

VSE/Advanced Functions



Diagnosis Reference Maintain System History Program

VSE/Advanced Functions



Diagnosis Reference Maintain System History Program

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page xi.

First Edition (May 1989)

This edition applies to Version 4, Release 1 of VSE/Advanced Functions, Program Number 5686-007, and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the addresses given below.

A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

You may also send your comments by FAX or via the Internet:

Internet: s390id@de.ibm.com
FAX (Germany): 07031-16-3456
FAX (other countries): (+49)+7031-16-3456

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1985, 1989. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	xi
Trademarks and Service Marks	xi
Preface	xiii
Chapter 1: Introduction	1
The Purpose of MSHP	1
The Parts of MSHP	1
Environmental Characteristics	1
Processing	3
System and Auxiliary History Files	3
Error Handling	3
Diagnostic Tools	4
Chapter 2: Design Information	5
Program Control Flow	5
Interface Control	8
History File Records	11
Summary Information	13
Module Description	17
Convention of Presentation	17
IKRAAPAR	18
IKRACMP1	19
IKRACOMP	20
IKRAGEN	21
IKRAPPLY	22
IKRAPRD	23
IKRAPTF	24
IKRARCH	25
IKRBARDT	26
IKRBGRDT	27
IKRBINS	28
IKRBIO	29
IKRBKPR	30
IKRBKT	31
IKRBKUP	32
IKRBPRDT	33
IKRCGT	34
IKRCHECK	36
IKRCKASS	38
IKRCKCMP	40
IKRCKCTL	41
IKRCKD	42
IKRCKFIX	43
IKRCKLV2	44
IKRCKPTF	45
IKRCLMSH	46
IKRCLOSE	47
IKRCMDS	48
IKRCNVRT	49

IKRCNVTB	51
IKRCOMR	52
IKRCOPY	53
IKRCPIN	54
IKRCREAT	55
IKRCUTHF	56
IKRCVTLU	57
IKRDATE	58
IKRDEBUG	59
IKRDEL	60
IKRDEQ	61
IKRDIAGN	63
IKRDLELE	64
IKRDLIN	65
IKREDMP	66
IKRDQFR	67
IKREAPL	68
IKREARC	69
IKREBCK	70
IKRECOR	71
IKRECPY	72
IKRECRE	73
IKREDMP	74
IKREINC	75
IKRELKP	76
IKRELST	77
IKREMBDY	78
IKREMRG	80
IKREMU	81
IKREPER	83
IKRERDC	84
IKRERMV	85
IKRERST	86
IKRERTC	87
IKRERVK	88
IKRESEL	89
IKRESTL	90
IKRETLR	91
IKREUND	92
IKREXIT	93
IKRFIX	94
IKRFRBUF	95
IKRGOGO	96
IKRGTAPR	99
IKRGTBUF	100
IKRGTCMP	101
IKRGTFR	102
IKRGTGEN	103
IKRGTGRP	104
IKRGTHD	105
IKRGTIMT	106
IKRGTINF	107
IKRGTMSH	108
IKRGTPTF	109

IKRGTREC	110
IKRGTTAP	111
IKRGWA	112
IKRHDBKT	113
IKRHDLV2	114
IKRHSLA	116
IKRIMSMT	117
IKRINMSH	118
IKRINSTL	119
IKRISLNK	121
IKRISVBK	122
IKRLAPA	124
IKRLCMPR	125
IKRLGNRC	126
IKRLIBST	127
IKRLINCM	134
IKRLKUP	135
IKRLMLF	136
IKRLMOD	137
IKRLOAD	138
IKRLOFI	140
IKRLSUP	142
IKRMAIN	143
IKRMERGE	144
IKRMOD	145
IKRMRGIN	146
IKRNOTE	147
IKROPBUF	148
IKROPEN	149
IKROPHF	150
IKROPLNK	151
IKROPMSH	153
IKRPAPAL	154
IKRPARSE	155
IKRPCREQ	156
IKRPDZAP	157
IKRPERS	158
IKRPGENL	159
IKRPHMSH	160
IKRPIO	161
IKRPPTFL	162
IKRPRAPL	163
IKRPRCS	164
IKRPSDMD	165
IKRPTLNK	166
IKRPTMSH	167
IKRQUEUE	168
IKRRACSV	169
IKRRAESV	170
IKRRALNK	171
IKRRASRV	172
IKRRCV	173
IKRRDCE	174
IKRRDIPT	178

IKRRDLV2	179
IKRRDNXT	180
IKRREGCK	181
IKRREQFR	182
IKRREWR	183
IKRRLZAP	184
IKRRL2	185
IKRRL3	186
IKRRL4	188
IKRRL5	189
IKRRL6	190
IKRRL7	191
IKRRMGEN	192
IKRRTC	193
IKRRTRVE	194
IKRRWBUF	195
IKRSAASM	196
IKRSALIB	197
IKRSAMNT	198
IKRSAPL	199
IKRSEND	200
IKRSERR	201
IKRSET	202
IKRSHAPR	207
IKRSHCMP	208
IKRSHGEN	209
IKRSHGRP	210
IKRSHPRD	211
IKRSHPTF	212
IKRSHREL	213
IKRSLA	214
IKRSLZAP	215
IKRSPCLS	217
IKRSREPQ	218
IKRSRTMD	219
IKRSTMTS	221
IKRSVCMD	222
IKRSWTHF	223
IKRTAIL	224
IKRTRLCP	226
IKRUIRP	227
IKRUNDO	228
IKRUOMT	229
IKRUPGR	230
IKRWKF	233
IKRWOF	234
IKRWOR	235
IKRWRSQ	236
IKRWTO	237
IKRWTP	238
Chapter 3: Organization Information	239
Module-Module Interrelations	240
Module-Macro Interrelations	255

Module-IKRCOMRG Interrelations	267
Chapter 4: Data Area Information	273
Layout of Data Areas	278
Common MSHP Tables and Layouts	279
Standardized Internal Table (SIT)	287
SIF Tables	288
Internal Representation	348
History File Records	377
Chapter 5: Diagnostic Aids	407
Module-Message Cross-Reference Lists	407
Module-Message Interrelations	407
History File Dump Analysis Example	417
Description of History File Records	434
Index	439

Figures

1.	MSHP Storage Layout	2
2.	MSHP Function Flow	7
3.	Interface Control Flow LOAD	8
4.	Interface Control Flow (OPEN, CLOSE)	9
5.	Interface Control Flow (PUT)	10
6.	Interface Control Flow (GET)	11
7.	Structure of the History File	12
8.	History File Queue Structure	13
9.	MSHP-to-Called Program Relationship	14
10.	Semantic Checker: Function-to-Module Relationship	15
11.	Processor Routines: Function-to-Module Relationship	16
12.	Information Retrieval Guide	239
13.	Example of MSHP Control Block Relationship for the APPLY Function	274
14.	Index to Data Areas	275
15.	History File Summary	378
16.	History File Sample Dump	418
17.	Sample Dump Queue Structure	432

Notices

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 product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, New York 10594, U.S.A.

Any pointers in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement. IBM accepts no responsibility for the content or use of non-IBM Web sites specifically mentioned in this publication or accessed through an IBM Web site that is mentioned in this publication.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in certain countries:

AT
IBM
RETAIN
SP
System/370

Preface

This manual, although a 0-edition, is a major revision of the previously available publication LY33-9113-0.

This manual is intended for use by IBM personnel responsible for servicing the VSE/Advanced Functions Maintain System History Program (MSHP). The manual supplements the MSHP program listings; it is organized into the following chapters:

CHAPTER 1: INTRODUCTION outlines the purpose and method of operation of MSHP, and discusses the system history files.

CHAPTER 2: DESIGN INFORMATION explains the MSHP functions, the function flow, and the interaction between MSHP and VSE system programs. It shows the layout of storage, and the structure of the history file. Summary information tells you which of the system programs may be called by MSHP, and gives function-to-module relationships.

Detail information (such as function description, entry point, input/output, and use of registers) is given in the section "Module Description".

CHAPTER 3: ORGANIZATION INFORMATION provides cross-reference information. It helps you, for example, to find out which modules are called by a given module, or which communication region fields are used by a certain module. An overview of the information is given in Figure 12 on page 239.

CHAPTER 4: DATA AREA INFORMATION contains the functional description and layout of all MSHP data areas and of the MSHP history file record types.

CHAPTER 5: DIAGNOSTIC AIDS lists relationships between modules and messages, and between modules and commands; it explains a history file dump.

Information about using MSHP is contained in other VSE/Advanced Functions manuals as follows:

VSE/Advanced Functions Planning and Installation, SC33-6351

It tells how to install VSE/Advanced Functions and how to apply PTFs or APARs to the system. It provides MSHP-resource requirements information, which is needed for planning the system.

VSE/System Package Diagnosis, SC33-6182

It explains how to display and retrieve history information.

VSE/Advanced Functions Maintain System History Program Reference, SC33-6357

It describes the MSHP commands.

Listed below are the diagnosis reference publications that document diagnosis information for related components of VSE/Advanced Functions:

VSE Central Functions Diagnosis Reference Supervisor, SC33-6323

VSE Central Functions Diagnosis Reference Serviceability Aids, SC33-6327

VSE Central Functions Diagnosis Reference Librarian, SC33-6330

VSE/ESA System Utilities, SC33-6617

VSE Central Functions Diagnosis Reference Linkage Editor, SC33-6328

Titles and abstracts of other related publications are listed in the *IBM System/370, 30xx, 4300, and 9370 Processors Bibliography*, GC20-0001.

Chapter 1: Introduction

The Purpose of MSHP

The Maintain System History Program (MSHP), a part of VSE/Advanced Functions, is used to:

- Install programming packages by means of the MSHP functions 'INSTALL SYSRES' and 'INSTALL PRODUCT'.
A package may contain "components" or "products" and each may be divided into a production and and a generation part.
- Install APAR/local fixes by means of the MSHP function 'CORRECT'.
- Install service tapes by means of the MSHP function 'INSTALL SERVICE'. This applies program temporary fixes (PTFs) to the operational VSE system for the purpose of preventive and corrective program service.
- Produce programming packages and service tapes.

For information on using MSHP, refer to the manuals listed in the "Preface".

The Parts of MSHP

MSHP consists of the modules identified by IKRxxxxx, the phases identified by MSHPxxxx, and the phase PTFBKUP. Note that the processing module IJPARSER is not part of MSHP; the IJPARSER routine is referred to in this manual as "parser".

Environmental Characteristics

MSHP can run in any partition. The processing modules of MSHP are loaded into the low address range of the partition. The high address range of the partition is reserved for system programs called by MSHP. The space between the upper limit of the MSHP load modules and the system program's address space is used for MSHP buffers and work areas.

The storage layout is shown in Figure 1 on page 2. The storage requirements for the "MSHP load module area" (fixed size) is 400K. The size of the "buffer" and "called system program" areas (variable size) depends on the partition size: For the buffer area, MSHP requires a minimum of 16K, for the called system program area, MSHP requires a minimum of 160K. The remaining partition storage is divided between those two areas in the following proportion: three parts for the buffer pool and one part for the called system program area. If the buffer pool reaches its maximum (512K), all remaining storage is given to the called system program area.

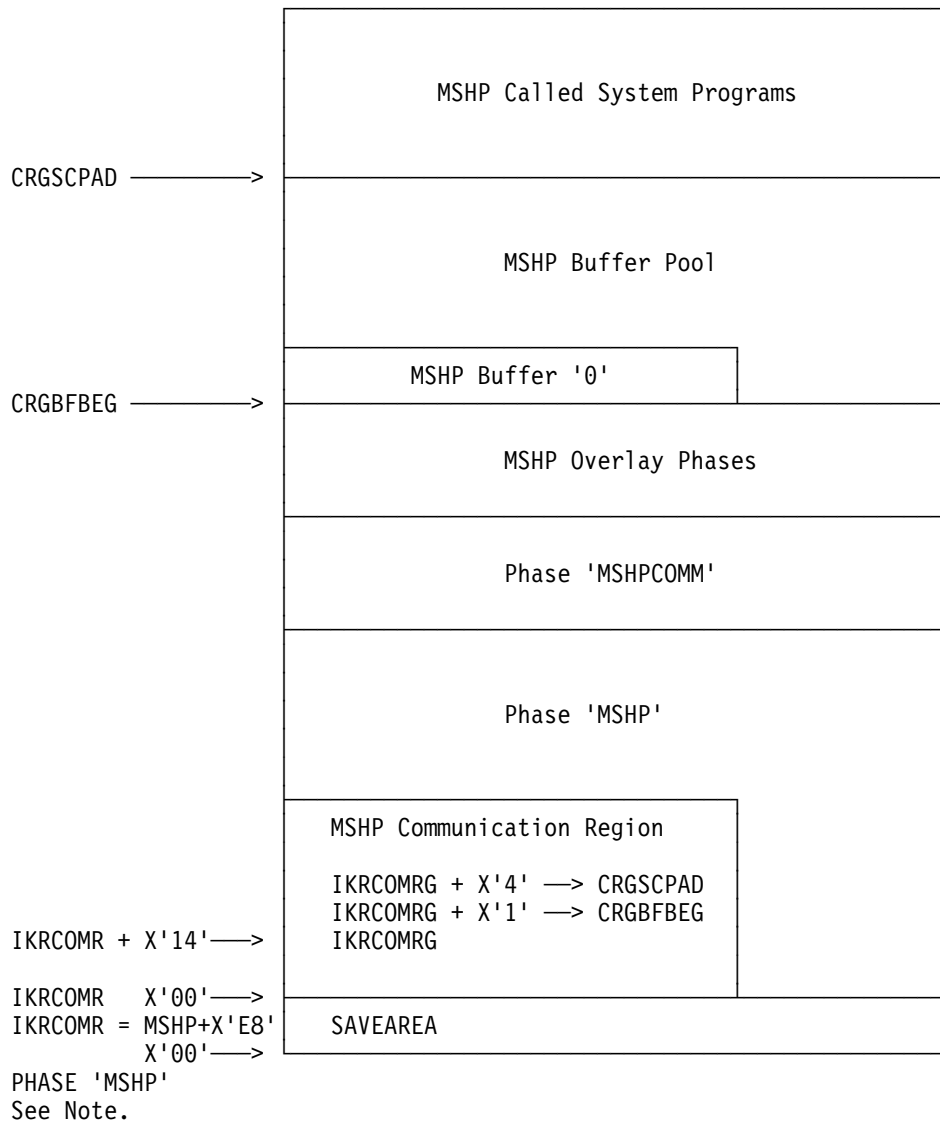


Figure 1. MSHP Storage Layout

Note: The phase begins at the start address of the partition which calls the MSHP.

Processing

Processing of MSHP is controlled by user-supplied MSHP function and detail control statements. MSHP reads the control statements either from SYSIPT or from SYSLOG and communicates with the user via SYSLST and SYSLOG.

