPARM	A list of constants is not delimited on the right by a closing parenthesis.
IDC3210I	RI0-11	IDCRI01	INREPEAT	The next repetition of a repeated subparameter list does not begin with a left parenthesis.
IDC3211I	RI0-12	IDCRI01	KWDPARM	The descriptor does not define the input keyword as part of the current parameter list.
			NXTFIELD	An input keyword exceeds the maximum allowable length for a keyword.
IDC3212I	RI0-13	IDCRI01	POSPARM	A positional parameter that is not defined as a list begins with a left parenthesis.
IDC3213I	RI0-14	IDCRI01	SETFLAG	An internal table (PARMFLAG) indicates that the keyword just found was found previously in this command.
IDC3214I	RI0-15	IDCRI01	GETDATA	A numeric constant begins with a B or X, but an apostrophe does not follow directly after this character.
IDC3216I	RI0-17	IDCRI01	ERROR1	The remainder of a command, being scanned for syntax-checking purposes only, was bypassed due to an error in it.
			ERROR2	The remainder of a command, being scanned for syntax-checking purposes only, was bypassed due to an error in it.
IDC3217I	RI0-18	IDCRI01	GETQUOTD	A password-delimiting slash appears following a constant that does not allow a password.
			GETSIMPL	A password-delimiting slash appears following a constant that does not allow a password.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC3218I	RI0-19	IDCRI01	INREPEAT	The number of sublist repetitions (REPCOUNT) for the current repeated sublist exceeds the maximum repetitions allowed (REPMAX) for this parameter according to the descriptor.
IDC3219I	RI0-20	IDCRI01	IDCRI02	The input verb name does not match any name in IDCRILT.
IDC3220I	RI0-21	IDCRI01	CONVERT	A numeric constant contains an invalid digit.
			PACKCVB	A numeric constant contains an invalid digit.
IDC3221I	RI0-22	IDCRI01	CONVERT	A numeric constant has a value outside the value range specified in the descriptor for this parameter.
			PACKCVB	A numeric constant is too large to fit into a binary fullword.
IDC3223I	RI0-24	IDCRI01	BUILDFDT	The number of constants found in a list (SCLRCNT) exceeds the number allowed (LISTMAX).
IDC3225I	RI0-26	IDCRI01	NEEDNOTS	A parameter always required for this command is missing, or parameter required when another parameter is coded is missing.
IDC3226I	RI0-27	IDCRI01	NEEDNOTS	An input parameter conflicts with some other input parameter.
IDC3240I	RI1-1	IDCRI04	BUILDFDT	PDENVMBR is larger than HIVALUE or smaller than LOWVALUE.
IDC3241I	RI1-2	IDCRI04	BUILDFDT	The length of a data set name or a ddname is wrong.
IDC3242I	RI1-3	IDCRI04	BUILDFDT	The length of a constant is wrong.
IDC3243I	RI1-4	IDCRI04	GETSPACE	There are too many constants in a list.
			REPLIST	There are more subparameter sets in a list than the command descriptor allows.
IDC3244I	RI1-5, 11	IDCRI04	NOTPARMS	Two conflicting parameters are coded.
IDC3246I	RI1-8	IDCRI04	RESOLVE	The TSO terminal user needs to choose between conflicting parameters.
IDC3247I	RI1-6	IDCRI04	NEEDPRMS	Parameters required by the command descriptor are missing.

Message	STID	Module	Procedure	Situation That Caused Message
IDC3248I	RI1-7	IDCRI04	GETSPACE	There are too many numbers in a list.
IDC3249I	RI1-9	IDCRI04	BUILDFDT	The TSO terminal user needs to reply with a correct constant.
IDC3250I	RI1-10	IDCRI04	TOOMANY	The TSO terminal user needs to choose whether or not to ignore excessive constants or subparameters.
IDC3251I	RI1-15	IDCRI04	IDCRI04	Parameter in error is a repeated subparameter.
IDC3253I	RI1-17	IDCRI04	ADDPARM	A UPROMPT to obtain a missing keyword has failed.
IDC3300I	IO0-1	IDCIO02	BLDOCMSG	An error occurred during open of a data set.
	IO05-4	IDCIO05	CHECKOPN	An error occurred during the open of a data set or the VTOC.
IDC3301I	IO0-2	IDCIO02	BLDOCMSG	An error occurred during close of a data set.
IDC3302I	IO0-3	IDCIO01	BLDAMSG	An error occurred while accessing a data set.
		IDCIO03	BLDAMSG	An error occurred while accessing a data set.
		IDCR506	RECMGMT	A logical error occurred during an I/O operation to the work file.
IDC3303I	IO0-4	IDCIO02	BUILDDBK	The data set to be opened for update processing is not a VSAM data set.
IDC3304I	IO0-5	IDCIO02	DSDATA	A job control statement specified for file to OPEN was not found.
IDC3305I	IO0-6	IDCIO02	DSDATA	An attempt was made to open an ISAM data set for output.
IDC3306I	IO0-7	IDCIO02	BUILDDBK	Cannot open an ISAM file for address processing.
			DSDATA	The data set to be opened for physical sequential processing is an ISAM data set.
IDC3307I	IO0-8	IDCIO02	BUILDDBK	The data set to be opened for keyed processing is not a VSAM or ISAM data set.
IDC3308I	IO0-10 IO0-35	IDCIO01	VSAMERR	A record with the same key or relative record number as the input record already exists in the output data set.
IDC3309I	IO0-12	IDCIO01	PUTNONVS	The length for a record to be written is invalid.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
			PUTVSAM	For a relative record data set, the length of the record to be written is not equal to the record size of the data set.
IDC3310I	IO0-13	IDCIO03	PTAMDS	The key provided is longer than the key length of the data set.
			PTISDS	The key provided is longer than the key length of the data set.
IDC3311I	IO0-14	IDCIO03	IDCIO03	The data set to be positioned is not a VSAM or ISAM data set.
IDC3312I	IO0-15	IDCIO02	CKNONOP	The DCB OPEN flag was not set by the system OPEN routines for magnetic tape or for a sequential disk file.
IDC3313I	IO0-16	IDCIO01	GETNONVS	An I/O error occurred while reading a non-VSAM data set.
			PUTNONVS	An I/O error occurred while writing a non-VSAM data set.
	IO5-5	IDCIO05	SYNAD	There was a SYNAD error in EXCP I/O processing.
IDC3314I	IO0-17	IDCIO01	VSAMERR	The record to be written has a lower key than the last record in the data set.
IDC3315I	IO0-44	IDCIO02	BUILDDBK	The record length exceeds 32767.
IDC3316I	IO0-19	IDCIO02	BUILDDBK	The data set to be opened is not a VSAM catalog.
IDC3317I	IO0-20	IDCIO01	VSAMERR	Physical error detected in a VSAM file.
		IDCIO02	DSDATA	I/O attempting to read the label cylinder.
		IDCIO03	PTAMDS	Physical error detected by VSAM POINT routines.
IDC3318I	IO0-21	IDCIO02	BUILDDBK	(1) Invalid environment or DLBL/TLBL parameters specified, (2) Block size is less than one, (3) Block size is invalid for a fixed length record format file, or (4) Block size is invalid for a variable length record format file.
			CKNONOP	The block size specified for an ISAM file is less than the file's true block size.
			DSDATA	Invalid parameters specified on the DLBL/TLBL statement.
IDC3321I	IO0-24	IDCIO02	CKNONOP	An open ABEND error was detected.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
			ENVFREE	A close ABEND error was detected.
IDC3322I	IO0-25	IDCIO01	IDCIOVY	The data set to be verified is not a VSAM data set.
IDC3325I	IO0-45	IDCIO01	IRSISYN	The block size specified for the portable data set is different than that of the portable data set.
IDC3326I	IO0-46	IDCIO02	OPENRTN	The REPLACE option has been specified for output through a path.
IDC3327I	IO0-47	IDCIO01	VSAMERR	Duplicate record in the upgrade set.
IDC3330I	IO0-18	IDCIO03	STOWRTN	Name doesn't exist in the partitioned data set directory.
IDC3331I	IO0-22	IDCIO03	STOWRTN	Name already exists in the partitioned data set directory.
IDC3332I	IO0-26	IDCIO02	BUILDACB	Storage was not available for one of the following: the ACB, the EXLST or area for a modified JFCB.
			BUILDDBK	Storage was not available for one of the following: the DCB, an update DECB, an area for a modified JFCB and DCB exit list.
			BUILDRPL	Storage was not available for one of the following: the input work area or the RPL.
			CKNONOP	No storage was available for an input work area.
			OPENRTN	Storage was not available for the IOCSTR.
		IDCIO03	STOWRTN	No storage available for USTOW macro.
		IDCSA02	IDCSA02	VSAM generic locate requires more storage than the UCIR macro can obtain.
IDC3333I	IO0-32	IDCIO01	COPYCAT	Output VSAM catalog is not empty.
IDC3334I	IO0-33	IDCIO01	COPYCAT	Invalid control interval number in VSAM catalog.
IDC3335I	IO0-51	IDCIO01	CSMSG	Encipher/decipher error.
IDC3336I	IO0-52	IDCIO01	CSMSG	Cryptographic request return code.
IDC3337I	IO0-53	IDCIO01	IDCIOCR	Nonempty target data set for encipher/decipher.
IDC3338I	IO0-54	IDCIO01	CRYPTRH	Data set to be deciphered was not enciphered by REPRO.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC3340I	IO0-56	IDCIO01	CRYPTRH	Information expected in encipher data set header was not present.
IDC3341I	IO0-57	IDCIO01	IDCIOCR CRYPTRH	Incorrect key for decipher.
IDC3350I	IO0-11	IDCIO02	DSDATA	Cannot open ISAM data set for output.
		IDCIO03	PTAMDS	An I/O error occurred during a VSAM point operation.
		IDCIO01	VSAMERR	An I/O error occurred in the VSAM access method.
IDC3351I	IO0-9	IDCIO01	VSAMERR	An error was detected by a VSAM macro. The error was not a duplicate record or a record out of sequence.
		IDCIO02	CLOSERTN	The ACB was not closed successfully.
			OPENRTN	The ACB was not opened successfully.
		IDCIO03	PTAMDS	A logical error occurred during a VSAM point operation.
		IDCRS06	RECMGMT	A logical error occurred during an I/O operation to the work file.
IDC3380I	SA6-13	IDCSA06	UCBCHECK	The unit for the specified ddname is not assigned for exclusive control. Therefore, open could not mount or demount volumes on this unit. Either the volume was already mounted for the job when this job was allocated, or the correct JCL parameters were not specified on the DD statement.
IDC3383I	SA6-3	IDCSA06	UCBCHECK	The UCB does not specify a virtual unit.
IDC3384I	SA6-4	IDCSA06	UCBCHECK	The UCB is not for a 3330 direct access device.
IDC3385I	SA6-5	IDCSA06	GETEXCL	The UCB indicates that the unit is allocated to a permanently resident or reserved volume.
IDC3392I	SA6-12	IDCSA06	FINDEXCL	The test enqueue indicates the volume is allocated exclusively to the job step, but no unit is available for mounting the volume. All nonshareable units have open VSAM or ICF catalogs.
IDC3500I	DE0-3	IDCDE03	IDCDE03	The object parameter list supplied by the user is incorrect.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC3501I	DE0-4	IDCDE02	MODELPRC	The entry type of a model object is not the same as that of the object being defined, or the entry type of a model object conflicts with the specification of INDEXED, NONINDEXED, or NUMBERED.
IDC3503I	DE0-1	IDCDE02	ALLCPROC	The number of elements in the volume list does not match the number of elements in the file sequence list.
IDC3504I	DE0-2	IDCDE02	KEYPROC	The length of the key range list retrieved from a model exceeded the space allotted for the list by IDCDE01.
IDC3505I	DE0-6	IDCDE01	INTGCHK	Space allocation was incorrectly specified for a VSAM or ICF catalog, data set, alternate index, or data space.
IDC3506I	DE0-7	IDCDE01	INTGCHK	Volumes were not correctly specified for a VSAM data set or alternate index.
IDC3507I	DE0-8	IDCDE01	INTGCHK	The record size was required but not specified for a VSAM data space.
IDC3513I	DE0-14	IDCDE01	IDCDE01	Dynamic allocation error.
IDC3514I	DE0-15	IDCDE02	KEYPROC	The key ranges specified by the user overlap.
		IDCMP01	RANGPROC	The key ranges specified by the user overlap.
IDC3515I	DE0-16	IDCDE02	ALLCPROC	The average record size exceeds the maximum record size.
IDC3516I	DE0-17	IDCDE01	INTGCHK	Key length and position were not specified for a key sequenced data set.
IDC3517I	DE0-18	IDCDE02	INTGCHK	Unequal record sizes were specified for a relative record data set.
IDC3518I	DE0-19	IDCDE01	INTGCHK	REUSE cannot be specified with UNIQUE or KEYRANGES.
IDC3519I	DE0-20	IDCDE01	INTGCHK	A REUSE conflict exists between data and index.
IDC3521I	DE0-22	IDCDE01	INTGCHK	A RECORDSIZE greater than 32761 was specified for a nonspanned data set.
IDC3522I	DE0-23	IDCDE01	INTGCHK	SPANNED cannot be specified for a relative record data set.