System and Auxiliary History Files

MSHP maintains a record of the entire system on the system history file. This file, IJSYSHF, reflects the components and products installed, and the maintenance activities that have occurred.

IJSYSHF can reside on any disk. The file should reside on the same logical device as SYSREC.

For the structure of the system history file, refer to Chapter 2; for a summary of the record types, refer to Chapter 4.

In addition, MSHP uses an auxiliary history file. This file, IJSYS02, is normally used as a work file for installation and certain maintenance activities (such as for installing a service tape).

IJSYS02 can reside on any disk. The file should reside on the system logical unit SYS002. If an auxiliary history file is carried from one system to another, its extents should be defined separately.

Both types of history files must be single extents.

Error Handling

The MSHP routines provide return codes upon execution of functions. These return codes are analyzed for error detection.

In case of an error, a message indicating the failing function is issued. The main types of error messages are:

Messages indicating that MSHP cannot execute the requested function, or indicating that an MSHP internal error exists.

Message code = Mxxxx

Messages indicating that the parser detected an error during syntactical analysis of MSHP statement.

Message code = 4Dxxx

Diagnostic Tools

MSHP does not have built-in diagnostic tools. The normal VSE diagnostic tools may be used to service this program. The same rules for debugging and repair are valid. See also Chapter 5: Diagnostic Aids.

Chapter 2: Design Information

This chapter tells about the:

- Program control flow within MSHP
- Structure of the system history file
- Relationship between functions, programs, and modules
- MSHP modules (IKRxxxx).
Refer to the section "Module Description" which gives logic information for the modules; the information is in alphabetical sequence of the module names.

Program Control Flow

MSHP consists of processing modules and of service modules serving all the processing modules for:

- I/O services (reading, communication, printing workfile/PTF file services).
- History file access services.

The processing modules are:

IKRMAIN (the main routine)
IJPARSER (referred to as "parser" in this manual),

and, for each MSHP function statement

a semantic checker, and one or more processor modules.

Figure 2 on page 7 shows the functional relationship of the main components of MSHP, including the relationship to IJPARSER; the purpose of the components is as follows:

IKRMAIN: Initializes the job, reads the first function statement and gets it analyzed. This is done through the parser interface module (IKRPARSE) and the parser routine (IJPARSER). Refer to A and B in Figure 2 on page 7 IKRMAIN then calls the specific semantic checker to get the function processed. On return from the semantic checker, IKRMAIN either causes the next function to be processed or goes to EOJ.

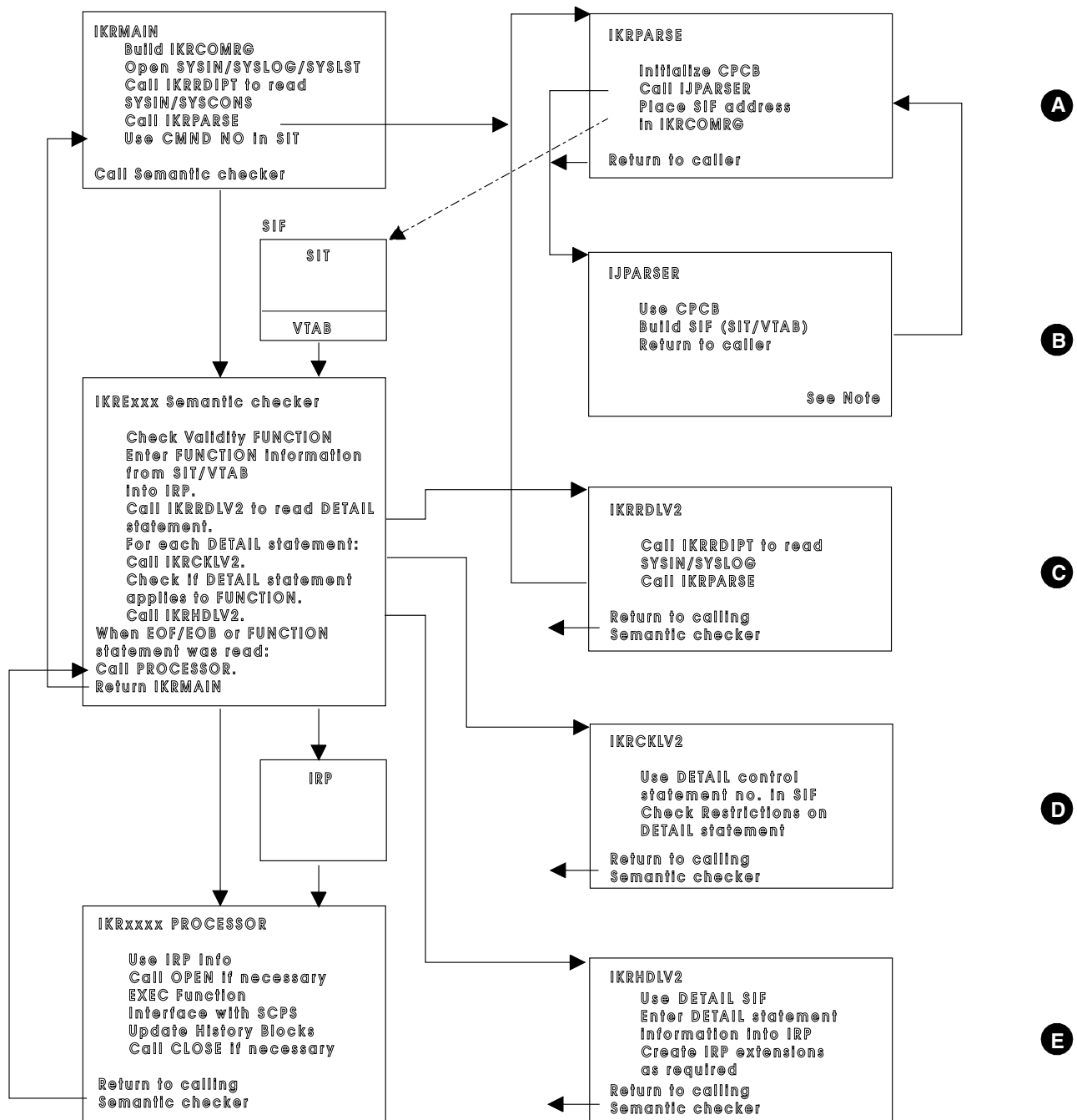
IJPARSER (which is not part of MSHP): Checks the statement submitted, extracts information from it, and builds a SIF (standardized internal format) table which consists of an SIT (standardized internal table) and a VTAB (value table). For more information on the parser; refer to *VSE/Advanced Functions Diagnosis Reference: Service Aids*.

Semantic Checker. The semantic checker reads the next statement and, if it is a detail statement, has the statement analyzed, builds an IRP (internal representation table) for the processor, and reads the next statement. The semantic checker continues to read input statements until it encounters EOF or a new function statement, at which time the checker passes the now complete IRP to the proper processor routine. Read level 2 (IKRRDLV2), check level 2 (IKRCKLV2), and handle level 2 (IKRHDLV2) are the main modules called by the semantic checker. See C, D, and

E in Figure 2 on page 7. Upon return from the processor, the semantic checker returns to IKRMAIN.

Processor Routines: Processor routines execute the function described in the IRP. The IRP indicates which function control statement is to be processed and contains the corresponding detail statement information. It loads and interfaces with the proper system program, if necessary. When the processor has executed its function, it returns to the calling semantic checker.

The function-to-module relationship is summarized in Figure 11 on page 16.



Note: For detailed information about MSHP modules refer to Section **MODULE CHARTS** later in this chapter.

For detail information about parser modules refer to **VSE/Advanced Functions Diagnosis Reference Serviceability Aids**.

Figure 2. MSHP Function Flow

Interface Control

MSHP interacts with VSE system programs via interface modules in MSHP. The modules perform functions such as the loading of system routines, OPEN and CLOSE services, and data transfer for PUT and GET functions. Registers must be saved and reloaded when control transfers between system programs and MSHP modules. MSHP modifies its partition's COMREG to contain a pointer to the interface file control table (IFCT). Refer to Chapter 4, Data Area Information, for detailed information regarding the IFCT.

MSHP requests system program modules to be loaded. Figure 3 shows the program flow between MSHP and system programs during a LOAD. Figure 4 on page 9 shows the OPEN, and a CLOSE operation. Figure 5 on page 10 describes the program control flow during a PUT operation initiated by the system program. Figure 6 on page 11 shows the program interaction during a GET operation initiated by the system program.

The numbers in the figures are provided to show the sequence of steps executed for the specified operation.

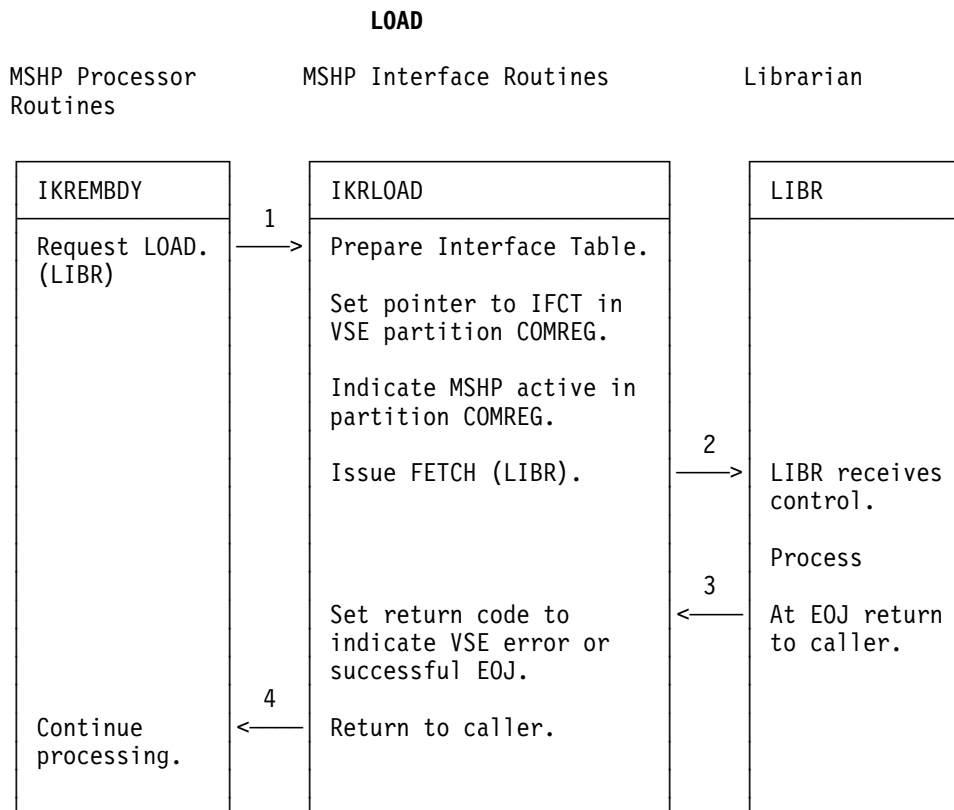


Figure 3. Interface Control Flow LOAD

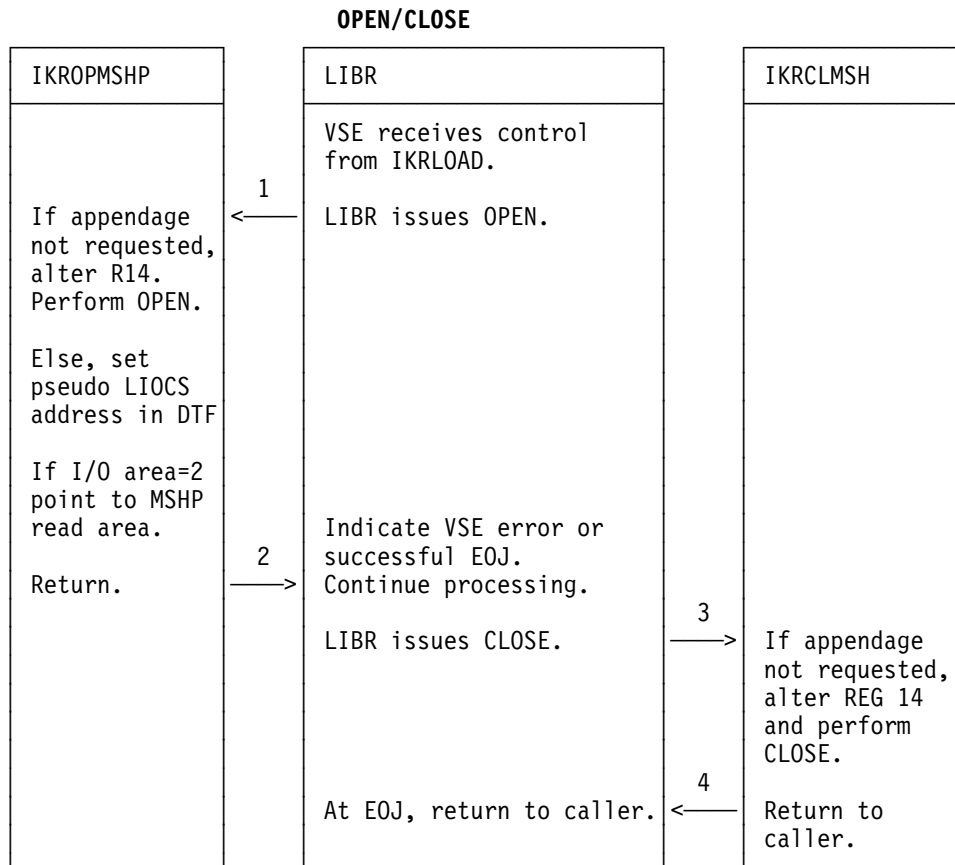


Figure 4. Interface Control Flow (OPEN, CLOSE)