Message	STID	Module	Procedure	Situation That Caused Message
IDC3523I	DE0-24	IDCDE02	NAMEPROC	Name specified for generation data group exceeds 35 characters.
IDC3524I	DE0-25	IDCDE01	INTGCHK	Key range values are longer than key length.
		IDCDE02	KEYPROC	Key ranges are not in ascending order.
IDC3525I	AL0-23	IDCAL01	CHECKPRC	The password supplied is insufficient to alter key values.
IDC3527I	AL0-3	IDCAL01	LOCATPRC CHECKPRC	The entry retrieved from the catalog was an invalid type for alter requests, or required fields could not be located.
IDC3528I	AL0-4	IDCAL01	LOCATPRC	Passwords were suppressed when the object to be altered was retrieved from the catalog.
IDC3529I	AL0-5	IDCAL01	IDCAL01	The data set name was longer than 44 characters after the * was replaced with a level name from a generic locate.
IDC3530I	AL0-6	IDCAL01	MEMRENAM	The entryname and the NEWNAME parameters both contain membername but each membername refers to a different partitioned data set.
IDC3536I	AL0-2	IDCAL01	IDCAL01	Either the entryname or the NEWNAME specified a generic name. Both must specify a generic name if one does.
IDC3537I	AL0-12	IDCAL01	CHECKPRC	UNIQUEKEY or UPGRADE was specified for a nonalternate index.
IDC3538I	AL0-13	IDCAL01	CHECKPRC	UNIQUEKEY or UPGRADE was specified for a nonempty alternate index.
IDC3539I	AL0-14	IDCAL01	CHECKPRC	KEYS or RECORDSIZE was specified for a nonempty object.
IDC3540I	AL0-15	IDCAL01	CHECKPRC	A conflict between the control interval and KEYS or RECORDSIZE values exists.
IDC3541I	AL0-16	IDCAL01	CHECKPRC	A conflict exists between the alternate index key values and the base cluster record size.
IDC3542I	AL0-17	IDCAL01	CHECKPRC	Unequal record sizes were specified for a relative record data set.
IDC3545I	AL0-20	IDCAL01	CHECKPRC	Invalid values were specified for KEYS or RECORDSIZE.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC3546I	AL0-21	IDCAL01	CHECKPRC	Invalid value specified for KEYS.
IDC3547I	AL0-22	IDCAL01	CHECKPRC	KEYS or RECORDSIZE is invalid with entry type.
IDC3558I	DLO-14	IDCDL01	PARAMCHK	The type of entry to be deleted was retrieved from the catalog, but the type conflicts with the FORCE option.
IDC3568I	LC1-11	IDCLC01	ENTPROC	Level name starts or ends with an '*'.
IDC3570I	PR0-18	IDCRP01	IDCRP01	Delimiters were specified for a catalog reload.
IDC3572I	PR0-20	IDCRP01	CATRELOD	Target catalog is too small to contain the backup catalog during catalog reload.
IDC3573I	PR0-21	IDCRP01	CATRELOD	Either the catalog name, the volume serial number, or the device type did not match during a catalog reload.
IDC3576I	PR0-27	IDCRP01	COPYICF	An attempt to change the catalog pointer in the VVR entry for this object failed.
IDC3577I	PR0-28	IDCRP01	COPYICF	A nonempty target catalog has been found during an ICF catalog copy request.
IDC3578I	PR0-29	IDCRP01	COPYICF	The source and target catalog names are the same.
IDC3579I	PR0-30	IDCRP01	COPYICF	The logical record length (LRECL) specified when defining the target was less than that specified when the source catalog was defined.
IDC3580I	PR0-31	IDCRP01	GETNAM	A VVDS name or the source catalog name has been specified by the entries parameter.
			GETNAM	A name specified in the ENTRIES parameter did not identify a base object.
IDC3581I	PR0-16	IDCRP01	IDCRP01	IDCRP01 found beginning and/or ending delimiters when a VSAM catalog is to be copied.
IDC3582I	PR0-14	IDCRP01	IDCRP01	The organization of the input data set is incompatible with that of the output data set.
IDC3583I	PR0-17	IDCRP01	DELIMSET	Invalid delimiters were specified for a data set copy operation.
		IDCPR01	DELIMSET	Invalid delimiters were specified for a data set copy operation.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC3584I	PR0-24	IDCRP01	REOVCAT	An attempt was made to copy from or into a catalog with the recoverable attribute.
IDC3585I	PR0-25	IDCRP01	IDCRP01	Invalid data set for encipher/decipher.
IDC3586I	PR0-26	IDCRP01	IDCRP01	Target record size too small for RRDS encipher.
IDC3587I	PR0-32	IDCRP01	IDCRP01	An attempt was made to unload or reload an ICF catalog.
IDC3588I	PR0-33	IDCRP01	IDCRP01	An attempt was made to merge a non-ICF catalog.
IDC3589I	PR0-34	IDCRP01	IDCRP01	A VVDS name has been specified as the source or target object.
IDC3592I	XP0-3	IDCXP01	CLUSPROC	The object retrieved from the catalog for export is not a cluster, an alternate index, or an ICF catalog.
IDC3593I	XP0-4	IDCXP01	CLUSPROC	The catalog did not return the entry type, data component name, or LRECL when the object to be exported was located.
		IDCRC01	SYNCH	No data association could be found.
		IDCRC02	ASOCPROC	The catalog did not return the entry type when the object to be recovered was located.
			CLUSPROC	Either (1) the catalog did not return the entry type, data component name, or LRECL when the object to be recovered was located, or (2) the entry type was not a cluster or alternate index.
			CONTRBL	The catalog did not return the entry type, data component name, or LRECL when the object to be recovered was located.
			GDGPROC	The catalog did not return the entry type when the object to be recovered was located.
			NVSMPROC	The catalog did not return the entry type when the object to be recovered was located.
		IDCCC01	CATTYPE	The catalog did not return the association when a LOCATE was issued.
IDC3596I	XP0-7	IDCXP01	CLUSPROC	The data set to be exported has been marked as not usable.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC3598I	XP0-9	IDCXP01	CLUSPROC	The object is a VSAM volume data set that cannot be exported.
IDC3599I	XP0-10	IDCXP01	CLUSPROC	The ICF catalog to be exported could not be serialized.
		IDCCC01	CVOLCNVT	The target catalog could not be serialized.
			VSAMCNVT	The source or target catalog could not be serialized.
IDC3602I	MP0-9	IDCMP01	IDCMP01	Import of the data set failed after a successful define.
		IDCRM01	IDCRM01	Import of the data set failed after a successful define.
IDC3605I	MP0-29	IDCMP01	CLUSPROC	An invalid object's subparameter was specified for an ICF catalog.
IDC3606I	MP0-1	IDCMP01	CLUSPROC	The portable data set's first record was not valid, or the record immediately prior to the data component's records was not valid.
		IDCRM01	IDCRM01	Opens of the portable data set failed.
			ALISPROC	A catalog control record for an alias entry was not read.
			CLUSPROC	There was no volume list from the input area.
			NVSMPROC	A catalog control record from the portable data set was not read.
			OPENPROC	Header flags were not set properly in the first record.
			UCATPROC	A catalog control record for a user catalog was not read.
IDC3607I	MP0-13	IDCMP01	DUPNPROC	The temporary flag is not set in the catalog entry with the same name as the object being imported. If NEWNAME is specified, the temporary flag is not set in the entry with the new name.
IDC3608I	MP0-10	IDCMP01	CNCTPROC	VSAM catalog management could not connect the user catalog.
IDC3609I	MP0-5	IDCMP01	CLUSPROC	The VOLUMES parameter was not specified.
		IDCRM01	CLUSPROC	No volume information was available.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC3610I	MP0-6	IDCMP01	CNCTPROC	The device list was not specified for connect of a user catalog.
IDC3612I	MP0-8	IDCMP01	DUPNPROC	The catalog entry with the same name as the object being imported is not a cluster or alternate index.
IDC3613I	MP0-14	IDCMP01	CLUSPROC	The open of the portable data set was not successful.
		IDCRM01	IDCRM01	The open of the portable data set was not successful.
IDC3614I	MP0-7	IDCMP01	CLUSPROC	The object names specified by the user do not match the object names found in the portable data set.
IDC3615I	MP0-15	IDCMP01	RECPROC	The data set name on the OUTFILE JCL statement does not agree with the name found in the portable data set or, if NEWNAME is specified, the new name for the data set, or the name specified is not the name of path over the object to be imported.
IDC3617I	MP0-17	IDCMP01	DUPNPROC	The attributes of a predefined data set conflict with those of the data set to be imported.
IDC3619I	MP0-19	IDCRM01	ALTRPROC	The catalog return code was nonzero.
IDC3624I	MP0-24	IDCRM01	IDCRM01	The UIOINFO issued to obtain the output data set name failed.
IDC3625I	MP0-25	IDCMP01	CLUSPROC	IDCMP01 found an empty data set with the same name but the INTOEMPTY keyword was not specified.
IDC3629I	MP0-30	IDCMP01 IDCRM01	CLUSPROC CLUSPROC	The portability data set format is invalid—CI mode not supported.
IDC3633I	CC0-5,6	IDCCC01	CONVERT	A valid catalog entry was not found while reading the OS/VS catalog.
			PUSHDOWN	An index control entry was not found while reading the OS/VS catalog.
			VOLINDEX	A volume index control entry was not found while reading the OS/VS catalog.
IDC3641I	BI0-2	IDCBI01	LOCPROC	The file identified in INFILE or INDATASET is not a base cluster.
IDC3643I	BI0-4	IDCBI01	OPENPROC	The base cluster is empty.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC3883I	LR1-14	IDCLR01	ERROR	More than 50 errors occurred while trying to complete the LISTCRA.
IDC3897I	DL0—29	IDCL01	MEMDELETE	A conflict exists between the entryname and the DD-specified PDS name.
IDC3900I	RI1-12	IDCSA02	IDCSA02	The PUTGET macro failed in one of the following ways: no buffer space available, invalid parameters, TSO profile command did not allow prompting.
IDC3901I	RI1-13	IDCSA02	IDCSA02	The TSO service routine did not qualify a data set name because no qualifiers were found, invalid parameters, or failure to locate the data set name.
IDC3902I	RI1-14	IDCSA02	IDCSA02	The TSO default service routine did not qualify a data set name because no qualifiers were found, invalid parameters, or failure to locate the data set name.
IDC4227I	RI0-28	IDCRI01	GETNEXT	An ELSE command appears without a matching IF-THEN command (THENFLAG is not on with DOFLAG off).
IDC4228I	RI0-29	IDCRI01	GETNEXT	An END command appears without a matching DO command (DOFLAG is off).
IDC4229I	RI0-30	IDCRI01	MODAIIF	An IF command relational expression does not follow the required format.
IDC4230I	RI0-31	IDCRI01	MODALSET	A SET command assignment expression does not follow the required format.
IDC4232I	RI0-33	IDCRI01	MODALIF	A THEN keyword does not appear in an IF command.
IDC4236I	RI0-37	IDCRI01	IDCRI03	End-of-file occurred, but EOFOK flag is off, indicating that end-of-file occurred in the middle of a command.
IDC4237I	RI0-38	IDCRI01	MODALIF	The current IF command nesting level (NESTLVL) exceeds the maximum level allowed (IFNSTMAX).
IDC4999I		IDCSA01	PRNTERR	UABORT error message printed. See ABORT codes.
IDC01002I	RS0-3	IDCRS01	INIT	Informational message indicating the catalog to be reset and the time stamp on the volume.
IDC01011I	RS0-12	IDCRS01	PROCCRA	Informational message indicating the CRA to be reset and the time stamp on the volume.