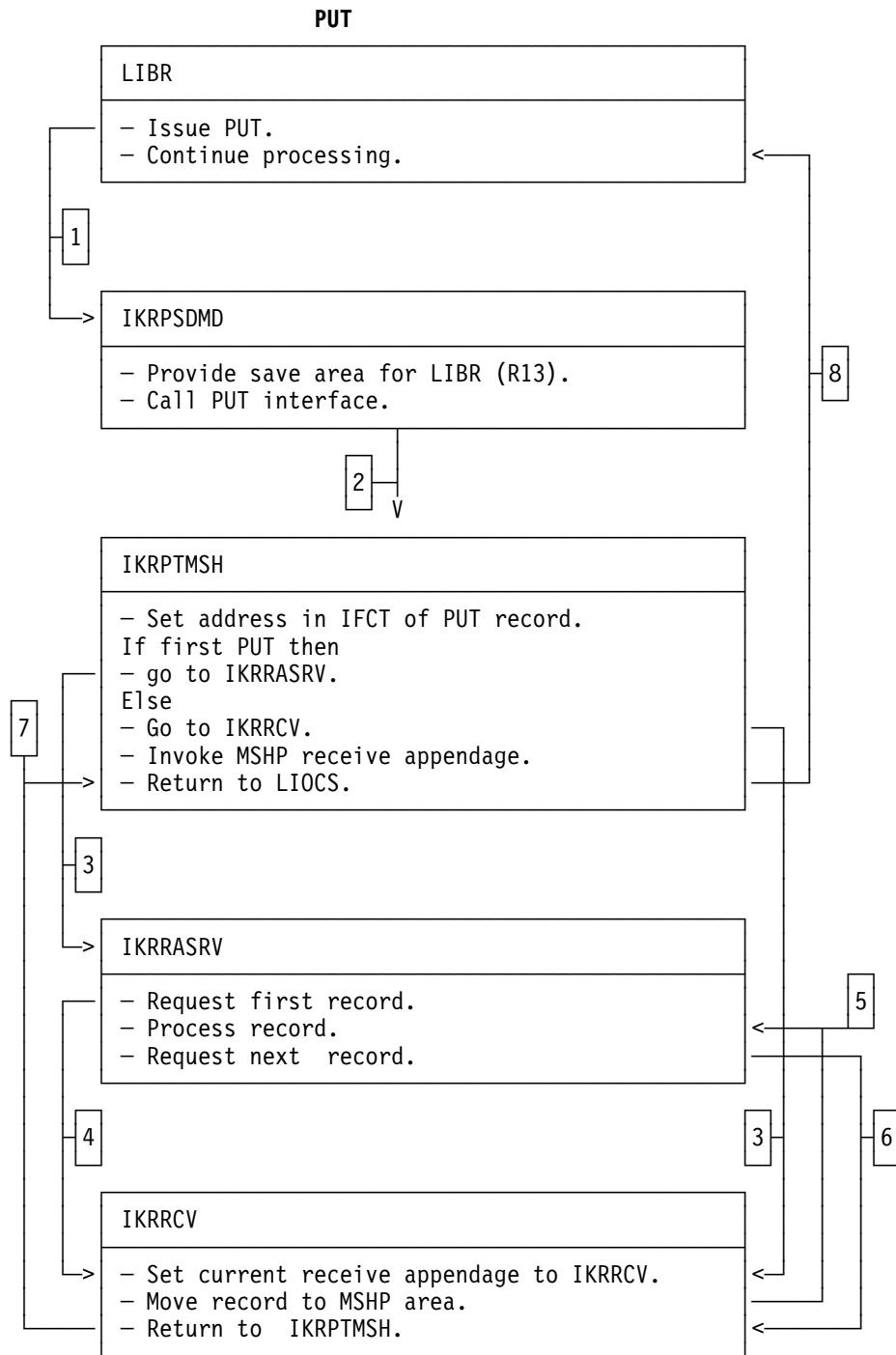


Figure 5. Interface Control Flow (PUT)

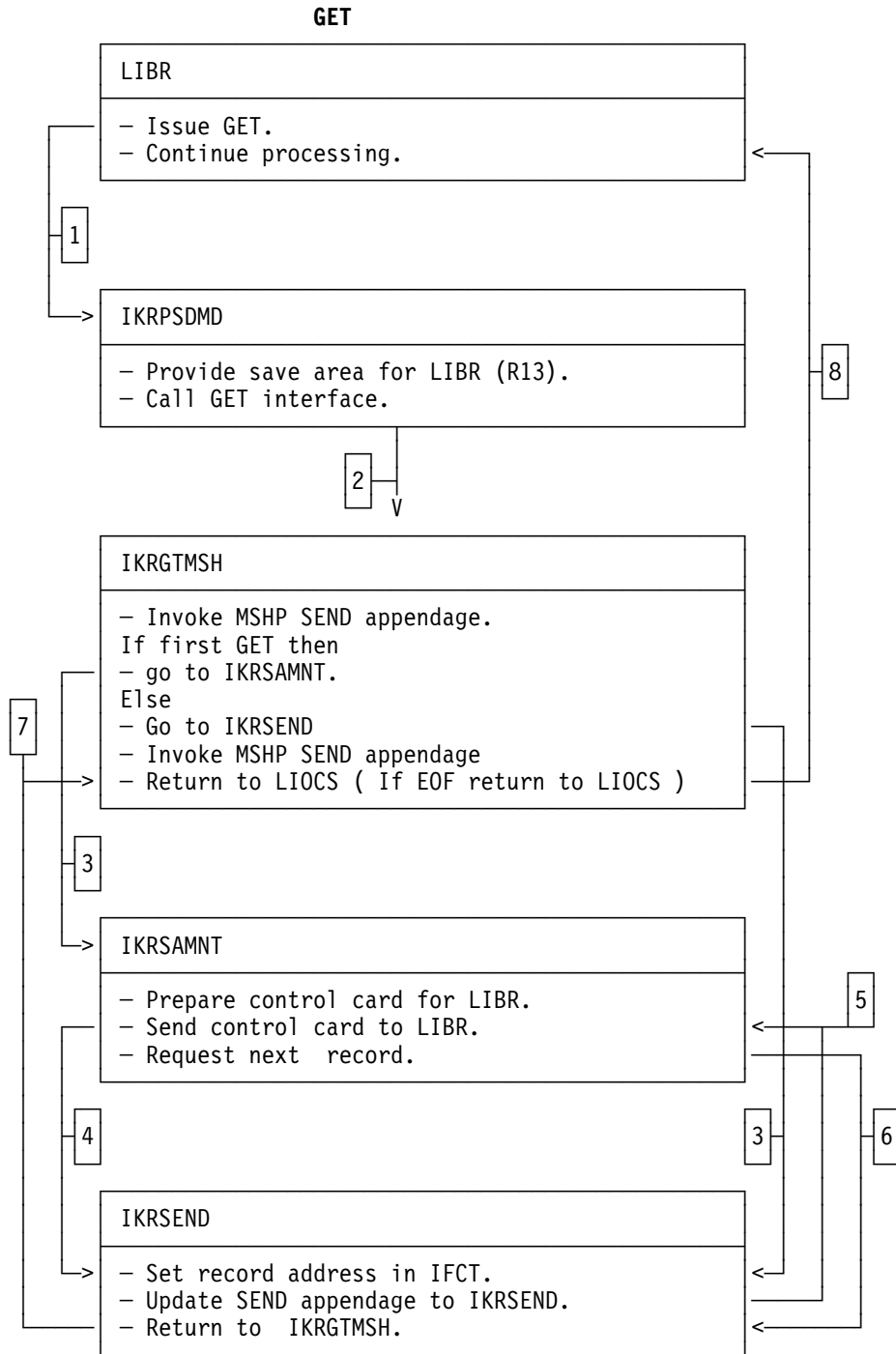


Figure 6. Interface Control Flow (GET)

History File Records

History files are organized sequentially. The MSHP history file is composed of several different record types. The records are maintained in the IJSYSHF extent. Each history file record is identified by an identifier of one or two bytes at the beginning of the record. For a summary on record types and identifier, refer to Figure 15 on page 378 in Chapter 4.

The record format is fixed unblocked (FIXUNB) with a block size of 1984 bytes. Each of the physical records contains 31 logical records of a length of 64 bytes. Figure 7 on page 12 shows the physical and logical structure of the history file records.

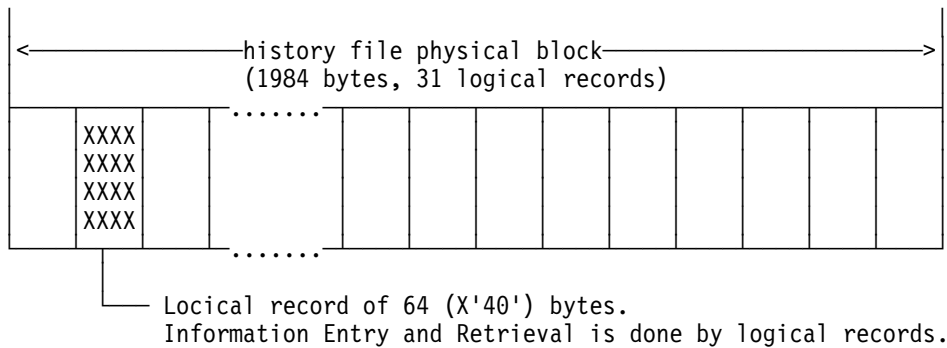


Figure 7. Structure of the History File

Note : Logical record locations within a physical block are calculated by MSHP functions internally.

A history file is organized as a pointer-connected list; the file is managed internally with 3-byte addresses called relative byte addresses (RBA). History file records are chained by forward and backward pointers to make up a certain queue structure, which is shown in Figure 8 on page 13. These pointers contain relative byte addresses (RBAs). An RBA is the hexadecimal byte displacement of the record relative to the beginning of the history file. Records at the end of a chain contain a pointer value of X'FFFFFF'.

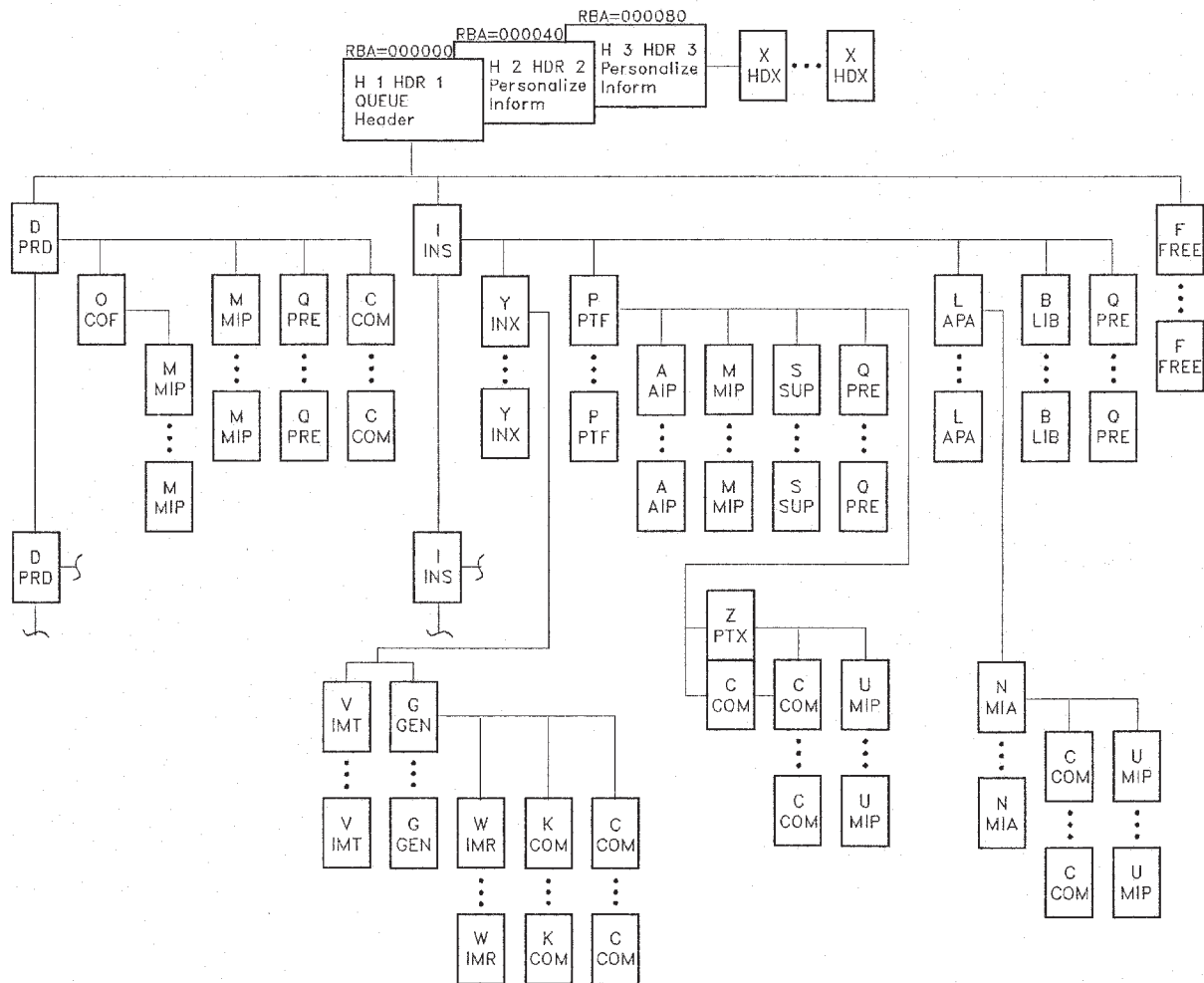


Figure 8. History File Queue Structure

Summary Information

System Programs Called by MSHP

MSHP may call one or more of the following system programs:

ASSEMBLY
 ESERV
 LIBR
 LNKEDT

For information on these programs, refer to the appropriate manual of the VSE/Advanced Functions library.

Figure 9 on page 14 shows the relationship between MSHP and the system programs it calls.

FUNCTION:	CALLED SYSTEM PROGRAMS:	CALLING MODULE:
APPLY/REVOKE/INCORPORATE	LIBR, LNKEDT	IKREMBDY IKRBKT
CORRECT/UNDO	LIBR, LNKEDT, ASSEMBLY, ESERV	IKRECORR IKRUNDO IKRPDZAP IKRRLZAP IKRSLZAP
INSTALL	LIBR, LNKEDT	IKRINSTL IKRISVBK IKRUPGR
RESTORE	LIBR	IKRINSTL
TAILOR	LIBR, LNKEDT, ASSEMBLY	IKRTAIL

Figure 9. MSHP-to-Called Program Relationship

Function	Module
APPLY	IKREAPL
ARCHIVE	IKREARC
BACKUP	IKREBCK
COPY	IKRECPY
CORRECT	IKRECOR
CREATE	IKRECRE
DUMP	IKREDMP
INCORPORATE	IKREINC
INSTALL	IKRESTL
LIST	IKRELST
LOOKUP	IKRELKP
MERGE	IKREMRG
PERSONALIZE	IKREPER
REMOVE	IKRERMV
RESIDENCE	IKRERDC
RESTORE	IKRERST
RETRACE	IKRERTC
REVOKE	IKRERVK
SELECT	IKRESEL
TAILOR	IKRETLR
UNDO	IKREUND

Figure 10. Semantic Checker: Function-to-Module Relationship

Function	Module
APPLY/REVOKE/INCORPORATE	IKRAPPLY
ARCHIVE	IKRARCH
COPY/BACKUP/RESTORE	IKRCOPY
CREATE	IKRCREAT
DUMP	IKRDEBUG
REMOVE	IKRDEL
CORRECT	IKRPDZAP
	IKRRLZAP
	IKRSLZAP
INSTALL	IKRINSTL
	IKRISVBK
	IKRUPGR
LOOKUP	IKRLKUP
MERGE	IKRMERGE
PERSONALIZE	IKRPERS
RESIDENCE	IKRRDCE
RETRACE	IKRRTC
SELECT	IKRSAPL
TAILOR	IKRTAIL
LIST SERVICETAPE	IKRSTMTS
UNDO	IKRUNDO

Figure 11. Processor Routines: Function-to-Module Relationship

Module Description

Convention of Presentation

The design information is described on a module level. For each module, the information consists of:

MODULE-NAME

This is the name of the MSHP module as it appears in the relocatable library. For example IKRAAPAR

DESCRIPTIVE-NAME

This is the expanded name of the mnemonic that follows IKR in the module name. For example, the descriptive name for mnemonic AAPAR in the module IKRAAPAR, is 'Archive local APAR fix'.