Message	STID	Module	Procedure	Situation That Caused Message
IDC01037I	RS0-47	IDCRS01	UPDCAT	Informational message indicating that RESETCAT processing has been completed for the indicated catalog.
IDC01371I	DA0-20	IDCDA03	RECDUMP	A record with an error in an entry is not being displayed.
IDC01402I	CC0-18	IDCCC01	VSAMCNVT	The sphere entry and any of its associations were successfully converted.
IDC01407I	CC0-23	IDCCC01	VSAMCNVT	The sphere entry and any of its associations were successfully backed out of the target catalog.
IDC01408I	CC0-24	IDCCC02	DELSPACE	The data spaces are removed from the specified volume.
IDC01460I	PR0-35	IDCRP01	MERGEICF	Indicates the number of objects merged from one ICF catalog to another.
IDC01500I	BD0-1 BD0-5 BD0-6	IDCBD01	IDCBD01	Function completed successfully.
IDC01551I	LA0-19	IDCLA01	IDCLA01	An informational message indicating the status of caching activity.
IDC01600I	SC0-1 SC0-5 SC0-6 SC0-7	IDCSC01	IDCSC01	Function completed successfully.
IDC01700I	XM0-1	IDCXM02	IDCXM02	Begin INDEXTEST.
IDC01701I	XM0-2	IDCXM03	IDCXM03	Begin DATATEST.
IDC01702I	XM0-3	IDCXM02	IDCXM02	Informational message providing high-used RBA.
			VALDATA	Informational message providing high-used RBA.
			VALINDX	Informational message providing high-used RBA.
			SEQIXTST	Informational message providing high-used RBA.
			GETIXCI	Informational message providing high-used RBA.
		IDCXM03	VALARDB	Informational message providing high-used RBA.
IDC01703I	XM0-4	IDCXM02	VALDATA	Informational message providing high-allocated RBA.

Message	STID	Module	Procedure	Situation That Caused Message
			VALINDX	Informational message providing high-allocated RBA.
			GETIXCI	Informational message providing high-allocated RBA.
		IDCXM03	VALARDB	Informational message providing high-allocated RBA.
IDC01704I	XM0-5	IDCXM02	IDCXM02	Informational message providing control interval size.
			VALDATA	Informational message providing control interval size.
			VALINDX	Informational message providing control interval size.
			SSTEST	Informational message providing control interval size.
		IDCXM03	VALAMDSB	Informational message providing control interval size.
			VALARDB	Informational message providing control interval size.
IDC01705I	XM0-6	IDCXM02	VALDATA	Informational message providing control area size.
			SEQIXTST	Informational message providing control area size.
			SSTEST	Informational message providing control area size.
			BRBTST	Informational message providing control area size.
		IDCXM03	VALARDB	Informational message providing control area size.
IDC01706I	XM0-7	IDCXM02	IDCXM02	Informational message providing RBA.
			MULIXTST	Informational message providing RBA.
			GETIXCI	Informational message providing RBA.
			BRBTST	Informational message providing RBA.
		IDCXM03	INDXPROC	Informational message providing RBA.
			DATAPROC	Informational message providing RBA.
			VALAMDSB	Informational message providing RBA.
			VALARDB	Informational message providing RBA.
			CKFRCI	Informational message providing RBA.

Message	STID	Module	Procedure	Situation That Caused Message
			RDDATACI	Informational message providing RBA.
IDC01707I	XM0-8	IDCXM02	MULIXTST	Informational message providing index level.
			XTEST	Informational message providing index level.
			GETIXCI	Informational message providing index level.
IDC01708I	XM0-9	IDCXM02	IDCXM02	Informational message providing number of control intervals encountered.
		IDCXM03	IDCXM03	Informational message providing number of control intervals encountered.
IDC01709I	XM0-10	IDCXM03	IDCXM03	DATATEST complete.
IDC01710I	XM0-11	IDCXM03	IDCXM03	Informational message providing number of data records.
IDC01711I	XM0-12	IDCXM03	IDCXM03	Informational message providing number of deleted control intervals.
IDC01712I	XM0-13	IDCXM03	IDCXM03	Informational message providing maximum length data record.
IDC01713I	XM0-14	IDCXM03	DSPDATCI	Informational message providing RBA of data control interval.
IDC01714I	XM0-15	IDCXM02	DUMPCI	Informational message providing offset value.
		IDCXM03	DSPINDCI	Informational message providing offset value.
			DSPDATCI	Informational message providing offset value.
IDC01716I	XM0-17	IDCXM03	INSEQTST	Informational message providing index key.
			CRSSCHCK	Informational message providing index key.
IDC01717I	XM0-18	IDCXM03	DASEQTST	Informational message providing data key.
			CRSSCHCK	Informational message providing data key.
IDC01718I	XM0-19	IDCXM03	SPANNED	Informational message providing spanned record update number.
IDC01720I	XM0-21	IDCXM02	DUMPCI	Informational message providing RBA of index control interval.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
		IDCXM03	DSPINDCI	Informational message providing RBA of index control interval.
IDC01722I	XM0-23	IDCXM03	IDCXM03	Informational message providing percentage of free space.
IDC01723I	XM0-24	IDCXM01	IDCXM01	Concurrent access may have caused errors.
IDC01724I	XM0-25	IDCXM02	IDCXM02	INDEXTEST complete.
IDC01725I	XM0-26	IDCXM03	DSPDATCI	Informational message providing RBA of referral data control interval.
IDC01726I	XM0-27	IDCXM02	DUMPCI	Informational message providing RBA of referral index control interval.
		IDCXM03	DSPINDCI	Informational message providing RBA of referral index control interval.
IDC11003I	RS0-4	IDCRS06	RECMGMT	IGNORE was specified and an I/O error was encountered.
IDC11015I	RS0-16	IDCRS06	RECMGMT	IGNORE was specified and an I/O error was encountered.
IDC11022I	RS0-22, 48	IDCRS02	PROCTYPE	An object contains a dependency on a record that does not exist.
IDC11023I	RS0-23, 24	IDCRS02	VERA VERC VERG VERR	An entry is chained to a record of a type different than anticipated or the object noted consists of an incomplete set of records. If the control interval number of the expected association is not given, then no association for that object exists in the base record; an association for that type is required for the entry name noted.
IDC11029I	RS0-31	IDCRS03	VLNRESET VLRESET	The suballocated data space has been corrected to reflect what is on the volume. This correction occurs if entries are deleted by RESETCAT or space stated as suballocated is not suballocated (that is, the space map is incorrect on entry to RESETCAT).
IDC11031I	RS0-33	IDCRS03	CHKUNQ	The unique data or index component has less space described than the data space. Informational message to indicate that space exists which is not in use.