FUNCTION

This is a description of the entire module. Modules with multiple entry points will also have a more specific description following each entry point name.

PARAMETERS

This is a description of both the input and output parameters.

ENTRY-POINT

For modules with only one entry point:

1. The entry point name is the same as the module name.
2. The function description for the entry point is that of the module and it is not repeated here.

For modules with multiple entry points:

1. The name of each entry point is specified.
2. The function, invoked via, input, and output descriptions are specified for each entry point.

FUNCTION

This item is omitted for modules with only one entry point.

CALLED VIA

This is the PL/S CALL statement necessary to invoke the proper entry point. The parameters required by the CALL statement are identified as 'In', 'Out', or 'In/Out' to indicate whether they are used as input, output, or both.

INPUT

Any input other than the parameters listed in the CALL statement is specified here.

OUTPUT

Any output other than the parameters listed in the CALL statement are specified here.

For information on exits, external references, and register use, refer to Chapter 5.

IKRAAPAR

Archive APAR fix into permanent history file.

FUNCTION

Archive an APAR fix and all related information (affected modules and fix descriptions) into a permanent history file. A check for sufficient space is done before.

PARAMETERS

None

ENTRY-POINT

IKRAAPAR

CALLED VIA

CALL IKRAAPAR

INPUT

1. IKRXENT - interface to semantic checker
2. Permanent history file

OUTPUT

Updated permanent history file.

IKRACMP1

Check for free space for adding a component.

FUNCTION

Check for sufficient space to add or incorporate an installation record and its related pre/co-requisite records to a permanent history file.

PARAMETERS

None

ENTRY-POINT

IKRACMP1

CALLED VIA

CALL IKRACMP1

INPUT

1. IKRXENT - internal representation for function 'ARCHIVE'

OUTPUT

None

IKRACOMP

Archive component into permanent history file.

FUNCTION

If the specified component has not already been applied, an installation record for this component is introduced into the permanent history file together with the corresponding pre/co- requisite and conflicting information. If the component has to be applied, a new installation record is created, initialized, and queued to the header (HD1) record. If a 'PRE', 'COR' or 'CONF' keyword is specified, the related information is queued to the installation record (INS).

PARAMETERS

None

ENTRY-POINT

IKRACOMP

CALLED VIA

CALL IKRACOMP

INPUT

1. IKRXENT - interface to semantic checker
2. Permanent history file

OUTPUT

Updated permanent history file.

IKRAGEN

Archive generated member entry into permanent history file.

FUNCTION

Archives a generated member entry and related information (macro- table, data, and comments) into a permanent history file. Before archivation, a space check is performed for the needed space in the history file.

PARAMETERS

```
HFTYPE BIN FIXED(15) /* KIND OF HISTORY FILE TO ACCESS */
1 WRKBUFTB /* WORK BUFFER CONTROL TABLE */
/* GIVEN AS CALL INPUT PARAMETER */
2 BFADRMX BIN(8) /* MAXIMUM # OF WORK BUFFERS */
2 BFADRACT BIN(8) /* ACTUAL # OF OBTAINED BUFFERS */
2 BFADR TAB(*) POINTER /* ADDRESSES OF OBTAINED BUFFERS */

1 RETINFO /* RETURN CONTROL INFORMATION */
/* GIVEN AS CALL INPUT PARAMETER */
2 RETRQ BIN FIXED(8) /* RETURN MODE REQUEST IN ERROR CASE */
/* 1 : EXIT, 2 : RETURN */
2 RETCD BIN FIXED(8) /* RETURN CODE */
/* 0 : OK, 1 : HISTORY FILE FULL */
RETNO CONSTANT(1) BIN(8) /* RETURN MODE: EXIT */
REYES CONSTANT(2) BIN(8) /* RETURN MODE: RETURN */
RETOK CONSTANT(0) BIN(8) /* RETURN CODE: OK */
RETHF CONSTANT(1) BIN(8) /* RETURN MODE: HISTORY FILE FULL */

CHECKCTL BIN FIXED(8) /* CONTROL OF FUNCTION */
CKONLY CONSTANT(1) BIN(8) /* CHECK SPACE ONLY */
CKARCH CONSTANT(2) BIN(8) /* CHECK SPACE AND ARCHIVE */
CKARNRW CONSTANT(3) BIN(8) /* CHECK SPACE AND ARCHIVE, NO REWR */
```

ENTRY-POINT

IKRAGEN

CALLED VIA

```
CALL IKRAGEN(HFTYPE,WRKBUFTB,RETINFO,CHECKCTL)
           (In ,In/Out ,In/Out ,In )
```

INPUT

1. IKRXENT - interface to semantic checker
2. Permanent history file

OUTPUT

Changed permanent history file.

IKRAPPLY

Prepare application of PTFs.

FUNCTION

This is the execution routine for the functions 'APPLY', 'INCORPORATE' and 'REVOKE'. The following cases are handled:

- If called under control of function 'INSTALL SERVICE/BACKOUT' and preparation is requested: Ensures that the needed component is installed in the system history file. For 'INSTALL BACKOUT' the internal representation block (IRP) is filled with the data found for the specified PTF in the system history file. Archives the IRP information into the system history file.
- If called under control of function 'INSTALL SERVICE/BACKOUT', and application is requested: Read from SYSIPT all MSHP comment records between the 'DATA' statement and the first LIBRARIAN control statement. Invokes the routine which picks up the data.
- If called under control of function 'LIST SERVICETAPE': Archives the specified component as well as the specified PTF into the auxiliary history file.
- If called directly for function 'APPLY', 'REVOKE', or 'INCORPORATE': Opens the system history file. Ensures that the needed component is installed. For 'REVOKE' the internal representation block (IRP) is filled with the data found in the system history file for that PTF. Invokes the second execution routine to do the library update and history file update.

PARAMETERS

None

ENTRY-POINT

IKRAPPLY

CALLED VIA

CALL IKRAPPLY

INPUT

1. IKRXENT - internal representation block (IRP)
2. System history file
3. Auxiliary history file
4. SYSIPT

OUTPUT

1. IKRXENT - internal representation block (IRP)
2. Auxiliary history file.

IKRAPRD

Archive product into permanent history file.

FUNCTION

Archive a product and all related information (affected phases, modules and macros, requirements, replaced components and comments) into a permanent history file. Search for the requested product. If the product is not in the history file, the product information is queued to the header record. If a 'PRE', 'COR' or 'CONF' keyword is specified, the related information is queued to the product record .

PARAMETERS

None

ENTRY-POINT

IKRAPRD

CALLED VIA

CALL IKRAPRD

INPUT

1. IKRXENT - Interface to semantic checker
2. Permanent history file

OUTPUT

Changed permanent history file.

IKRAPTF

Archive PTF into permanent history file.

FUNCTION

Archive a PTF and all related information (resolved APARS, affected phases, modules and macros, requirements, superseded PTFs and comments) into a permanent history file.

PARAMETERS

None

ENTRY-POINT

IKRAPTF

CALLED VIA

CALL IKRAPTF

INPUT

1. IKRXENT- interface to semantic checker
2. Permanent history file

OUTPUT

Changed permanent history file.

IKRARCH

Archive into permanent history file

FUNCTION

Open permanent history file, find the record type to be archived and call the corresponding ARCHIVE-routine. Finally, rewrite the buffers and close open data sets.

PARAMETERS

1	WRKBUFFB		/* WORK BUFFER CONTROL TABLE	*/
			/* GIVEN AS CALL INPUT PARAMETER	*/
2	BFADRMX	BIN(8)	/* MAXIMUM # OF WORK BUFFERS	*/
2	BFADRACT	BIN(8)	/* ACTUAL # OF OBTAINED BUFFERS	*/
2	BFADRTAB(*)	POINTER	/* ADDR. OF OBTAINED BUFFERS	*/

ENTRY-POINT

IKRARCH

CALLED VIA

CALL IKRARCH(WRKBUFFB)
(In/Out)

INPUT

1. IKRXENT - interface to semantic checker
2. Permanent History File

OUTPUT

Changed permanent history file.

IKRBARDT

Build and sort APAR record table.

FUNCTION

Builds two types of sort records for each component for which it is called (provided the component has not been made obsolete): TYPE 1 SORT RECORDS for all APARS resolved by PTFS (correctly applied and neither superseded nor revoked) and TYPE 2 SORT RECORDS for all LOCAL/APAR corrections. Insertion into the SORT TABLE is by APAR number combined with the PTF number and the one byte TYPE field. This results in all APARs fixed by PTFs coming first followed by all LOCAL/APAR fixes; both groups are in APAR number sequence.

PARAMETERS

1	SORTVAL		
2	ASORTST	PTR(32)	/* SORT AREA START ADDRESS */
2	ARDTBLST	PTR(32)	/* RECORD TABLE START ADDRESS */
2	ASQTLST	PTR(32)	/* SEQUENCE TABLE START ADDRESS */
2	ASORTND	PTR(32)	/* SORT AREA END ADDRESS */
2	ARDTBLND	PTR(32)	/* RECORD TABLE END ADDRESS */
2	ASQTLND	PTR(32)	/* SEQUENCE TABLE END ADDRESS */
2	INSRDN	FIXED(15)	/* INSERTED RECORD NUMBER */
2	NINSRDN	FIXED(15)	/* NOT INSERTED RECORD NUMBER */
2	LSRTRD	FIXED(8)	/* LENGTH OF SORT RECORD */
2	LSF	FIXED(8)	/* LENGTH OF SORT FIELD */
2	FSTCMPRB	PTR(24)	/* FIRST COMPONENT RBA */
2	FSTFTRRB	PTR(24)	/* FIRST FEATURE RBA */
2	NXTCMPRB	PTR(24)	/* NEXT COMPONENT RBA */

ENTRY-POINT

IKRBARDT

CALLED VIA

CALL IKRBARDT(SORTVAL)
(In/Out)

INPUT

None

OUTPUT

APAR record table and sequence table.

IKRBGRDT

Build generated member record table.

FUNCTION

Builds a generated member sort record for each generated member of the component. The generated member sort record contains:

1. The sort field (generated member name and one byte containing G)
2. The RBA of the generated member
3. The RBA of the component
4. The library type of the generated member

PARAMETERS

1 SORTVAL			
2 ASORTST	PTR(32)	/* SORT AREA START ADDRESS	*/
2 ARDTBLST	PTR(32)	/* RECORD TABLE START ADDRESS	*/
2 ASQTBLST	PTR(32)	/* SEQUENCE TABLE START ADDRESS	*/
2 ASORTND	PTR(32)	/* SORT AREA END ADDRESS	*/
2 ARDTBLND	PTR(32)	/* RECORD TABLE END ADDRESS	*/
2 ASQTBLND	PTR(32)	/* SEQUENCE TABLE END ADDRESS	*/
2 INSRDN	FIXED(15)	/* INSERTED RECORD NUMBER	*/
2 NINSRDN	FIXED(15)	/* NOT INSERTED RECORD NUMBER	*/
2 LSRTRD	FIXED(8)	/* LENGTH OF SORT RECORD	*/
2 LSF	FIXED(8)	/* LENGTH OF SORT FIELD	*/
2 FSTCMPRB	PTR(24)	/* FIRST COMPONENT RBA	*/
2 FSTFTRRB	PTR(24)	/* FIRST FEATURE RBA	*/
2 NXCMPRB	PTR(24)	/* NEXT COMPONENT RBA	*/

ENTRY-POINT

IKRBGRDT

CALLED VIA

CALL IKRBGRDT(SORTVAL)
(In/Out)

INPUT

None

OUTPUT

Record table and sequence table.

IKRBINS

Binary insert in record table.

FUNCTION

Inserts the sort record, passed as a parameter, into the sort record table. The record itself is entered in the next free slot of the sort record table.

An index, locating the record in the sort record table, is entered in the sort sequence table, such that the indices in the sort sequence table from right to left (higher to lower addresses) map the records in the sort record table in ascending order relative to the sort field of the records (EBCDIC collating sequence). The binary search technique is used to make the sort sequence table entries.

PARAMETERS

1 SORTVAL			
2 ASORTST PTR(32)		/* SORT AREA START ADDRESS	*/
2 ARDTBLST PTR(32)		/* RECORD TABLE START ADDRESS	*/
2 ASQTBLST PTR(32)		/* SEQUENCE TABLE START ADDRESS	*/
2 ASORTND PTR(32)		/* SORT AREA END ADDRESS	*/
2 ARDTBLND PTR(32)		/* RECORD TABLE END ADDRESS	*/
2 ASQTBLND PTR(32)		/* SEQUENCE TABLE END ADDRESS	*/
2 INSRDN FIXED(15)		/* INSERTED RECORD NUMBER	*/
2 NINSRDN FIXED(15)		/* NOT INSERTED RECORD NUMBER	*/
2 LSRTRD FIXED(8)		/* LENGTH OF SORT RECORD	*/
2 LSF FIXED(8)		/* LENGTH OF SORT FIELD	*/
2 FSTCMPRB PTR(24)		/* FIRST COMPONENT RBA	*/
2 FSTFTRRB PTR(24)		/* FIRST FEATURE RBA	*/
2 NXTCMPRB PTR(24)		/* NEXT COMPONENT RBA	*/
1 NRD		/* SORT RECORD FOR INSERTION	*/
2 NSF CHAR(12)		/* NEW SORT FIELD	*/
2 NDF CHAR(16)		/* NEW DATA FIELD	*/

ENTRY-POINT

IKRBINS

CALLED VIA

CALL IKRBINS(SORTVAL,NRD)
(In/Out ,In)

INPUT

None

OUTPUT

Record table and sequence table.

IKRBIO

I/O routine for backout PTF generation.

FUNCTION

Punches the submitted data on either SYSPCH or - if called for function INSTALL SERVICE - on the backout tape.

PARAMETERS

WORKA CHAR(80) /* data to be written */

ENTRY-POINT

IKRBIO

CALLED VIA

CALL IKRBIO(WORKA)

INPUT

None

OUTPUT

1. SYSPCH
2. The backout tape.

IKRBKPR

Product backup support

FUNCTION

Backup to tape, for selected products, the production and generation sublibraries where the products reside in. Other products that reside in those libraries are also backed up. Checks are done for completeness of all products being in the concerned sublibraries. The history information for those products is collected in a work history file which will be backed up before the sublibraries. The LIBRARIAN is loaded and used by submitting the BACKUP command.

PARAMETERS

None

ENTRY-POINT :

IKRBKPR

CALLED VIA

CALL IKRBKPR

INPUT

IKXCPY - IRP with command information

OUTPUT

Tape with backed up product history file and sublibraries.

IKRBKT

Generate the data for a backout PTF.

FUNCTION

Generates the body of a backout PTF: A librarian control statement is generated according to the name and type of the requested member and calls the librarian to punch it out. The punched cards are picked up by the receive appendage, which directs them to the backout device. The librarian return code is submitted to the caller.

PARAMETERS

1 BKMOD	/* Library member to be punched */
3 LIBMBRLB CHAR(1)	/* Library member type */
3 LIBMBRCL CHAR(1)	/* Library member class */
3 LIBMBRNM CHAR(8)	/* Library member name */
RTCODE BIN(15)	/* Return code (as given by LIBR)*/

ENTRY-POINT

IKRBKT

CALLED VIA

```
CALL IKRBKT(BKMOD,RTCODE)
           ( In , Out )
```

INPUT

Accessed library(ies)

OUTPUT

Backout device (tape or SYSPCH).

IKRBKUP

Main root for PTFBKUP

FUNCTION

This is the main routine for BACKUP of the system history file; it controls a subset of the MSHP modules. To control the subset, an input matrix to copy the permanent history file to tape is prepared. The history file BACKUP is also used by the BACKUP function of the librarian.

PARAMETERS

None

ENTRY-POINT

IKRBKUP

CALLED VIA

Executing PTFBKUP which was linked by linkbook IKRLBKUP

INPUT

None

OUTPUT

None

IKRBPRDT

Build PTF record table.

FUNCTION

Builds a PTF sort record for each PTF of the component for which the module is called. The PTF sort record contains the sort field (PTF number + one byte containing 0), the RBA of the PTF, and the RBA of the component. If the PTF supersedes another PTF, then for each superseded PTF, a superseded-PTF record is also built and inserted into the sort table. A superseded-PTF sort record contains the number of the superseded PTF, the number of the superseding PTF, and a one byte type field of value 255 ('FF'X). The number of the superseded PTF and the type field make up the sort field. This results in the superseded-PTF record being inserted into the sort table right after the PTF sort record of the superseded PTF itself.

PARAMETERS

1 SORTVAL		
2 ASORTST PTR(32)	/* SORT AREA START ADDRESS	*/
2 ARDTBLST PTR(32)	/* RECORD TABLE START ADDRESS	*/
2 ASQTBLS PTR(32)	/* SEQUENCE TABLE START ADDRESS	*/
2 ASORTND PTR(32)	/* SORT AREA END ADDRESS	*/
2 ARDTBLND PTR(32)	/* RECORD TABLE END ADDRESS	*/
2 ASQTBLSND PTR(32)	/* SEQUENCE TABLE END ADDRESS	*/
2 INSRDN FIXED(15)	/* INSERTED RECORD NUMBER	*/
2 NINSRDN FIXED	/* NOT INSERTED RECORD NUMBER	*/
2 LSRTRD FIXED(8)	/* LENGTH OF SORT RECORD	*/
2 LSF FIXED(8)	/* LENGTH OF SORT FIELD	*/
2 FSTCMRFB PTR(24)	/* FIRST COMPONENT RBA	*/
2 FSTFTRRB PTR(24)	/* FIRST FEATURE RBA	*/
2 NXTCMRFB PTR(24)	/* NEXT COMPONENT RBA	*/

ENTRY-POINT

IKRBPRDT

CALLED VIA

CALL IKRBPRDT(SORTVAL)
(IN/OUT)

INPUT

None

OUTPUT

Sort record table and sort sequence table.

IKRCGT

Access to PTF file and generation file.

FUNCTION

Provides all functions needed to access the files of the service tape or the generation file (open, get, skip). For details see the function descriptions for each entry point.

PARAMETERS

WORKA CHAR(80)	/* read-in area	*/
EOF BIT(1)	/* 1:=end of file reached	*/

ENTRY-POINT

IKRCGTOP

FUNCTION

Determine the device on which the requested file resides, choose the correct DTF and open the file. The DTF address is stored as 'CUMFILE' in the MSHP file list of the COMREG.

CALLED VIA

CALL IKRCGTOP

INPUT

CUMFILE

- either the generation file (if called for function SELECT GENFILE). Logical record length 80 or 81 bytes, unblocked. If on disk, filename must be 'GENFILE'.
- or a file of the service tape (if called for function INSTALL SERVICE or LIST SERVICETAPE). Logical record length 80 bytes, blocked with blocksize of 3440 bytes.

OUTPUT

None

ENTRY-POINT

IKRCGT

FUNCTION

Open the requested file (if not yet done), read the next logical record and submit it to caller. The address of the DTF used is found as CUMFILE in the MSHP file list of the MSHP communication region. In case where end of file is reached, the corresponding indication is set.

CALLED VIA

CALL IKRCGT(WORKA,EOF)
 (Out ,Out)

INPUT

CUMFILE - see description for IKRCGTOP, above.

OUTPUT

None

ENTRY-POINT

IKRCGTPS

FUNCTION

Skip to end of tape file (if called for function INSTALL SERVICE/BACKOUT or LIST SERVICETAPE).

CALLED VIA

CALL IKRCGTPS

INPUT

CUMFILE - see description above.

OUTPUT

None.

IKRCHECK

Check if pre/co/not-requirements are satisfied.

FUNCTION

Handles requirement checking and printing of failing requirements. For details see description for each entry point.

```
IKRCHECK /* check the requirements */
IKRCHEPR /* print failing requirement */
```

PARAMETERS

```
HFTYPE BIN(15) /* type of history file */
REQPTR PTR(32) /* Pointer to requirement block */

1EXITINFO /* exit information: */
 2 EXITTYPE BIN(15) /* type of exit (CANCEL, RETURN)*/
 2 EXITRETC BIN(15) /* exit return code */

1COREQ /* coreq information */
 2 COREQST BIN(15) /* status (accepted or not) */
 2 COREQRC BIN(15) /* return code */
```

ENTRY-POINT

```
IKRCHECK /* check the requirements */
```

FUNCTION

Check if the requirements as specified in the submitted requirement block are satisfied. The checks are done in the system history file or in both (system and auxiliary) depending on the value of parameter 1.

If both history files are to be checked, the system history file is scanned first.

Negative requirements are always checked against the system history file only.

Superseded PTFs are treated as being applied.

If 'RETURN' is requested, the result is indicated by the exit return code (1:=satisfied, 2:=not satisfied).

If 'CANCEL' is requested, MSHP is terminated if a requirement is not satisfied.

In both cases the list of all failing requirements is printed. If 'RETURN WITHOUT PRINTING' is requested, nothing is printed but the buffer containing all failing requirements is saved.

Via the COREQ parameter you may specify to accept even if a coreq requirement is not met. Such a situation is indicated via the coreq return code.

CALLED VIA

```
CALL IKRCHECK(HFTYPE,REQPTR,EXITINFO,COREQ )
          ( In , In , In/Out ,In/Out)
```

INPUT

1. System history file
2. Auxiliary history file (if checking in both requested)

OUTPUT

List of failing requirements (if printing requested)

ENTRY-POINT

```
IKRCHEPR /* print failing requirements */
```

FUNCTION

Prints the requirements which failed during a previously invocation of IKRCHECK with 'RETURN WITHOUT PRINTING'.

CALLED VIA

```
CALL IKRCHEPR
```

INPUT

None

OUTPUT

List of failing requirements.

IKRCKASS

Check actual assignment against requested device type.

FUNCTION

This routine checks the actual assignment of a logical unit and retrieves all needed device dependent informations for it.

The unit may be specified as

- a logical unit number in CCB format
- the last three characters of the logical unit (SYSxxx)
- the last two characters of the file name (IJSYSxx)

In case where the file name was submitted, an indication is returned whether the file is VSAM controlled or not.

The error handling action depends on what the caller has requested: If 'CANCEL' was specified, the user is informed and MSHP is terminated. Otherwise the 'NOT ASSIGNED' return code is set and control is given back to the caller.

Following subfunctions are supported:

```
IKRCKASS /* check assignment */
IKRCKADI /* check assignment and provide device characteristics */
```

PARAMETERS

```
1LU CHAR(3) /* logical unit information */
2 LUIND CHAR(1) /* type indicator: */
/* 'X':=CCB form 'F':=file name */
2 LUX BIT(16) /* logical unit in CCB format */
3 LUF CHAR(2) /* logical unit as file name */

UTYPE BIN(15) /* device type to be checked: */
/* disk:=1,2 tape:=3 any:=4 */

1ULAYOUT /* device characteristics */
2 TRKS BIN(32) /* tracks per cylinder (CKD) */
/* 'FF000000' (FBA) */
2 BLKS BIN(32) /* blocks per track (CKD), */
/* block size (FBA) */
2 CAPY BIN(32) /* Bytes per track (CKD) */
2 DTFD CHAR(1) /* DTF Device type code (CKD) */

1EXITINFO /* Exit information */
2 EXITTYPE BIN(15) /* type: 1:=cancel, 2:=return */
2 EXITRETC BIN(8) /* return code: 1:=OK, 2:=failed */
2 EXITVSAM BIT(1) /* 1:=VSAM file */
```

ENTRY-POINT

```
IKRCKASS /* check assignment */
```

FUNCTION

Check for the specified logical unit whether the actual assignment matches the requested device type. The result is returned (if requested) in the EXITINFO parameter.

CALLED VIA

```
CALL IKRCKASS(LU,UTYPE,EXITINFO)
              (In, In , In/Out )
```

INPUT

LUBTAB, PUBTAB and label area

OUTPUT

None

ENTRY-POINT

```
IKRCKADI /* check assignment and provide device characteristics */
```

FUNCTION

Check for the specified logical unit whether the actual assignment matches the requested device type and return the result (if requested) to the caller.

In case where it matches, volume characteristics are returned for DASDs: For FBA DASDs the block size only, for CKD the number of bytes per track, the number of blocks per track, the number of tracks per cylinder.

In order to retrieve the correct number of blocks per track, the block size must be specified at entry in parameter BLKS.

CALLED VIA

```
CALL IKRCKADI(LU,UTYPE,ULAYOUT,EXITINFO)
              (In, In , In/Out, In/Out )
```

INPUT

LUBTAB, PUBTAB, and label area

OUTPUT

None.

IKRCKCMP

Find all occurrences of component record.

FUNCTION

Search for all occurrences of the specified COMPONENT. If the COMPONENT is installed only once , the caller will get its CELL-ID back.

If the COMPONENT is installed several times, the user will be informed about all COMPONENTS/LEVELS installed and is asked to enter the requested LEVEL. The CELL-ID of the requested COMPONENT/LEVEL is returned to the caller.

PARAMETERS

HFTYPE BIN(15)	/* HISTORY FILE TYPE	*/
	/* (1=PERM,2=TEMP)	*/
RECTYPE BIN(15)	/* RECORD TYPE	*/
	/* (1=SHORTERM,2=LONGTERM)	*/
1CELLID	/* CELL-ID OF THE REQUESTED	*/
	/* RECORD	*/
3 ADDRESS POINTER(32)	/* ITS ADDRESS	*/
3 RBA POINTER(24)	/* ITS RBA	*/
1ARGUMENT	/* SEARCH ARGUMENT	*/
3 COMPON CHAR(9)	/* COMPONENT ID	*/
3 LVLARG CHAR(3)	/* LEVEL ID	*/

ENTRY-POINT

IKRCKCMP

CALLED VIA

CALL IKRCKCMP (HFTYPE,RECTYPE,CELLID,ARGUMENT)
(In , In , Out , In)

INPUT

None

OUTPUT

None.

IKRCKCTL

Check control statements of system control program.

FUNCTION

Check if the submitted input line is a 'PHASE' statement (as needed for linkage editor), or a 'CATALR', or a 'CATALS', or a 'CATALOG' statement (as needed for the LIBRARIAN).

If so, ensure that it is correct in order to generate the member which is denoted in the IRP and indicate that the checking has been done.

PARAMETERS

CARD CHAR(80)	/* Line to be checked	*/
OK BIT(1)	/* 1:=checks successful	*/

ENTRY-POINT

IKRCKCTL

CALLED VIA

```
CALL IKRCKCTL(CARD,OK )  
           ( In ,Out)
```

INPUT

IKRXENT - internal representation block for function TAILOR

OUTPUT

IKRXENT - internal representation block for function TAILOR

IKRCKD

Count-key-data disk DTFs used by MSHP.

FUNCTION

Table-module (not executable) containing CKD DTFs.

PARAMETERS

None

ENTRY-POINT

IKRCKD

CALLED VIA

NOT CALLED

INPUT

None

OUTPUT

None.

IKRCKFIX

Check local APAR/FIXES.

FUNCTION

Check if the specified phase/module/macro is already changed at the same location by a preceding local APAR/FIX.

PARAMETERS

1	SHARG		/* SEARCH ARGUMENT	*/
2	SHCOMP	CHAR(12)	/* COMPONENT ID	*/
2	SHMODID	CHAR(10)	/* PHASE/MODULE/MACRO ID	*/
3	*	CHAR(2)		
3	SHMOD	CHAR(8)	/* PHASE/MODULE/MACRO NAME	*/
2	SHNEXT	BIT(1)		
1	ADDRESS		/* CHANGE ADDRESS	*/
2	ADDRESS1	PTR	/* START ADDRESS OF CHANGE AREA	*/
2	ADDRESS2	PTR	/* END ADDRESS OF CHANGE AREA	*/
1	CONFLICT	BIT(1)	/* 1:= CONFLICT FOUND	*/

ENTRY-POINT

IKRCKFIX

CALLED VIA

```
CALL IKRCKFIX(SHARG,ADDRESS,CONFLICT)
           (In  ,In  ,Out  )
```

INPUT

IKRXENT - IRP for function CORRECT

OUTPUT

None.

IKRCKLV2

Semantic check routine for detail control statements.

FUNCTION

Check whether the specified detail control statement is semantically correct. Semantically means in the context of its function control statement, other detail statements and their sequence of invocation.

The result of the check is returned in parameter VALIDSW.