Message	STID	Module	Procedure	Situation That Caused Message
IDC11033I	RS0-35	IDCRS03	CHKUNQ VLNRESET	A unique data set on a volume not being reset has no corresponding data or index component.
IDC11036I	RS0-46	IDCRS03	CHKDSDIR	The data set named may have invalid space information. The extents occupied by the named data set are not in conflict with any other VSAM data set or with the system; however, a self-checking field failed to check.
IDC11040I	RS0-38	IDCRS03	VOLCHK	The VSAM format-1 DSCB did not have a corresponding space header in the volume record. Therefore, the catalog does not account for the space allocated to the data set.
IDC11041I	RS0-39	IDCRS03	VOLCHK	The extents in the space header for the data space noted were not identical to the extents in the corresponding format-1 DSCB.
IDC11042I	RS0-40	IDCRS03	VOLCHK	The space header for the data space referred to a nonexistent format-1 DSCB.
IDC11043I	RS0-41	IDCRS03	VOLCHK	The timestamp for the volume record did not match the timestamp in the VTOC.
IDC11044I	RS0-42	IDCRS03	VOLCHK	The attempt to scratch the file for the reason stated in message IDC11040I failed.
IDC11360I	DA0-1	IDCDA03	SUMNOERR	Entries with no errors are being listed.
IDC11361I	DA0-2	IDCDA03	SUMINCL	Summarizing the INCLUDEs not found.
	DA0-9	IDCDA03	SUMEXCL	Summarizing the EXCLUDEs not found.
IDC11362I	DA0-3	IDCDA03	SUMVOLNF	Summarizing the volumes not found.
IDC11367I	DA0-3	IDCDA03	SUMCATNF	Summarizing the catalogs not found.
IDC11373I	DA0-24	IDCDA03	SUMCMPR	Summarizing the compares not found.
IDC11441I	LC1-14	IDCLC02	CDIPROC	The entry received contained interruption flags indicating that an attempt was previously made to delete or update the entry, but the process did not complete successfully.
IDC11462I	PR0-37	IDCRP01	IDCRP01	Requested range end beyond end of data set.
IDC11603I	SC0-2	IDCSC01	IDCSC01	The first I/O operation received an equipment check.
IDC11700I	XM0-30	IDCXM02	SEQIXTST	Examine found data set error.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
			MULIXTST	Examine found data set error.
IDC11701I	XM0-31	IDCXM02	SEQIXTST	Examine found data set error.
			MULIXTST	Examine found data set error.
IDC11702I	XM0-32	IDCXM02	SEQIXTST	Examine found data set error.
IDC11703I	XM0-33	IDCXM02	STORIXP	Examine found data set error.
		IDCXM03	INSEQTST	Examine found data set error.
IDC11704I	XM0-34	IDCXM02	STORIXP	Examine found data set error.
		IDCXM03	INSEQTST	Examine found data set error.
IDC11705I	XM0-35	IDCXM02	DUPTEST	Examine found data set error.
IDC11706I	XM0-36	IDCXM02	SSTEST	Examine found data set error.
		IDCXM03	INDXPROC	Examine found data set error.
IDC11707I	XM0-37	IDCXM02	DUPTEST	Examine found data set error.
			CKFRCI	Examine found data set error.
IDC11708I	XM0-38	IDCXM02	IDCXM02	Examine found data set error.
IDC11709I	XM0-39	IDCXM02	VALDATA	Examine found data set error.
		IDCXM03	VALARDB	Examine found data set error.
IDC11710I	XM0-40	IDCXM02	VALDATA	Examine found data set error.
		IDCXM03	VALARDB	Examine found data set error.
IDC11711I	XM0-41	IDCXM02	IDCXM02	Examine found data set error.
IDC11712I	XM0-42	IDCXM02	VALDATA	Examine found data set error.
		IDCXM03	VALARDB	Examine found data set error.
IDC11714I	XM0-44	IDCXM02	VALINDX	Examine found data set error.
IDC11715I	XM0-45	IDCXM02	VALINDX	Examine found data set error.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
		IDCXM03	VALARDB	Examine found data set error.
IDC11716I	XM0-46	IDCXM02	VALINDX	Examine found data set error.
		IDCXM03	VALARDB	Examine found data set error.
IDC11717I	XM0-47	IDCXM02	MULIXTST	Examine found data set error.
IDC11718I	XM0-48	IDCXM02	SEQIXTST	Examine found data set error.
IDC11719I	XM0-49	IDCXM02	MULIXTST	Examine found data set error.
			XTEST	Examine found data set error.
IDC11720I	XM0-50	IDCXM02	IDCXM02	Examine found data set error.
IDC11721I	XM0-51	IDCXM02	SSTEST	Examine found data set error.
		ICDXM03	DATAPROC	Examine found data set error.
			CKFRCI	Examine found data set error.
IDC11722I	XM0-52	IDCXM02	BRBTST	Examine found data set error.
		IDCXM03	VINDXHDR	Examine found data set error.
IDC11723I	XM0-53	IDCXM02	BRBTST	Examine found data set error.
		IDCXM03	VINDXHDR	Examine found data set error.
IDC11724I	XM0-54	IDCXM02	BRBTST	Examine found data set error.
		IDCXM03	INDXPROC	Examine found data set error.
IDC11725I	XM0-55	IDCXM02	MULIXTST	Examine found data set error.
IDC11726I	XM0-56	IDCXM02	XTEST	Examine found data set error.
IDC11727I	XM0-57	IDCXM02	VALINDX	Examine found data set error.
		IDCXM03	VALARDB	Examine found data set error.
IDC11728I	XM0-58	IDCXM03	CKFRCI	Examine found data set error.
			CKDLCI	Examine found data set error.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC11730I	XM0-60	IDCXM03	INDXPROC	Examine found data set error.
			UNSPNNED	Examine found data set error.
IDC11731I	XM0-61	IDCXM03	SPANNED	Examine found data set error.
IDC11732I	XM0-62	IDCXM03	DATAPROC	Ex. nine found data set error.
IDC11733I	XM0-63	IDCXM03	DASEQTST	Examine found data set error.
IDC11734I	XM0-64	IDCXM03	CRSSCHCK	Examine found data set error.
IDC11735I	XM0-65	IDCXM03	CKDLCI	Examine found data set error.
IDC11736I	XM0-66	IDCXM02	GETIXCI	Examine found data set error.
		IDCXM03	VALRDF	Examine found data set error.
IDC11737I	XM0-67	IDCXM03	VALRDF	Examine found data set error.
IDC11738I	XM0-68	IDCXM03	VALRDF	Examine found data set error.
IDC11739I	XM0-69	IDCXM03	SPANNED	Examine found data set error.
IDC11740I	XM0-70	IDCXM03	DATAPROC	Examine found data set error.
			VALRDF	Examine found data set error.
IDC11741I	XM0-71	IDCXM03	DASEQTST	Examine found data set error.
IDC11742I	XM0-72	IDCXM03	INDXPROC	Examine found data set error.
IDC11743I	XM0-73	IDCXM02	GETIXCI	Examine found data set error.
		IDCXM03	INDXPROC	Examine found data set error.
IDC11744I	XM0-74	IDCXM03	INDXPROC	Examine found data set error.
IDC11745I	XM0-75	IDCXM02	GETIXCI	Examine found data set error.
		IDCXM03	DATAPROC	Examine found data set error.
			CKFRCI	Examine found data set error.
IDC11751I	XM0-81	IDCXM03	VALRDF	Examine found data set error.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC11755I	XM0-85	IDCXM03	DATAPROC	Examine found data set error.
IDC11756I	XM0-86	IDCXM03	CKFRCI	Examine found data set error.
IDC11757I	XM0-87	IDCXM03	DATAPROC	Examine found data set error.
IDC11758I	XM0-88	IDCXM02	GETIXCI	Examine found data set error.
IDC11760I	XM0-90	IDCXM02	IDCXM02	Examine found data set error.
IDC11761I	XM0-91	IDCXM03	VALAMDSB	Examine found data set error.
			VALRDF	Examine found data set error.
IDC11762I	XM0-92	IDCXM02	SSTEST	Examine found data set error.
			STORIXP	Examine found data set error.
			FCICNT	Examine found data set error.
IDC11763I	XM0-93	IDCXM02	GETIXCI	Examine found data set error.
		IDCXM03	VALARDB	Examine found data set error.
IDC11764I	XM0-94	IDCXM02	IDCXM02	Examine found data set error.
		IDCXM03	VALAMDSB	Examine found data set error.
IDC11765I	XM0-95	IDCXM02	VALDATA	Examine found data set error.
		IDCXM03	VALARDB	Examine found data set error.
IDC11766I	XM0-96	IDCXM02	STORIXP	Examine found data set error.
			FCICNT	Examine found data set error.
		IDCXM03	INDXPROC	Examine found data set error.
			VINDXHDR	Examine found data set error.
IDC11767I	XM0-97	IDCXM02	VALDATA	Examine found data set error.
		IDCXM03	VALARDB	Examine found data set error.
IDC11768I	XM0-98	IDCXM02	GETIXCI	Examine found data set error.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
		IDCXM03	INDXPROC	Examine found data set error.
			RDDATACI	Examine found data set error.
IDC11769I	XM0-99	IDCXM02	SEQIXTST	Examine found data set error.
			SSTEST	Examine found data set error.
			STORIXP	Examine found data set error.
			STORHHP	Examine found data set error.
IDC11770I	XM0-100	IDCXM02	XTEST	Examine found data set error.
		IDCXM03	VINDXHDR	Examine found data set error.
IDC11771I	XM0-101	IDCXM02	GETIXCI	Examine found data set error.
		IDCXM03	DATAPROC	Examine found data set error.
			VALAMDSB	Examine found data set error.
			CKFRCI	Examine found data set error.
IDC11772I	XM0-102	IDCXM02	XTEST	Examine found data set error.
		IDCXM03	VINDXHDR	Examine found data set error.
IDC21009I	RS0-10	IDCRS01 IDCRS03	INIT MARKUNUS	A multivolume file existed on a volume prior to reset.
IDC21020I	RS0-21	IDCRS06	CRAMNT	A volume needed for the reset was not specified in a CRAFILES parameter.
IDC21024I	RS0-25	IDCRS02	VERX	The alias chain for a USERCATALOG or NONVSAM entry is invalid.
IDC21025I	RS0-26	IDCRS03	VERB	The records associating the GDG data set with the GDG base are in error.
IDC21026I	RS0-27	IDCRS02	SETCI	A previous message indicated an error which resulted in this entry being deleted from the catalog.
IDC21027I	RS0-28	IDCRS03	VLRESET	The CRA extents or catalog extents.
	RS0-29	IDCRS03	VLRESET	have no matching extents in the data space.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC21030I	RS0-32	IDCRS03	MARKUNUS	The entry noted claims space on a volume. That space is not allocated to that entry.
IDC21032I	RS0-34	IDCRS02 IDCRS03	VERCI VERB	An object of the type specified was defined over the entry named as *entryname*. However, the records describing the object could not be found. Therefore, an object of the type specified was deleted from the given entryname's description. No name for the deleted object is given because the record with its name cannot be found.
			VLRESET	What space is available for suballocation on a volume, is not the correct length in the catalog.
IDC21045I	RS0-43	IDCRS07	RENAMEP	An attempt was made to reset an object which has the same name as some other object in the catalog.
IDC21046I	RS0-44	IDCRS07	RENAMEP	An attempt was made to reset a unique object into a catalog which contains an object of the same name.
IDC21047I	RS0-45	IDCRS07	RENAMEP	An attempt was made to reset a unique object into a catalog which contains an object of the same name.
IDC21363I	DA0-4	IDCDA03	SUMERROR	Listing the entries with errors.
IDC21364I	DA0-5 DA0-19	IDCDA02	LOGERROR	An error has been found in an entry.
IDC21365I	DA0-6 DA0-18 DA0-21	IDCDA03	RECDUMP	A record with an entry in error is being displayed.
IDC21372I	DA0-22	IDCDA01	INITCMPR	A compare data set should be, but is not a VVDS.
IDC21404I	CC0-20	IDCCC02	ASSOCTNS	The base sphere has an invalid association type.
IDC21409I	CC0-25	IDCCC01	VSAMCNVT	Conversion failed for the specified sphere.
IDC21410I	CC0-26	IDCCC02	VSAMCNVT	Deletion of the catalog entry for the specified sphere failed.
IDC21411I	CC0-27	IDCCC02	DEFCLUS	The specified data set is marked as not usable.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC21412I	CC0-28	IDCCC02	DEFNVSM CNVTGDG DEFCLUS CNVTUCAT CNVTALIS CNVTPATH	The define of the specified object into the target catalog failed.
IDC21700I	XM0-110	IDCXM02	IDCXM02	INDEXTEST found minor errors.
IDC21701I	XM0-111	IDCXM02	IDCXM02	INDEXTEST found major errors.
IDC21702I	XM0-112	IDCXM03	IDCXM03	DATATEST found minor errors.
IDC21703I	XM0-113	IDCXM03	IDCXM03	DATATEST found major errors.
IDC31000I	RS0-1	IDCRS02	INIT	The catalog specified for reset is not a recoverable catalog.
IDC31001I	RS0-2	IDCRS01	INIT	The master catalog was specified for reset.
IDC31004I	RS0-5	IDCRS06	WFDEF	DEFINE failed for the workfile.
IDC31005I	RS0-6	IDCRS02	INIT	The workfile was defined in the catalog to be reset.
IDC31006I	RS0-7	IDCRS07	CATEOV	A physical I/O error was encountered when accessing the catalog.
IDC31007I	RS0-8	IDCRS07	CATEOV	A logical I/O error was encountered while extending the catalog.
IDC31008I	RS0-9	IDCRS01	INIT	An error was encountered when trying to access the data set specified in the CATALOG parameter.
IDC31010I	RS0-11	IDCRS01	MERGECRA	The CRA was specified for reset, but it belongs to a catalog other than the catalog to be reset.
IDC31012I	RS0-13	IDCRS06	RECMGMT	The workfile relative record number limit has been exceeded.
IDC31013I	RS0-14	DCRS01	MERGECRA	A preceding message indicates that either open failed for the CRA, close failed for the CRA, or the CRA does not belong to the catalog to be reset.
IDC31014I	RS0-15	IDCRS06	WFDEL	DELETE failed for the work file.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC31016I	RS0-17	IDCRS01	INIT	The CRAFILES parameter specified no CRA with the ALL option; therefore, no volume was specified for reset.
IDC31017I	RS0-18	IDCRS01	INIT	Some other task is open to the catalog requested to be reset.
IDC31018I	RS0-19	IDCRS01	UPDCAT	RESETCAT required a volume that could not be allocated.
IDC31019I	RS0-20	IDCRS01	INIT	The CRAFILES parameter specified the same volume serial number via *dnames.*
IDC31035I	RS0-37	IDCRS01 IDCRS03	UPDCAT VLNRESET	In a CRA, either the volume record for the *volser* indicated does not exist or one of its secondary records does not exist.
IDC31038I	RS0-49	IDCRS01	UPDCRA	Either open or close failed for the CRA.
	DA0-23	IDCDA01	INITCMPR	A compare data set should be, but is not, an ICFCATALOG.
IDC31366I	DA0-7	IDCDA01	INITICF	The input data set is not an ICFCATALOG as specified.
	DA0-10	IDCDA01	NFOVINFL NFOVINDS	The input data set is not a VVDS as specified.
IDC31368I	DA0-11	IDCDA01	INIT	The "CATALOG" qualifier has been given with INCLUDE or EXCLUDE, but is not allowed for an ICFCATALOG DIAGNOSE.
IDC31369I	DA0-14	IDCDA01	ICFPROC VVDSPROC FORMATCK	The value given or defaulted with "ERRORLIMIT" has been reached. Reading of the input data set stops. DIAGNOSE proceeds to summarize processing to this point.
IDC31370I	DA0-13	IDCDA01	INITICF	SVC OBTAIN against the named ICF CATALOG has failed.
			INITCMPR	SVC OBTAIN against the named COMPARE data set has failed.
			BIDENQ	An attempt to retrieve information on the named DD statement failed.
			NFOCINDS	After dynamically allocating the ICF CATALOG, an attempt to retrieve information on the DD statement failed.
IDC31376I	DA0-28	IDCDA02	INITDLST	An attempt to position to the first VVDS entry in the input BCS has failed.