PARAMETERS

CMDNO CHAR(2)	/* detail control statement number*/	
10PENFL BIT(8)	/* sequence control flags.	*/
	/* last processed command was:	*/
2 AFFOPEN BIT(1)	/* 1:= AFFECTS	*/
2 REQOPEN BIT(1)	/* 1:= REQUIRES	*/
2 OROPEN BIT(1)	/* 1:= OR	*/
2 SCOPEN BIT(1)	/* 1:= SCAN	*/
VALIDSW BIT(1)	/* 0:=statement in error	*/

ENTRY-POINT

IKRCHECK

CALLED VIA

```
CALL IKRCHECK(CMDNO,OPENFL,VALIDSW)
      ( In ,In/Out, Out )
```

INPUT

1. The SIT (Parser output) of the specified command, pointed to by CRGSITP.
2. The IRP of the function to which the specified command is a detail control statement.

OUTPUT

None.

IKRCKPTF

Check for sufficient space for archiving a PTF.

FUNCTION

Calculate how many free records are needed to archive a PTF and all related information. Check if sufficient free records are available and if not, take the error exit.

PARAMETERS

None

ENTRY-POINT

IKRCKPTF

CALLED VIA

CALL IKRCKPTF

INPUT

IKRXENT - interface to semantic checker

OUTPUT

None.

IKRCLMSH

MSHCLOSE interface

FUNCTION

De-activate the corresponding appendage (send or receive)

PARAMETERS

DTF CHAR(L'DTFCP OR DTFDI), DTFDI, or DTFCP.

ENTRY-POINT

IKRCLMSH

CALLED VIA

CALL IKRCLMSH

INPUT

DTF CHAR(DTFCP OR DTFDI), DTFDI, or DTFCP.

OUTPUT

None.

IKRCLOSE

Close all opened data files.

FUNCTION

Close all files which have been opened previously by MSHP. The DTF and their status is found in the MSHP communication region.

PARAMETERS

None

ENTRY-POINT

IKRCLOSE

CALLED VIA

CALL IKRCLOSE

INPUT

1. IKRCOMRG - MSHP communication region
2. DTFs as denoted in IKRCOMRG

OUTPUT

None.

IKRCMDS

MSHP command directory

FUNCTION

This module defines the entire set of MSHP commands and their shortest possible abbreviation. Each command has a pointer associated to the command syntax table.

PARAMETERS

None

ENTRY-POINT

IKRCMD

CALLED VIA

CALL IKRCMDS

INPUT

Command verb

OUTPUT

Address of correct command syntax table.

IKRCNVRT

Convert to new product-id.

FUNCTION

This routine establishes the relations between "old" feature numbers (prior to Rel. 2.1) and the "new" product-ids and between components and their product code. Following subfunctions are supported:

```
IKRCNVFP /* convert feature number to product-id */
IKRCNVCP /* determine product code */
```

PARAMETERS

```
FEATUR CHAR(LENGTH(PRDVLV)) /* "old" feature number */
1PRODUCT /* "new" product-id */
2 PRDPPP CHAR(LENGTH(PRDPPP)) /* product code */
2 PRODRFF CHAR(LENGTH(PRDVLV)) /* product level */
1COMPON /* component-id */
2 CHAR(4)
2 COMPPPP CHAR(LENGTH(PRDPPP)) /* product-code part */
2 CHAR(2)
PRODCODE CHAR(LENGTH(PRDPPP)) /* product code */
FOUND BIT(1) /* 1:= product-id found */
```

ENTRY-POINT

IKRCNVFP

FUNCTION

Scan table FEATLIST and return product-id as it is found there for the specified feature number. If the feature number cannot be found in the table, indicate that to the caller.

CALLED VIA

```
CALL IKRCNVFP(FEATUR, PRODUCT, FOUND)
           ( In , Out , Out )
```

INPUT

FEATLIST - table that relates old feature numbers with new product-ids

OUTPUT

None

ENTRY-POINT

IKRCNVCP

FUNCTION

Scan table COMPLIST and return as product code what is found there for the specified component. If the component cannot be found in the table, return character 4-6 of the component-id as product code.

CALLED VIA

```
CALL IKRCNVCP(COMPON,PRODCODE)
           ( In , Out )
```

INPUT

COMPLIST - table which relates component-ids with their product code

OUTPUT

None.

IKRCNVTB

Conversion tables

FUNCTION

This module performs no function but contains two data areas which are used for feature number to product-id conversion:

FEATLIST - table which relates "old" feature numbers with the corresponding product-id

COMPLIST - table which relates component-ids with their product code.

PARAMETERS

None

ENTRY-POINT

IKRCNVTB

CALLED VIA

NOT CALLED

INPUT

None

OUTPUT

None.

IKRCOMR

MSHP communication region

FUNCTION

Reserves storage for the MSHP communication region.

PARAMETERS

None

ENTRY-POINT

None

CALLED VIA

NOT CALLED

INPUT

None

OUTPUT

None.

IKRCOPY

Copy history file (control module)

FUNCTION

Open specified source and target H.FILES and invoke the internal COPY routine for those files. The open data sets are closed after copying. In case of RESTORE of BACKUP, the tape is positioned before the tape mark written by CLOSE.

PARAMETERS

None

ENTRY-POINT

IKRCOPY

CALLED VIA

CALL IKRCOPY

INPUT

IKRXCPY - Internal representation (IRP) Source history file

OUTPUT

Target history file.

IKRCPIN

Copy history file to history file.

FUNCTION

Copy the source history file, block by block, into the target history file. Ignore those blocks which have never been used by now.

An error situation occurs if the target history file is too small to receive the source history file.

PARAMETERS

1. FROM BIN(15) - SOURCE HISTORY FILE TYPE
 2. TO BIN(15) - TARGET HISTORY FILE TYPE
- | | |
|------------|-----------------------------|
| | 1 := PERMANENT HISTORY FILE |
| VALUERANGE | 2 := TEMPORARY HISTORY FILE |
| | 3 := HISTORY FILE ON TAPE |

ENTRY-POINT

IKRCPIN

CALLED VIA

IKRCPIN(FROM,TO)
(IN ,IN)

INPUT

History file on disk (permanent or temporary) or on tape

OUTPUT

History file on disk (permanent or temporary) or on tape.

IKRCREAT

Create a new history file.

FUNCTION

Formats a history file of the requested type (system or auxiliary) in the following manner:

The first records are initialized as header records, containing the anchor point of the free-record chain, the indication that the first block only contains meaningful information, the free record counter and the release indicator.

All other records of the first block are initialized as free records and chained together.

The rest of the history file is not formatted at all.

PARAMETERS

None

ENTRY-POINT

IKRCREAT

CALLED VIA

CALL IKRCREAT

INPUT

IKRXCRE - the internal representation block (IRP)

OUTPUT

Formatted history file on disk.

IKRCUTHF

Cut off any unused blocks from history file.

FUNCTION

Determine which blocks of the specified history file have never been used. Return the highest used RBA to the caller and denote that RBA in the header records.

PARAMETERS

HFTYPE BIN(15) /* history file (1=system 2=aux) */

LRECRBA PTR(24) /* high-water-mark RBA */

ENTRY-POINT

IKRCUTHF

CALLED VIA

```
CALL IKRCUTHF(HFTYPE,LRECRBA)
           ( In , Out )
```

INPUT

Specified history file

OUTPUT

Specified history file.

IKRCVTLU

Convert logical unit.

FUNCTION

Convert logical unit:

- if input is given in character notation (e.g. 'REC') return the corresponding hexadecimal notation as used in CCB (e.g. '000A')
- if input is given in hexadecimal notation return the corresponding character notation.

PARAMETERS

LUC CHAR(3)

/* LU in character notation*/

LUX BIT(16)

/* LU in hexadecimal notation */

ENTRY-POINT

IKRCVTLU

CALLED VIA

```
CALL IKRCVTLU (LUC,LUX)
              ( In,Out)
              (Out, In)
```

INPUT

None

OUTPUT

None.

IKRDATE

Get jobdate.

FUNCTION

Following functions are supported:

1. Get date (compressed) and time.

The date is fetched from partition comreg and compressed. The time will be inserted.

2. Get date in printable format (MM/DD/YY).

Get date from partition comreg and submit it to caller in printable format.

3. Make compressed date printable (MM/DD/YY).

Decompress a compressed 2-byte date ('DATE2') into printable format ('DATE8' - 'MM/DD/YY').

4. Compress printable (short form 'MMDDYY') date.

This part of the routine compresses a date ('DATE6') of format 'MMDDYY' into 2 bytes into the format ('DATE2' - '7 bits YY, 4 bits MM, 5 bits DD'), and sets the time to zero.

PARAMETERS

ACTION BIN(8)	/* 1:=get compressed date */
	/* 2:=get printable date */
	/* 3:=decompress date */
	/* 4:=compress date */
1HFDATTIM BASED	/* date/time in h.file record */
2 HFDATE CHAR(2)	/* compressed date */
	/* YY: 7 bits,MM: 4 bits, */
	/* DD: 5 bits */
2 HFTIME CHAR(4)	/* time in hist. file record */

ENTRY-POINT

IKRDATE

CALLED VIA

```
CALL IKRDATE(ACTION,HFDATE-TIME)
           ( In , Out )
```

INPUT

Communication region

OUTPUT

None.

IKRDEBUG

Format and dump a permanent or temporary history file.

FUNCTION

Open and read the respective history file.

Format the input and print.

PARAMETERS

None

ENTRY-POINT

IKRDEBUG

CALLED VIA

CALL IKRDEBUG

INPUT

IKRXDMP (POINTED BY CRGMATRP)

OUTPUT

Printout of the history file.

IKRDEL

Delete permanent history file.

FUNCTION

Open the permanent history file and call the routine which deletes entries in the permanent history file. Rewrite the history file buffers afterward, and CLOSE files that are open.

PARAMETERS

None

ENTRY-POINT

IKRDEL

CALLED VIA

CALL IKRDEL

INPUT

Permanent history file

OUTPUT

Changed permanent history file.

IKRDEQ

Dequeue a history file record.

FUNCTION

Remove a record from a history file queue to which it actually belongs and chain it into another queue. The following subfunctions are supported:

IKRDEQ /* dequeue and enqueue into free record chain */
IKRDEQQU /* dequeue and enqueue into specified chain */

PARAMETERS

HFTYPE BIN(15) /* History file type */

1 OLDCELL /* cell of record to be dequeue */
3 OLDADD PTR(32) /* its address */
3 OLDRBA PTR(24) BDY(BYTE) /* its RBA */

1 NEWCELL /* cell of record where to queue*/
3 NEWADD PTR(32) /* its address */
3 NEWRBA PTR(24) BDY(BYTE) /* its RBA */

ENTRY-POINT

IKRDEQ /* dequeue and enqueue into free record chain */

FUNCTION

Remove a record from a history file queue to which it actually belongs and chain it into the free record chain.

CALLED VIA

CALL IKRDEQ(HFTYPE,OLDCELL)
(In , In)

INPUT

History file

OUTPUT

History file

ENTRY-POINT

IKRDEQQU /* dequeue and enqueue into specified chain */

FUNCTION

Remove a record from a history file queue to which it actually belongs and chain it to the specified record.

CALLED VIA

CALL IKRDEQQU(HFTYPE,OLDCELL,NEWCELL)
(In , In , In)

INPUT

History file

OUTPUT

History file.

IKRDIAGN

Issue diagnostics.

FUNCTION

The module translates an input message number into the corresponding full text message (using the message library, the message dictionary and the submitted text insertions) and writes it out to SYSLST, SYSLOG or both, depending upon the message.

For messages which ask for a reply, the reply is read. Triggered by the message type, the reply may also be printed.

PARAMETERS

P1 BIN FIXED (15)	/* message number	*/
P2 CHAR(*)	/* first insertion text	*/
P3 CHAR(*)	/* second insertion text	*/
P4 CHAR(*)	/* third insertion text	*/
P5 CHAR(*)	/* fourth insertion text	*/
1INSERT BASED	/* layout of insertion text	*/
2 INSLNGTH BIN FIXED (8)	/* length of insertion text	*/
2 INSTEXT CHAR (*)	/* insertion text itself	*/

ENTRY-POINT

IKRDIAGN

CALLED VIA

CALL IKRDIAGN(P1,P2,P3,P4,P5) /* P2,P3,P4,P5 optional */

INPUT

1. IKRDICT - message dictionary
2. IKRMLIB - message library

OUTPUT

Message on SYSLST and/or SYSLOG.

IKRDLELE

Delete element from history file.

FUNCTION

Remove the specified record type (PTF, APAR or PRODUCT) and all associated queues from the specified history file.

After this, the routine checks for module-in, APAR-records, and for comprised records. If there are any, the records are deleted from the specified history file record.

PARAMETERS

HFTYPE BIN(15)	/* History file type	*/
1ELEMCELL	/* Record to be removed	*/
2 ELEMADR PTR	/* Storage address	*/
2 ELEMABA PTR(24)	/* RBA in history file	*/

ENTRY-POINT

IKRDLELE

CALLED VIA

```
CALL IKRDLELE(HFTYPE,ELEMCELL)
           ( In , In )
```

INPUT

History file on disk

OUTPUT

Changed history file.

IKRDLIN

Delete entries in history file.

FUNCTION

Determine which type of record is to be removed from the history file. Set up the interface and call the proper routine which deletes elements and associated queues from a history file. If 'OBSOLETE' is specified, this is done for each obsolete component.

PARAMETERS

1. FILETYPE BIN(15) - HISTORY FILE TYPE
VALUERANGE: 1 := PERMANENT H.FILE
 2 := TEMPORARY H.FILE

ENTRY-POINT

IKRDLIN

CALLED VIA

IKRDLIN(FILETYPE)
 (IN)

INPUT

1. IKRXXRMV - interface to semantic checker
2. History file on disk

OUTPUT

Updated history file.

IKREDMP

Semantic check routine for function 'DUMP'

FUNCTION

Analyze function control statement 'DUMP' and translate the parsed information (SIT) into the internal representation (IRP). Pick up detail control statement 'DEFINE HISTORY', perform the semantic checks for it and call the label area change routine.

PARAMETERS

None

ENTRY-POINT

IKREDMP MODULE NAME

CALLED VIA

CALL IKREDMP

INPUT

IKR#DMP - SIT for function 'DUMP'

OUTPUT

IKRXDMP - Internal representation (IRP).

IKRDQFR

Dequeue a free record and pass it for updating.

FUNCTION

Remove a free record from the free record queue and give the obtained record to the caller.

The routine gets the header record and looks for the first free record.

If a free record is available ,the free record queue and the backward record queue are updated. The highest-used-RBA and the free record counter are updated with new values.