Messages-to-Module Cross-Reference

Message	STID	Module	Procedure	Situation That Caused Message
IDC31377I	DA0-29	IDCDA02	INITDLST	An attempt to position to the first entry of the input BCS has failed.
IDC31400I	CC0-16	IDCCC01	CATCHECK	A UIOINFO macro call failed to obtain the catalog name.
IDC31401I	CC0-17	IDCCC01	VSAMCNVT	A LISTC get next type LOCATE failed to get the next catalog entry.
IDC31403I	CC0-19	IDCCC01	CATTYPE	A LOCATE to the catalog to determine its type failed.
IDC31405I	CC0-21	IDCCC01	CATCHECK	The source and target catalog types are an invalid combination.
IDC31406I	CC0-22	IDCCC01	CATCHECK	The source catalog is a master catalog.
IDC31440I	LC1-13	IDCLC01	GNXTPROC ENTPROC	Space parameter is invalid for an extended function catalog.
IDC31442I	LC1-15	IDCLC01	INITPROC	Space and user catalog are conflicting parameters.
IDC31461I	PR0-36	IDCRP01	MERGEICF	The source catalog specification identifies the system master catalog.
IDC31463I	PR0-38	IDCRP01	IDCRP01	Required PDS member name missing.
IDC31502I	BD0-3	IDCBD01	IDCBD01	The CCHH is not in bounds.
IDC31503I	BD0-4	IDCBD01	IDCBD01	The DEVTYP macro was issued to obtain CCHH boundaries, but the attempt failed.
IDC31504I	BD0-9	IDCBD01	IDCBD01	The amount of data to be bound a specified by the LOWCCHH and HIGHCCHH parameters was more than the available cache space.
IDC31550I	LA0-1	IDCLA01	IDCLA01	3380/3350 subsystem devices online could not be located.
IDC31601I	SC0-3	IDCLA01 IDCBD01 IDCSC01	IDCLA01 IDCBD01 IDCSC01	The specified volume cannot be found on a subsystem.
IDC31602I	SC0-4	IDCLA01 IDCBD01 IDCSC01	IDCLA01 IDCBD01 IDCSC01	A load real address instruction failed.

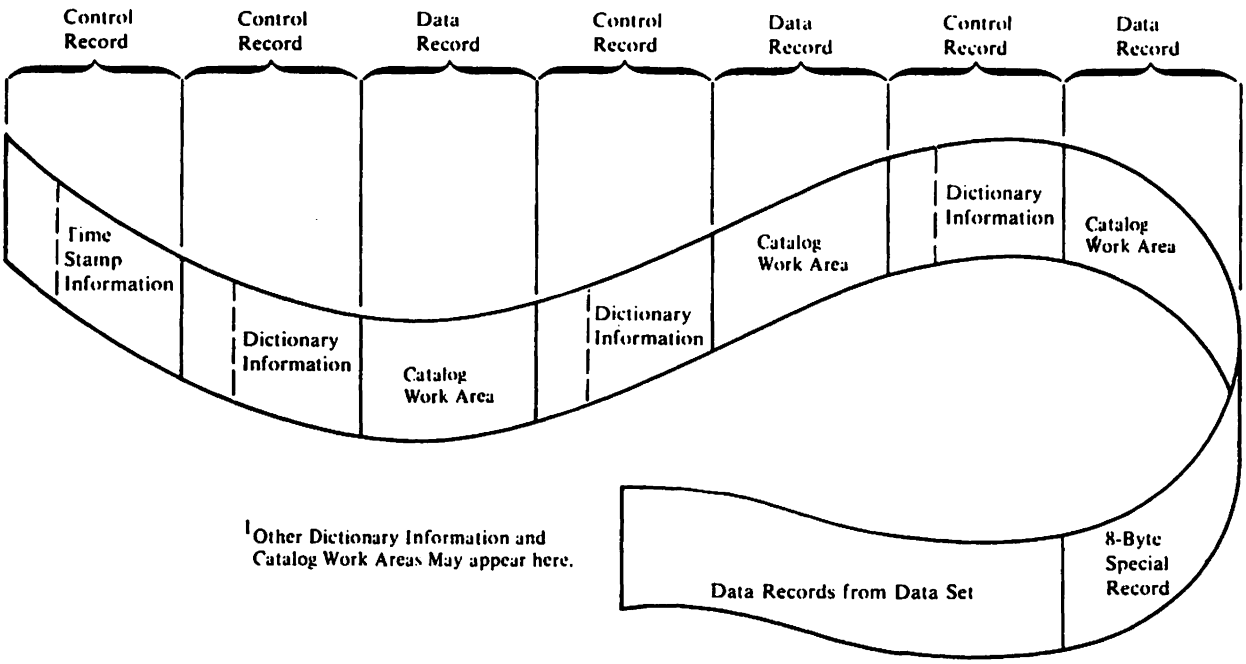
Message	STID	Module	Procedure	Situation That Caused Message
IDC31604I	SC0-10	IDCLA01 IDCBD01 IDCSC01	IDCLA01 IDCBD01 IDCSC01	The specified unit type is invalid.
IDC31700I	XM0-120	IDCXM01	IDCXM01	VSAM open error.
IDC31701I	XM0-121	IDCXM02	IDCXM02	Insufficient virtual storage.
			SEQIXTST	Insufficient virtual storage.
			MULIXTST	Insufficient virtual storage.
		IDCXM03	GPOOL	Insufficient virtual storage.
IDC31702I	XM0-122	IDCXM01	IDCXM01	Data set not a KSDS.
IDC31703I	XM0-123	IDCXM01	IDCXM01	Data set in create mode.
IDC31704I	XM0-124	IDCXM01	IDCXM01	Data set open for output or not correctly closed.
IDC31705I	XM0-125	IDCXM01	IDCXM01	DATATEST not performed due to INDEXTEST test errors.
IDC31706I	XM0-126	IDCXM02	VALDATA	VSAM control block error.
			VALINDX	VSAM control block error.
		IDCXM03	VALARDB	VSAM control block error.
IDC31707I	XM0-127	IDCXM01	IDCXM01	Unable to get qualified name.
IDC31708I	XM0-128	IDCXM01	IDCXM01	Obtain format one DSCB error.
IDC31709I	XM0-129	IDCXM01	IDCXM01	Unable to get volume information.
IDC31710I	XM0-130	IDCXM02	GETIXCI	Control interval access error.
		IDCXM03	INDXPROC	Control interval access error.
			RDDATACI	Control interval access error.