PARAMETERS

HFTYPE BIN(15)	/* HISTORY FILE TYPE	*/
	/* 1=PERM,2=TEMP	*/
RECTYPE BIN(15)	/* RECORD TYPE	*/
	/* 1=SHORTERM,2=LONGTERM	*/
1CELLID	/* CELL-ID OF THE REQ. RECORD	*/
3 ADDRESS POINTER(32)	/* ITS ADDRESS,NULL IF NONE	*/
3 RBA POINTER(24)	/* ITS RBA, UNDEF IF NONE	*/

ENTRY-POINT

IKRDQFR

CALLED VIA

CALL IKRDQFR(HFTYPE,RECTYPE,CELLID)
(In , In , Out)

INPUT

None

OUTPUT

None.

IKREAPL

Semantic check routine for function 'APPLY'

FUNCTION

Analyze the function control statement 'APPLY' and translate the parsed information (SIT) into the internal representation (IRP). Pick up all detail control statements for this function, perform the semantic checks for each of them and fill the extracted information into the IRP.

Check if all mandatory detail control statements have been specified and for inconsistencies between them.

PARAMETERS

None

ENTRY-POINT

IKREAPL

CALLED VIA

CALL IKREAPL

INPUT

IKR#APL - SIT for function 'APPLY'

OUTPUT

IKRXENT - Internal representation (IRP).

IKREARC

Semantic check routine for function 'ARCHIVE'

FUNCTION

Analyze the function control statement 'ARCHIVE' and translate the parsed information (SIT) into the internal representation (IRP). Pick up all detail control statements for this function, perform the semantic checks for each of them and fill the extracted information into the IRP.

Check if all mandatory detail control statements have been specified and for inconsistencies between them.

PARAMETERS

None

ENTRY-POINT

IKREARC

CALLED VIA

CALL IKREARC

INPUT

IKR#ARC - SIT for function 'ARCHIVE'

OUTPUT

IKRXENT - Internal representation (IRP).

IKREBCK

Semantic check routine for function 'BACKUP'

FUNCTION

Analyze function control statement 'BACKUP' and translate the parsed information (SIT) into the internal representation (IRP). Pick up detail control statement 'DEFINE HISTORY', perform the semantic checks for it and invoke the label area change routine for history backup or the product routine.

PARAMETERS

None

ENTRY-POINT

IKREBCK

CALLED VIA

CALL IKREBCK

INPUT

IKR#BCK - SIT for function 'BACKUP'

OUTPUT

IKRXCPY - internal representation (IRP).

IKRECOR

Semantic check for function 'CORRECT'

FUNCTION

Pick up all detail control statements for function 'CORRECT' and check them for validity. For 'AFFECTS' invoke the execution routine corresponding to the member type. For 'INVOLVES' invoke the linkage editor. Allow a 'REQUIRES' only at the beginning. If a problem occurs during correction, ensure that all fixes are undone again.

PARAMETERS

None

ENTRY-POINT

IKRECOR

CALLED VIA

CALL IKRECOR

INPUT

IKR#DFN - SIF for "DEFINE"

IKR#COR - SIF for "CORRECT"

IKR#AFF - SIF for "AFFECTS"

IKR#IVL - SIF for "INVOLVES"

Partition communication region

OUTPUT

IKRXENT - IRP for 'CORRECT'.

IKRECPY

Semantic check routine for function 'COPY'

FUNCTION

Analyze function control statement 'COPY' and translate the parsed information (SIT) into the internal representation (IRP). Pick up detail control statement 'DEFINE HISTORY', perform the semantic checks for it and call the label area change routine.

PARAMETERS

None

ENTRY-POINT

IKRECPY

CALLED VIA

CALL IKRECPY

INPUT

IKR#CPY - SIT for function 'COPY'

OUTPUT

IKRXCPY - Internal representation (IRP).

IKRECRE

Semantic check routine for function 'CREATE'

FUNCTION

Analyze function control statement 'CREATE' and translate the parsed information (SIT) into the internal representation (IRP). Pick up detail control statement 'DEFINE HISTORY', perform the semantic checks for it and invoke the label area change routine. Open is done and the create routine is called.

PARAMETERS

None

ENTRY-POINT

IKRECRE

CALLED VIA

CALL IKRECRE

INPUT

IKR#CRE - SIT for function 'CREATE'

OUTPUT

IKRXCRE - Internal representation (IRP).

IKREDMP

Semantic check routine for function 'DUMP'

FUNCTION

Analyze function control statement 'DUMP' and translate the parsed information (SIT) into the internal representation (IRP). Pick up detail control statement 'DEFINE HISTORY', perform the semantic checks for it and call the label area change routine.

PARAMETERS

None

ENTRY-POINT

IKREDMP MODULE NAME

CALLED VIA

CALL IKREDMP

INPUT

IKR#DMP - SIT for function 'DUMP'

OUTPUT

IKRXDMP - Internal representation (IRP).

IKREINC

Semantic check routine for function 'INCORPORATE'

FUNCTION

Convert the PARSER output of the function control statement 'INCORPORATE' into the internal representation. Read all specified detail control statements, check them for validity and insert their information also into the internal representation block. Prompt the user for the target library and sublibrary and store the specified names also into the IRP. Invoke the execution routine.

PARAMETERS

None

ENTRY-POINT

IKREINC

CALLED VIA

CALL IKREINC

INPUT

IKR#INC: SIT of 'INCORPORATE' statement from PARSER

CMDNO: Command number of detail statement from PARSER

OUTPUT

IKRXENT: Internal representation block (IRP) for function 'INCORPORATE'.

IKRELKP

Semantic check routine for function 'LOOKUP'

FUNCTION

Perform all necessary semantic checks, fill the interface matrix and call the execution routine for the function 'LOOKUP'.

PARAMETERS

None

ENTRY-POINT

IKRELKP

CALLED VIA

IKRELKP

INPUT

IKR#LKP - SIT for function 'LOOKUP'

OUTPUT

IKRXLKP - Interface matrix.

IKRELST

Semantic check routine for function 'LIST SERVICETAPE'

FUNCTION

Converts the PARSER output of the function control statement 'LIST SERVICETAPE' into the internal representation. Reads all specified detail control statements, checks them for validity and inserts their information also into the internal representation block. Invokes the execution routines according to the specified options.

PARAMETERS

None

ENTRY-POINT

IKRELST

CALLED VIA

CALL IKRELST

INPUT

IKR#LIST: SIT of 'LIST' statement from PARSER

CMDNO : Command number of detail statement from PARSER

OUTPUT

IKRXCUM : Internal representation block (IRP) for function 'LIST SERVICETAPE'.

IKREMBDY

Embody PTF or component data.

FUNCTION

This module performs two functions:

```
IKREMBDY /* incorporate a component */
IKREMBDA /* pick up data for PTF application */
```

The functional details are described below for each entry point.

PARAMETERS

```
CURRMEMB BIN(15) /* set of members to be processed */
CURRLIBI BIN(15) /* set of currently accessed */
/* sublibraries */
```

ENTRY-POINT

IKREMBDY

FUNCTION

Entry point for execution of function "INCORPORATE": It checks the requirements and ensures that the specified target sublibrary is accessible.

All members of the component are cataloged into the sublib and all requested link steps are performed.

A component entry and a corresponding product entry are created in the system history file.

CALLED VIA

```
CALL IKREMBDY
```

INPUT

1. IKRXENT - IRP for function INCORPORATE
2. Component data
3. System history file

OUTPUT

1. User libraries
2. System history file

ENTRY-POINT

IKREMBDA

FUNCTION

Entry point to pick up data for PTF application and to catalog them into the sublibraries in which its component resides. Parameter "CURRMEMB" determines which members are to be read from the data input device:

If CURRMEMB=0, all members are to be cataloged, otherwise only those which are element of the set "CURRMEMB". Parameter "CURRLIBI" denotes which sublibraries are currently accessed and in which order.

CALLED VIA

```
CALL IKREMBDA(CURRMEMB,CURRLIBI)
           ( In , In/Out )
```

INPUT

1. IKRXENT - IRP for function APPLY and REVOKE
2. PTF data
3. System history file

OUTPUT

1. User libraries
2. System history file.

IKREMRG

Semantic check routine for function 'MERGE'

FUNCTION

Analyze function control statement 'MERGE' and translate the parsed information (SIT) into the internal representation (IRP). Pick up detail control statement 'DEFINE HISTORY', perform the semantic checks for it and invoke the label area change routine.

PARAMETERS

None

ENTRY-POINT

IKREMRG

CALLED VIA :

CALL IKREMRG

INPUT

IKR#MRG - SIT for function 'MERGE'

OUTPUT

IKRXMRG - Internal representation (IRP).

IKREMU

Emulate history file.

FUNCTION

Determine the status of the specified history file and perform all emulations needed to reach the highest level.

Emulation from Rel.1 to Rel.2 format:

- Create header extension record.
- Change to new date format.
- Introduce anchor for residence information.

Emulation from Rel.2 to Rel.3 format:

- Introduce anchor for component extension record.
- Introduce in APAR record the anchors for influenced members records and for requirement records.

Emulation from Rel.3 to Rel.2.1 format:

- Introduce high-water-mark for highest used RBA and set the free-record counter.
- Remove all obsolete components.
- Remove all residence information.
- convert all feature numbers to new product-id via the conversion table. Features comprising components which are not installed are deleted. Features which cannot be converted (not found in conversion table) and which do not have COMPRISES-information are deleted. For those features which do not have COMPRISES-information, but whose components can be identified via the conversion table, comprises information is built.
- Convert all component-ids into the new format (component name and level) via the conversion table. For those components which do not have a corresponding product entry, such an entry is created: The product code is determined via the conversion table and the level is extracted from the release number.
- Reorganize the PTFs: Queue all PTFs to the component on the correct level.

PARAMETERS

```
HFTYPE BIN(15)          /* history file type:      */
                        /* 1:=system, 2:=auxiliary */
```

ENTRY-POINT

IKREMU

CALLED VIA

```
CALL IKREMU(HFTYPE)
      ( In )
```

INPUT

Specified history file

OUTPUT

Specified history file.

IKREPER

Semantic check routine for function 'PERSONALIZE'

FUNCTION

Convert the PARSER output of the function control statement 'PERSONALIZE' into the internal representation. Read all specified detail control statements, check them for validity and insert their information also into the internal representation block.

Open the system history file, invoke the execution routine, and close all open files after execution.

PARAMETERS

None

ENTRY-POINT

IKREPER

CALLED VIA

CALL IKREPER

INPUT

IKR#PER : SIT of 'PERSONALIZE' statement from PARSER

CMDNO : Command number of detail statement from PARSER

OUTPUT

IKRXPER : Internal representation block (IRP) for function 'PERSONALIZE'.

IKRERDC

Semantic check routine for function 'RESIDENCE'

FUNCTION

Convert the PARSER output of the function control statement 'RESIDENCE' into the internal representation.

Read all specified detail control statements, check them for validity, and insert their information also into the internal representation block.

Open the system history file, invoke the execution routine, and close all open files after execution.

PARAMETERS

None

ENTRY-POINT

IKRERDC

CALLED VIA

CALL IKRERDC

INPUT

IKR#RDC : SIT of 'RESIDENCE' statement from PARSER

CMDNO : Command number of detail statement from PARSER

OUTPUT

IKRXRDC : Internal representation block (IRP) for function 'RESIDENCE'.

IKRERMV

Semantic check routine for function 'REMOVE'

FUNCTION

Analyze the function control statement 'REMOVE' and translate the parsed information (SIT) into the internal representation (IRP). Fill the extracted information into the IRP , pick up detail control statement 'DEFINE HISTORY', and perform the semantic checks for it. Invoke the label area change and the execution routine for function 'REMOVE'.

PARAMETERS

None

ENTRY-POINT

IKRERMV

CALLED VIA

CALL IKRERMV

INPUT

IKR#RMV - SIT for function 'REMOVE'

OUTPUT

IKRXRMV - Internal representation (IRP).

IKRERST

Semantic check routine for function 'RESTORE'

FUNCTION

Convert the PARSER output of the function control statement 'RESTORE' into the internal representation.

Read all specified detail control statements, check them for validity, and insert their information also into the internal representation block.

Invoke the execution routines according to the specified subfunction (RESTORE HISTORY or RESTORE PRODUCT/SYSRES).

PARAMETERS

None

ENTRY-POINT

IKRERST

CALLED VIA

CALL IKRERST

INPUT

IKR#RST : SIT of 'RESTORE' statement from PARSER

CMDNO : Command number of detail statement from PARSER

OUTPUT

IKRXSTL : Internal representation block (IRP) for function 'RESTORE PRODUCT/SYSRES'

IKRXCPY : Internal representation block (IRP) for function 'RESTORE HISTORY'.

IKRERTC

Semantic check routine for function 'RETRACE'

FUNCTION

Analyze the function control statement 'RETRACE' and translate the parsed information (SIT) into the internal representation (IRP). Fill the extracted information into the IRP and invoke the execution routine for function 'RETRACE'.

PARAMETERS

None

ENTRY-POINT

IKRERTC

CALLED VIA

CALL IKRERTC

INPUT

IKR#RTC - SIT for function 'RETRACE'

OUTPUT

IKRXRTC - Interface matrix.

IKRERVK

Semantic check routine for function 'REVOKE'

FUNCTION

Analyze the function control statement 'REVOKE' and translate the parsed information (SIT) into the internal representation (IRP). Pick up all detail control statements for this function, perform the semantic checks for each of them, and fill the extracted information into the IRP.

Check if all mandatory detail control statements have been specified and for inconsistencies between them.

PARAMETERS

None

ENTRY-POINT

IKRERVK

CALLED VIA

CALL IKRERVK

INPUT

IKR#RVK - SIT for function 'REVOKE'

OUTPUT

IKRXENT - Internal representation (IRP).

IKRESEL

Semantic check routine for function 'SELECT'

FUNCTION

Convert the PARSER output of the function control statement 'SELECT' into the internal representation.

Read all specified detail control statements, check them for validity, and insert their information also into the internal representation block.

Ensure that at least one member has been selected for generation. Ensure that all needed disk work files are assigned. Invoke the execution routine.

PARAMETERS

None

ENTRY-POINT

IKRESEL

CALLED VIA

CALL IKRESEL

INPUT

IKR#LIST: SIT of 'SELECT' statement from PARSER

CMDNO : Command number of detail statement from PARSER

OUTPUT

IKRXCUM : Internal representation block (IRP) for function 'SELECT'.

IKRESTL

Semantic check routine for function 'INSTALL'

FUNCTION

Convert the PARSER output of the function control statement 'INSTALL' into the internal representation.

Read all specified detail control statements, check them for validity, and insert their information also into the internal representation block.

Invoke the execution routines according to the specified subfunction (INSTALL SERVICE/BACKOUT or INSTALL PRODUCT/SYSRES).