Appendix A. Portable Data Sets Created by the EXPORT Command

When a VSAM cluster, alternate index, or ICF catalog is exported via the access method services EXPORT command, catalog information needed to define the VSAM data set plus all the records from the data component are written to a non-VSAM data set or catalog called the portable data set. The following list shows the attributes of the portable data set.

Attribute	**Value**
LRECL	The larger of: 1. Maximum VSAM data set record size +4. 2. For portable data sets created on systems without a selectable unit, Supervisor Performance #2, VS2.03.807: 264 (for key sequenced and entry sequenced data sets) or 268 (for relative record data sets). For portable data sets created on systems with Data Facility Extended Function or selectable unit, Supervisor Performance #2, VS2.03.807: 272 (for non-RRDSs) or 276 (for RRDSs).
BLKSIZE	As specified by the user. The default is 2048.
RECFM	VBS
DSORG	PS
DEVTYPE	Tape or disk

The portable data set contains two *major* types of records: control records and data records. Control records contain one of two types of information: a time stamp or a dictionary. Data records also contain one of two types of information: a catalog work area or a data record from the data component of the cluster or alternate index exported. Figure 32 on page 454 shows the general layout of control records and data records in the portable data set. The types of records and the types of information within those records are explained in this appendix.



[1] Other Dictionary Information and Catalog Work Areas May appear here.

Figure 32. Layout of Control Records and Data Records in the Portable Data Set

EXPORT Control Records

EXPORT control records all have the same general format as shown in Figure 33. The first 4 bytes of each control record contain header information. The next 4 bytes contain associated data. The remainder of the record contains the time stamp or dictionary information.

0	4	8
Header	Associated Data	Variable Data—Time Stamp or Dictionary

Figure 33. General Format of Control Records Created by EXPORT Command

Control Record Containing Time Stamp Information

The first record on every portable data set is a control record that contains time stamp information, as well as other fields. The format of this record is shown in Figure 34.

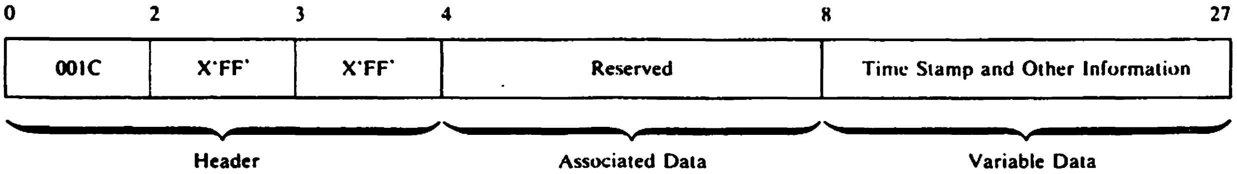


Figure 34. Control Record Containing Time Stamp Information

The first 2 bytes of the header contain the length of this control record. The next 2 bytes indicate that this control record contains time stamp information. There is no associated data, and those 4 bytes are reserved.

The format of the time stamp information is:

Displacement[1] **Description**

8 (8) Number of cluster or alternate index components and paths being exported.

9 (9) Flags:

Bit	**Meaning When Set**
0	1 indicates a unique data set. 0 indicates a nonunique data set.
1	1 indicates an inhibited target. 0 indicates a noninhibited target.
2	1 indicates path associations are present. 0 indicates no paths are present.
3	If bit 2 is 1: 1 indicates that the base object has both data and index components. 0 indicates that the base object has only a data component.

10(A) Access method services release number in EBCDIC.

11(B) Flags:

Bit	**Meaning When Set**
0	1 indicates CIMODE (DOS only). 0 indicates RECORDMODE.
1	1 indicates NOALLOCATE file (DOS only).

[1] The displacement is from the beginning of the control record.

2 1 indicates SAM ESDS file (DOS only).
3 Reserved.
4 1 indicates empty file exported (DOS only).
5 1 indicates ICF catalog exported (does not apply to VSE or OS/VS1).
6 Reserved.
7 Reserved.

12(C) Time of EXPORT in EBCDIC, in the form hh.mm.ss, where hh is the number of hours, mm the number of minutes, and ss the number of seconds.

20(14) Date of EXPORT in EBCDIC, in the form mm/dd/yy, where mm is the month in digits, dd the day, and yy the year.

Control Records Containing Dictionary Information

A control record containing dictionary information is written for the cluster or alternate index being exported and for each component within that cluster or alternate index. In addition, one control record is written for each path association of the object being exported. These records in essence describe the data record containing the catalog work area which follows. The format of control records containing dictionary information is shown in Figure 35.

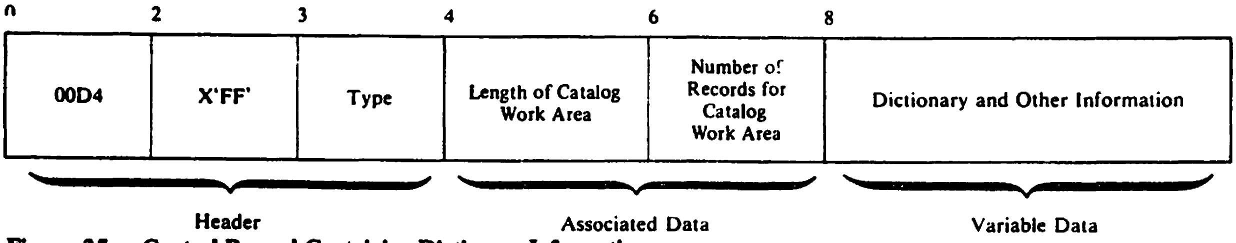

0
2
3
4
6
8
00D4
X'FF'
Type
Length of Catalog Work Area
Number of Records for Catalog Work Area
Dictionary and Other Information
Header
Associated Data
Variable Data

Figure 35. Control Record Containing Dictionary Information

The first 2 bytes of the header contain the length of this control record. The next 2 bytes indicate that this record contains dictionary information and the type of component that the associated catalog work area information describes. The type of component is indicated by 'C' for cluster, 'D' for data, 'I' for index, 'G' for alternate index, or 'R' for path.

The associated data portion of the control record contains the length of the associated catalog work area (2 bytes) and the number of records into which the associated catalog work area is broken (2 bytes).

The variable data portion of the control record contains the dictionary information. This portion of the control record begins with a 4-byte field that contains the number of entries in the dictionary. The entries themselves follow. Each entry consists of a pair of 4-byte fields. The first 4 bytes contain the length of the associated catalog field in the catalog work area. (Remember, the catalog work area information is in a data record immediately following one of these control records.) The second 4 bytes contain the displacement of that field within the associated data record. If an associated catalog field contains no information, both

4-byte fields in the dictionary entry contain zeros. The dictionary entries always point to the associated fields in the order shown in the following list.

Order of Associated Catalog Fields

Order	**Associated Field in Catalog Work area**	**Description**
1	ENTYPE	Component type.
2	ENTNAME	Component name.
3	DSATTR	Data set attributes.
4	OWNERID	Data set owner.
5	DSETCRDT	Data set creation date.
6	DSETEXDT	Data set expiration date.
7	BUFSIZE	Minimum buffer size.
8	LRECL	Logical record size.
9	SPACEPARM	Primary and secondary space.
10	PASSWORD	Four 8-character passwords.
11	PASSPRMT	Password prompting code name.
12	PASSATMP	Maximum number of attempts for password.
13	USVRMDUL	User security verification module.
14	USERAREC	User authorization record.
15	LOKEYV	Low key on volume.
16	HIKEYV	High key on volume.
17	VOLSER	Volume serial numbers.
18	AMDSBCAT	AMDSB, from which the next nine fields are taken.
19	AMDATTR	Attributes.
20	AMDRKP	Relative key position.
21	AMDKEYLN	Key length.

Order of Associated Catalog Fields

Order	Associated Field in Catalog Work area	Description
22	AMDCINV	Control interval size.
23	AMDLRECL	Maximum record size.
24	AMDPCTCA	Percent of free control intervals in control area.
25	AMDPCTCI	Percent of free bytes in control intervals.
26	AMDATTR3	Attributes.
27	AMDAXRKP	Position of alternate key in base cluster record.
28	EXCPEXIT	Exception exit.
29	RGATTR	Alternate index or path attributes.
30	RELATE | PATHENTRY	Alternate index related name or pathentry name.
31	PASSREL	Master password of pathentry component.
32	SECFLAGS	RACF security field.

Data Records

Data records contain one of two types of information: the catalog work area or data records from the data component.

Data Records Containing Catalog Work Area

Each control record that contains dictionary information is followed by a data record that contains the catalog work area for a given component. The format of these records is shown in Figure 36.

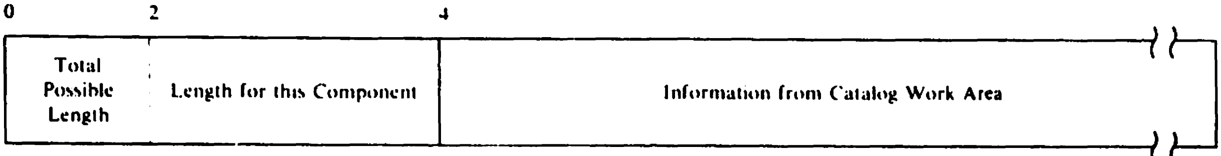


Figure 36. Data Record Containing Catalog Work Area

The first 2 bytes of each record contain the total possible length of the catalog work area. The next 2 bytes contain the length of the work area used for this component. Following these first 4 bytes are the fields from the catalog work area. The order of these fields is basically as described in the preceding topic. If there is no information for one of the fields, the field is completely omitted.

Figure 37 shows the relationship of the dictionary and catalog work area information.

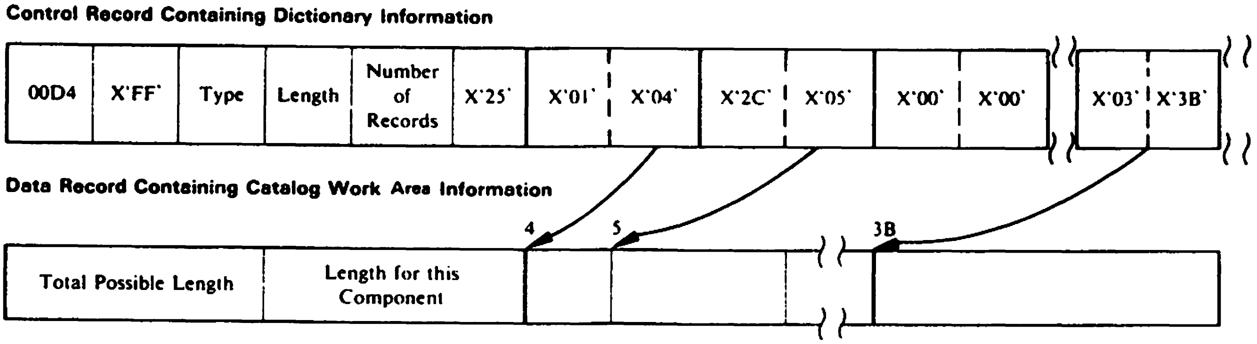

Control Record Containing Dictionary Information
00D4
X'FF'
Type
Length
Number of Records
X'25'
X'01'
X'04'
X'2C'
X'05'
X'00'
X'00'
X'03'
X'3B'
Data Record Containing Catalog Work Area Information
4
5
3B
Total Possible Length
Length for this Component

Figure 37. Relationship of Dictionary and Catalog Work Area Information

Data Records Containing Data Records from the Data Component

All of the control records and data records that contain dictionary information are followed by a special record which marks the beginning of the data records from the data component. This special record is 8 bytes in length. The record always has the format shown in Figure 38.

0	2	3
X'0008'	01	Reserved

Figure 38. Special Record at Beginning of Data Records from the Data Component

This special record is followed by all of the data records from the data component being exported.

Appendix B. Portable Data Sets Created by the EXPORTRA Command

When the EXPORTRA command of access method services executes, it produces a portable data set which contains catalog information obtained from a catalog recovery area (CRA) and data records for VSAM clusters and alternate indexes, and also catalog information for user catalog pointers. In addition, portable data sets created by EXPORTRA (referred to as recovery portable data sets in this appendix) on OS/VS systems may contain catalog information for non-VSAM, alias, and generation data group (GDG) base objects. The following list shows the attributes of the portable data set.

Attribute **Value**

LRECL The larger of:

1. Maximum VSAM data set record size +8.

2. For portable data sets created on systems without selectable unit, Supervisor Performance #2, VS2.03.807: 268 (if the portable data set does not contain any VSAM relative record data sets) or 272 (if the portable data set contains any VSAM relative record data sets).

   For portable data sets created on systems with Data Facility Extended Function or selectable units, Supervisor Performance #2, VS2.03.807: 276 (if the portable data set does not contain any VSAM relative record data sets) or 280 (if the portable data set contains any VSAM relative record data sets).

BLKSIZE As specified by the user (the default is 2048).

RECFM VBS

DSORG PS

DEVTYPE (Tape or disk)

Each record of the recovery portable data set has a special 4-byte header added that precedes the record itself. Information for unrelated objects on the recovery portable data set is separated by one or more software ends of file. These ends of file are special records that consist only of the 4-byte header. Only Figure 39 on page 462 indicates that this particular type of header precedes each data record; the other figures do not show it.

The recovery portable data set contains two *major* types of records: control records and data records. Control records contain one of two types of information: a time stamp or a dictionary. Data records also contain one of two types of information: a catalog work area or a data record from the data component of the cluster exported. Figure 39 shows the general layout of control records and data records in the recovery portable data set. The types of records and the types of information within those records are explained in this appendix.

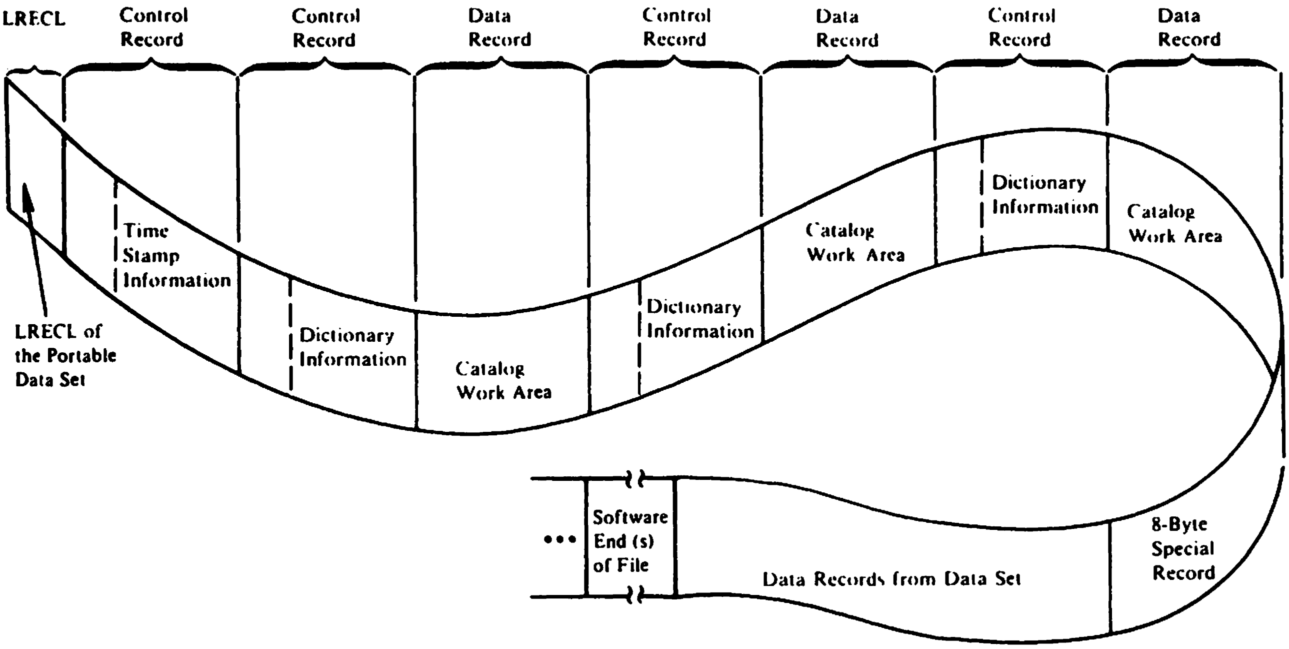


Figure 39. Layout of Control Records and Data Records in the Recovery Portable Data Set

EXPORTRA Control Records

EXPORTRA control records all have the same general format as shown in Figure 40. The first 4 bytes of each control record contain header information. The next 4 bytes contain associated data. The remainder of the record contains the time stamp, dictionary information, or logical record length.

0	4	8
Header	Associated Data or Logical Record Length	Variable Data–Time Stamp or Dictionary

Figure 40. General Format of Control Records Created by EXPORTRA Command

Control Record Containing the Logical Record Length

The first record of every recovery portable data set is a control record containing the logical record length of the portable data set itself. The format of this record is shown in Figure 41.

0	2	3	4	8
0008	X'FF'		LRECL	

Figure 41. Control Record Containing the Logical Record Length

Control Record Containing Time Stamp Information

The first record for each item on the recovery portable data set is a control record that contains time stamp information, as well as other fields. The format of this record is shown in Figure 42.

0	2	3	4	8	27
001C	X'FF'	X'FF'	Reserved	Time Stamp and Other Information	
Header			Associated Data	Variable Data	

Figure 42. Control Record Containing Time Stamp Information

The first 2 bytes of the header contain the length of this control record. The next 2 bytes indicate that this control record contains time stamp information. There is no associated data, and those 4 bytes are reserved.

The format of the time stamp information is:

Displacement[1]	Description
8(8)	The maximum number of components associated with this item.
9(9)	Flags:

Bit	Meaning When Set
0	1 indicates a unique data set. 0 indicates a nonunique data set.
1	1 indicates an inhibited target. 0 indicates a noninhibited target.

[1] The displacement is from the beginning of the control record.

2 1 indicates path associations are present.
0 indicates no paths are present.

3 If bit 2 is 1:
1 indicates that the base object has both data and index components.
0 indicates that the base object has only a data component.

4 1 always 1 for a recovery portable data set.

5 1 indicates a non-VSAM object.
0 indicates an object other than a non-VSAM.

6 1 indicates a GDG base object.
0 indicates an object other than a GDG base.

7 1 indicates a user catalog pointer.
0 indicates a nonuser catalog pointer.

10(A) Access method services release number in EBCDIC.

11(B) Reserved.

12(C) Time of export in EBCDIC, in the form hh.mm.ss, where hh is the number of hours, mm the number of minutes, and ss the number of seconds.

20(14) Date of export in EBCDIC, in the form mm/dd/yy, where mm is the month in digits, dd the day, and yy the year.

Control Records Containing Dictionary Information

A control record containing dictionary information is written for each object being exported and for each component associated with that object. These records in essence describe the data record containing the catalog work area which follows. The general format of control records containing dictionary information is shown in Figure 43.

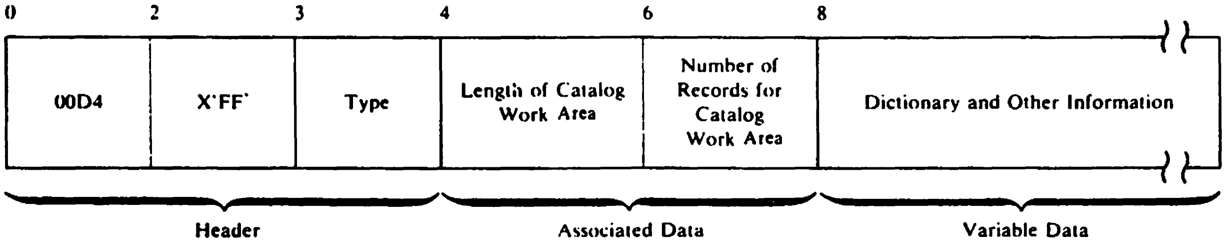

0
2
3
4
6
8
00D4
X'FF'
Type
Length of Catalog Work Area
Number of Records for Catalog Work Area
Dictionary and Other Information
Header
Associated Data
Variable Data

Figure 43. Control Record Containing Dictionary Information

The first 2 bytes of the header contain the length of this control record. The next 2 bytes indicate that this record contains dictionary information and the type of component that the associated catalog work area information describes. The type of component is indicated by 'C' for cluster, 'D' for data, 'I' for index, 'G' for alternate index, 'R' for path, 'A' for non-VSAM, 'B' for GDG base, 'X' for alias, or 'U' for user catalog pointer.

The associated data portion of the control record contains the length of the associated catalog work area (2 bytes) and the number of records into which the associated catalog work area is broken (2 bytes).

The variable data portion of the control record contains the dictionary information. This portion of the control record begins with a 4-byte field that contains the number of entries in the dictionary. The entries themselves follow. Each entry consists of a pair of 4-byte fields. The first 4 bytes contain the length of the associated catalog field in the catalog work area. (Remember, the catalog work area information is in a data record immediately following one of these control records.) The second 4 bytes contain the displacement of that field within the associated data record. If an associated catalog field contains no information, both 4-byte fields in the dictionary entry contain zeros.

The number of dictionary entries and their order depends upon the type of object being described. Dictionary formats are described for each possible kind of item in the following list.

Order of Associated Catalog Fields

Order VSAM Components	**Associated Field in Catalog Work Area**	**Description**
1	ENTYPE	Component type.
2	ENTNAME	Component name.
3	DSATTR	Data set attributes.
4	OWNERID	Data set owner.
5	DSETCRDT	Data set creation date.
6	DSETEXDT	Data set expiration date.
7	BUFSIZE	Minimum buffer size.
8	LRECL	Logical record size.
9	SPACEPARM	Primary and secondary space.
10	PASSWORD	Four 8-character passwords.
11	PASSPRMT	Password prompting code name.
12	PASSATMP	Maximum number of attempts for password.

Order of Associated Catalog Fields

Order VSAM Components	Associated Field in Catalog Work Area	Description
13	USVRMDUL	User security verification module.
14	USERAREC	User authorization record.
15	LOKEYV	Low key on volume.
16	HIKEYV	High key on volume.
17	VOLSER	Volume serial numbers.
18	AMDSBCAT	AMDSB from which the next nine fields are taken.
19	AMDATTR	Attributes.
20	AMDRKP	Relative key position.
21	AMDKEYLN	Key length.
22	AMDCINV	Control interval size.
23	AMDLRECL	Maximum record size.
24	AMDPCTCA	Percent of free control intervals in control area.
25	AMDPCTCI	Percent of free bytes in control intervals.
26	AMDATTR3	Attributes.
27	AMDAXRKP	Position of alternate index key in base cluster record.
28	EXCPEXIT	Exception exit.
29	RGATTR	Alternate index or path attributes.
30	RELATE | PATHENTRY	Alternate index related name or path entry name.

Order of Associated Catalog Fields

Order VSAM Components	Associated Field in Catalog Work Area	Description
31	PASSREL	Master password of path entry component.
32	SECFLAGS	RACF security fields.
Non-VSAM		
1	ENTYPE	Entry type.
2	ENTNAME	Entry name.
3	VOLSER	Volume serial numbers.
4	DEVTYP	Device types.
5	FILESEQ	File sequence numbers.
6	OWNERID	Data set owner.
7	DSETCRDT	Data set creation date.
8	DSETEXDT	Data set expiration date.
User Catalog Pointers		
1	ENTYPE	Entry type.
2	ENTNAME	Entry name.
3	VOLSER	Volume serial numbers.
4	DEVTYP	Device types.
Aliases		
1	ENTYPE	Entry type.

Order of Associated Catalog Fields

Order VSAM Components	Associated Field in Catalog Work Area	Description
2	ENTNAME	Entry name.
GDG Bases		
1	ENTYPE	Entry type.
2	ENTNAME	Entry name.
3	GDGLIMIT	GDG limit value.
4	GDGATTR	GDG attributes.
5	OWNERID	Data set owner.
6	DSETCRDT	Data set creation date.
7	DSETEXDT	Data set expiration date.

Data Records

Data records contain one of two types of information: the catalog work area or data records from the data component of a VSAM cluster.

Data Records Containing Catalog Work Area

Each control record that contains dictionary information is followed by a data record that contains the catalog work area for a given component. The format of these records is shown in Figure 44.

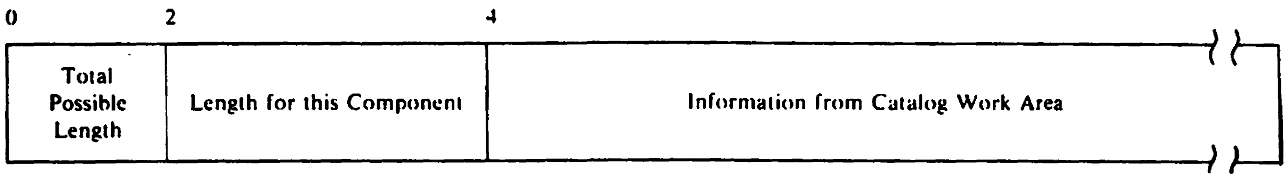


Figure 44. Data Record Containing Catalog Work Area

The first two 2 of each record contain the total possible length of the catalog work area. The next 2 bytes contain the length of the work area used for this component. Following these first 4 bytes are the fields from the catalog work area. The order of these fields is basically as described in the preceding topic. If there is no information for one of the fields, the field is completely omitted.

Figure 45 on page 469 shows the relationship of the dictionary and catalog work area information.

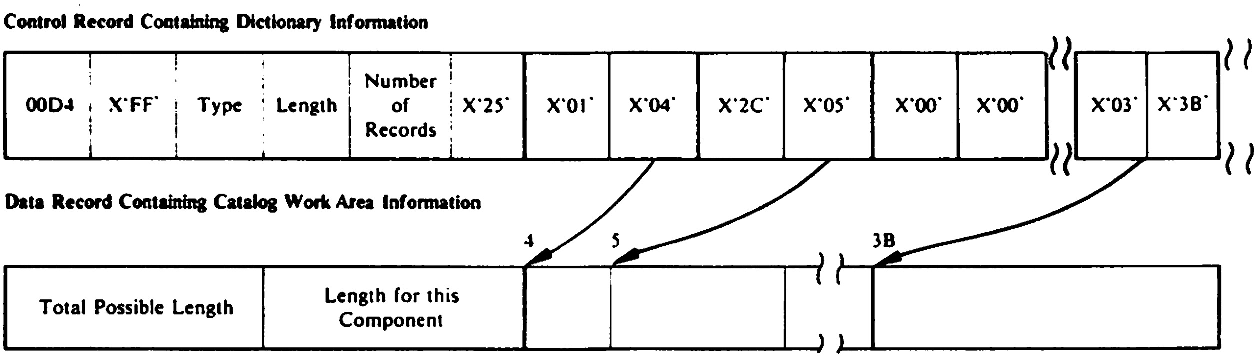

Control Record Containing Dictionary Information
00D4
X'FF'
Type
Length
Number of Records
X'25'
X'01'
X'04'
X'2C'
X'05'
X'00'
X'00'
X'03'
X'3B'
Data Record Containing Catalog Work Area Information
4
5
3B
Total Possible Length
Length for this Component

Figure 45. Relationship of Dictionary and Catalog Work Area Information

Data Records Containing Data Records from the Data Component

For a VSAM cluster or alternate index, all control records and data records that contain dictionary information are followed by a special record which marks the beginning of the data records from the data component. This special record is 8 bytes in length. The record always has the format shown in Figure 46.

0	2	3
X'0008'	01	Reserved

Figure 46. Special Record at Beginning of Data Records from the Data Component

This special record is followed by all of the data records from the data component being exported.

Associated Objects for User Catalog Pointers, Non-VSAMs, and GDGs

The aliases of a user catalog pointer or a non-VSAM are exported as associated objects. Similarly, the non-VSAMs that belong to a GDG base are exported as associated objects of the GDG; these non-VSAMs may, in turn, have aliases. An item and its associated objects are preceded by one time stamp control record and followed by one software end of file.

Appendix C. Enciphered Data Sets Created by the REPRO Command

When a data set is enciphered through the access method services REPRO command, information needed to decipher the enciphered data set is written as a header at the beginning of the enciphered data set. This header will require one or more records, depending on the record size of the enciphered data set. REPRO will write the required number of header records to the target data set when enciphering, and will read the required number of header records from the source data set when deciphering.

The format of the header is:

	Bytes and		Description:
Offset	**Bit Pattern**	**Field Name**	**Content, Meaning, Use**
0(0)	1	RPEHRCDN	Number of header records.
1(1)	7	RPEHID	Header ID = 'RPEHDID'.
8(8)	2	RPEHLEN	Length of header.
10(A)	8	RPEHTIME	Time of enciphering in EBCDIC, in the form hh.mm.ss.
18(12)	8	RPEHDATE	Date of enciphering in EBCDIC, in the form mm/dd/yy.
26(1A)	32	RPEHUSRD	User data.
58(3A)	8	RPEHDK	Enciphered data key.
66(42)	8	RPEHFK	File keyname.
74(4A)	8	RPEHENRN	Enciphered random number.
82(52)	8	RPEHSD	Seed.
90(5A)	8	RPEHENRS	Enciphered RN X OR seed.
98(62)	1	RPEHCBN	Cipher block number.

Offset	Bytes and Bit Pattern	Field Name	Description: Content, Meaning, Use
99(63)	1	RPEHFLG1	Flags.
	1... ...	RPHESTKN	Header contains file keyname.
	.1..	RPEHSTEK	Header contains data key enciphered under file key.
	..1.	RPEHRRDS	Enciphered RRDS.
	...x xxxx	*	Reserved.
100(64)	16	*	Reserved.

Index

LY26-3930-0

Reader's Comment Form

MVS/XA Access Method Services Logic Volume 2

This manual is part of a library that serves as a reference source for system analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Do not use this form to request IBM publications. If you do, your order will be delayed because publications are not stocked at the address printed on the reverse side. Instead, you should direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

If you have applied any technical newsletters (TNLs) to this book, please list them here:

Last TNL ______________________

Previous TNL ______________________

Fold on two lines, tape, and mail. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.) Thank you for your cooperation.

LY26-3930-0

Reader's Comment Form

Fold and tape

Please do not staple

Fold and tape

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

IBM Corporation
P.O. Box 50020
Programming Publishing
San Jose, California 95150

Fold and tape

Please do not staple

Fold and tape

IBM®

MVS/370 Access Method Services Logic Volume 2 (File No. S370-30) Printed in U.S.A. LY26-3930-0

MVS/370
Access Method Services Logic
Volume 2

Order No. LY26-3930-0
File No. S370-30

LY26-3930-00

Printed in U.S.A.