PARAMETERS

None

ENTRY-POINT

IKRESTL

CALLED VIA

CALL IKRESTL

INPUT

IKR#STL : SIT of 'INSTALL' statement from PARSER

CMDNO : Command number of detail statement from PARSER

OUTPUT

IKRXSTL : Internal representation block (IRP) for function 'INSTALL PRODUCT/SYSRES'

IKRXCUM : Internal representation block (IRP) for function 'INSTALL SERVICE/BACKOUT'.

IKRETLR

Semantic check routine for function 'TAILOR'

FUNCTION

Convert the PARSER output of the function control statement 'TAILOR' into the internal representation.

Read all specified detail control statements, check them for validity, and insert their information also into the internal representation block.

Ensure correct sequence of called system control programs: To generate a phase, LNKEDT must be the last program called; for others, the librarian must be the last program called. Invoke the execution routine.

PARAMETERS

None

ENTRY-POINT

IKRETLR

CALLED VIA

CALL IKRETLR

INPUT

IKR#TLR : SIT of 'TAILOR' statement from PARSER

CMDNO : Command number of detail statement from PARSER

OUTPUT

IKRXENT : Internal representation block (IRP) for function 'TAILOR'.

IKREUND

Semantic check routine for function 'UNDO'

FUNCTION

Prints function control statement, builds interface for function "UNDO", picks up "DEFINE" statement, opens the system history file and calls the execution routine. After execution, it pages out the changed history file buffers and closes all data sets.

PARAMETERS

None

ENTRY-POINT

IKREUND

CALLED VIA

CALL IKREUND

INPUT

IKR#UND - SIT for function 'UNDO'

OUTPUT

IKRXENT - IRP for function 'UNDO'.

IKREXIT

Unrecoverable error exit

FUNCTION

IKREXIT calls IKRDIAGN (the message issuing routine) and passes to it the same arguments used to call IKREXIT; one of the following will then happen:

1. If MSHP was started from SYSRDR, the VSE macro CANCEL is used to terminate MSHP; and SYSIPT, which may contain MSHP control statements, is flushed.
2. If MSHP was started from SYSLOG, the VSE macro EOJ is used to terminate MSHP.
3. In the special case that IKREXIT is called while applying a PTF under the control of SELECT or ACCEPT, only the APPLY function is terminated, message 44 is issued, and control is returned to the SELECT/ACCEPT/APPLY processor IKRSAPL.

PARAMETERS

None

ENTRY-POINT

IKREXIT

CALLED VIA

CALL IKREXIT

INPUT

Pointer in register one to parameter list

Two byte message number up to four insertion texts to replace corresponding place holders in the message skeleton. Each insertion text is structured as follows:

A one-byte count field

As many bytes of the text as the count field indicates.

OUTPUT

Message on SYSLST and/or SYSLOG.

IKRFIX

Set a history file buffer as 'longterm'.

FUNCTION

Determine the altered buffer number and set the longterm bit in its respective buffer table entry.

PARAMETERS

BUFFADDR POINTER(32) /* Address of the 'longterm' record */

ENTRY-POINT

IKRFIX

CALLED VIA

CALL IKRFIX(BUFFADDR)
(In)

INPUT

None

OUTPUT

None.

IKRFRBUF

Free an MSHP buffer.

FUNCTION

Remove the flag that disables obtaining the respective buffer.

PARAMETERS

ADDRESS POINTER(32) /* Address of the freed buffer */

ENTRY-POINT

IKRFRBUF

CALLED VIA

CALL IKRFRBUF(BUFADDR)

INPUT

None

OUTPUT

None.

IKRGOGO

Select MSHP jobs.

FUNCTION

Determines from information found in the IRP whether or not a specific PTF or TAILOR job is to be selected for application. Following subfunctions are supported:

```
IKRGOGO /* select PTF for coverletter printing */
IKRGOGMB /* select generated member for application */
IKRGOGSD /* select PTF for library update */
IKRGOGXP /* determine PTFs which must be excluded */
IKRGOGCP /* determine PTFs which can be accepted */
```

PARAMETERS

```
JOBNAME CHAR(LENGTH(SELPTFNO)) /* PTF to be tested */

1GENNAME, /* gener. member to test */
  2 GENCMP CHAR(LENGTH(CUMCMP)), /* component name */
  3 GENCMPID CHAR(LENGTH(CUMCMPID)), /* component id */
  3 GENCMPLV CHAR(LENGTH(CUMCMPLV)), /* level */
  2 GENMEM CHAR(LENGTH(CUMGENID)) /* member id */

1FOUNDINF, /* component/PTF to test */
  2 FOUNDCMP CHAR(LENGTH(CUMCMP)), /* component name */
  3 FNDCMPID CHAR(LENGTH(CUMCMPID)), /* component id */
  3 FNDCMPLV CHAR(LENGTH(CUMCMPLV)), /* level */
  2 FOUNDPTF CHAR(LENGTH(SELPTFNO)) /* PTF name */

GO BIT(1) /* 1:=object is to be */
          /* applied */

OVER BIT(1) /* 1:=all selectable */
            /* have been processed */
```

ENTRY-POINT

```
IKRGOGO /* select PTF for coverletter printing */
```

FUNCTION

Determine whether the submitted PTF is element of 'CUMPTFL'. If so, indicate that it has been processed and indicate the check result to the caller. Denote also to caller if all members of 'CUMPTFL' have been processed.

CALLED VIA

```
CALL IKRGOGO(JOBNAME,GO ,OVER)
           ( In ,Out, Out)
```

INPUT

IKRXCUM - internal representation block (IRP)

OUTPUT

IKRXCUM - internal representation block (IRP)

ENTRY-POINT

IKRGOGMB /* select generated member for application */

FUNCTION

Determine whether the submitted generated member is element of 'CUMGENS'. If so, indicate that it has been processed and denote the check result to the caller. Denote also to caller if all members of 'CUMGENS' have been processed.

CALLED VIA

```
CALL IKRGOGMB(GENNAME,GO ,OVER)
      ( In ,Out, Out)
```

INPUT

IKRXCUM - internal representation block (IRP)

OUTPUT

IKRXCUM - internal representation block (IRP)

ENTRY-POINT

IKRGOGSD /* select PTF for library update */

FUNCTION

Determine whether the submitted PTF contains data which is to be picked up (= element of 'ISVPTFMB'). If so, denote the check result to the caller. Denote also to caller if all members of 'ISVPTFMB' have been processed.

CALLED VIA

```
CALL IKRGOGSD(JOBNAME,GO ,OVER)
      ( In ,Out, Out)
```

INPUT

IKRXCUM - internal representation block (IRP)

OUTPUT

IKRXCUM - internal representation block (IRP)

ENTRY-POINT

IKRGOGXP /* determine PTFs which must be excluded */

FUNCTION

Determine whether the submitted PTF has been explicitly excluded via an 'EXCLUDE PTF=...' detail control statement, and inform the caller about the result of the check.

CALLED VIA

```
CALL IKRGOGXP(JOBNAME,GO )  
      ( In ,Out)
```

INPUT

IKRXCUM - internal representation block (IRP)

OUTPUT

None

ENTRY-POINT

IKRGOGCP /* determine PTFs which can be accepted */

FUNCTION

Determine whether the submitted component/PTF can be accepted since it was explicitly included via an 'INCLUDE ...' detail statement or at least not explicitly excluded via an 'EXCLUDE ...' detail statement. Inform the caller about the check result and save the name of the accepted PTFs.

CALLED VIA

```
CALL IKRGOGCP(FOUNDINF,GO )  
      ( In ,Out)
```

INPUT

IKRXCUM - internal representation block (IRP)

OUTPUT

IKRXCUM - internal representation block (IRP).

IKRGTAPR

Get the first APAR record from a component record.

FUNCTION

Select the pointer to the first local fix (APAR) record and, if there is one, read it.

PARAMETERS

HFTYPE	BIN(15)	/* HISTORY FILE TYPE	*/
		/* 1=PERMANENT, 2=TEMPORARY	*/
RECTYPE	BIN(15)	/* HISTORY FILE RECORD TYPE	*/
		/* 1=SHORTERM, 2=LONGTERM	*/
CELLID		/* CELL-ID OF THE REQUESTED RECORD	*/
ADDRESS	POINTER(32)	/* ADDRESS OF THE REQUESTED RECORD	*/
		/* NULL IF THERE IS NONE	*/
RBA	POINTER(24)	/* RBA OF THE REQUESTED RECORD	*/
		/* UNDEF IF THERE IS NONE	*/
INSTCELL		/* CELL-ID OF THE COMPONENT RECORD	*/
INSTADDR	POINTER(32)	/* ADDRESS OF THE COMPONENT RECORD	*/
INSTRBA	POINTER(24)	/* RBA OF THE COMPONENT RECORD	*/

ENTRY-POINT

IKRGTAPR

CALLED VIA

CALL IKRGTAPR(HFTYPE,RECTYPE,CELLID,INSTCELL)
(In ,In ,Out ,In)

INPUT

None

OUTPUT

None.

IKRGTBUF

Get a 2K buffer from MSHP buffer pool.

FUNCTION

Search for an unused buffer in the MSHP buffer pool and, if found, give it to the requestor and update the buffer table.

INPUT

PARAMETERS

1) BUFTYPE BIN(15)	/*	BUFFER TYPE	*/
(IN)	/*	1 = LONGTERM	*/
	/*	2 = SHORTERM	*/
	/*	3 = WORKAREA	*/
2) BUFADDR POINTER(32)	/*	OBTAINED BUFFER ADDRESS	*/
(OUT)			

ENTRY-POINT

IKRGTBUF

CALLED VIA

CALL IKRGTBUF (BUFADDR,BUFTYPE)
(Out ,In)

INPUT

None

OUTPUT

None.

IKRGTCMP

Get the first component record.

FUNCTION

Rear the HEADER 1 record, extract the pointer to the first component record and, if there is one, read the component record.

PARAMETERS

HFTYPE	BIN(15)	/* HISTORY FILE TYPE	*/
		/* 1=PERMANENT, 2=TEMPORARY	*/
RECTYPE	BIN(15)	/* HISTORY FILE RECORD TYPE	*/
		/* 1=SHORTERM, 2=LONGTERM	*/
CELLID		/* CELL-ID OF THE REQUESTED RECORD	*/
ADDRESS	POINTER(32)	/* ADDRESS OF THE REQUESTED RECORD	*/
		/* NULL IF THERE IS NONE	*/
RBA	POINTER(24)	/* RBA OF THE REQUESTED RECORD	*/
		/* UNDEF IF THERE IS NONE	*/

ENTRY-POINT

IKRGTCMP

CALLED VIA

CALL IKRGTCMP (HFTYPE,RECTYPE,CELLID)
(In ,In ,Out)

INPUT

None

OUTPUT

None.

IKRGTFR

Get the first MSHP free record.

FUNCTION

Read the header record, extract the pointer to the first free record and, if there is one, read it.

PARAMETERS

HFTYPE	BIN(15)	/* HISTORY FILE TYPE	*/
(IN)		/* 1=PERMANENT, 2=TEMPORARY	*/
RECTYPE	BIN(15)	/* HISTORY FILE RECORD TYPE	*/
(IN)		/* 1=SHORTERM, 2=LONGTERM	*/
CELLID		/* CELL-ID OF THE REQUESTED RECORD	*/
(OUT)			
ADDRESS	POINTER(32)	/* ADDRESS OF THE REQUESTED RECORD	*/
		/* NULL IF THERE IS NONE	*/
RBA	POINTER(24)	/* RBA OF THE REQUESTED RECORD	*/
		/* UNDEF IF THERE IS NONE	*/

ENTRY-POINT

IKRGTFR

CALLED VIA

CALL IKRGTFR (HFTYPE,RECTYPE,CELLID)
(In ,In ,Out)

INPUT

None

OUTPUT

None.

IKRGTGEN

Get the first generated member record of a component.

FUNCTION

Select the pointer to the first generated member record and read it if there is one.

PARAMETERS

HFTYPE	BIN(15)	/* HISTORY FILE TYPE	*/
		/* 1=PERMANENT, 2=TEMPORARY	*/
RECTYPE	BIN(15)	/* HISTORY FILE RECORD TYPE	*/
		/* 1=SHORTERM, 2=LONGTERM	*/
INSTCELL		/* CELL-ID OF THE COMPONENT RECORD	*/
INSTADDR	POINTER(32)	/* ADDRESS OF THE COMPONENT RECORD	*/
INSTRBA	POINTER(24)	/* RBA OF THE COMPONENT RECORD	*/
CELLID		/* CELL-ID OF THE REQUESTED RECORD	*/
ADDRESS	POINTER(32)	/* ADDRESS OF THE REQUESTED RECORD	*/
		/* NULL IF THERE IS NONE	*/
RBA	POINTER(24)	/* RBA OF THE REQUESTED RECORD	*/
		/* UNDEF IF THERE IS NONE	*/

ENTRY-POINT

IKRGTGEN

CALLED VIA

CALL IKRGTGEN (HFTYPE,RECTYPE,CELLID,INSTCELL)
 (In ,In ,Out ,In)

INPUT

None

OUTPUT

None.

IKRGTGRP

Get the first product record.

FUNCTION

Read the header record, extract the pointer to the first product record and, if there is one, read it.

PARAMETERS

1) HFTYPE BIN(15)	/* HISTORY FILE TYPE	*/
	/* 1 = PERMANENT	*/
	/* 2 = TEMPORARY	*/
2) RECTYPE BIN(15)	/* H.FILE RECORD TYPE	*/
	/* 1 = LONGTERM	*/
	/* 2 = SHORTERM	*/
3) CELLID	/* CELL-ID OF THE REQUESTED RECORD	*/
	/* POINTER(32) ITS ADDRESS	*/
	/* (NULL IF NONE)	*/
	/* POINTER(24) ITS RBA	*/
	/* (UNDEF IF NONE)	*/

ENTRY-POINT

IKRGTGRP

CALLED VIA

CALL IKRGTGRP (HFTYPE,RECTYPE,CELLID)
(In ,In ,Out)

INPUT

None

OUTPUT

None.

IKRGTHD

Get MSHP header record 1.

FUNCTION

Read MSHP header record(record 1).

PARAMETERS

1) HFTYPE BIN(15)	/* HISTORY FILE TYPE	*/
	/* 1 = PERMANENT	*/
	/* 2 = TEMPORARY	*/
2) RECTYPE BIN(15)	/* H.FILE RECORD TYPE	*/
	/* 1 = LONGTERM	*/
	/* 2 = SHORTERM	*/
3) CELLID	/* CELL-ID OF THE REQUESTED RECORD	*/
	/* POINTER(32) ITS ADDRESS	*/
	/* (NULL IF NONE)	*/
	/* POINTER(24) ITS RBA	*/
	/* (UNDEF IF NONE)	*/

ENTRY-POINT

IKRGTHD

CALLED VIA

CALL IKRGTHD (HFTYPE,RECTYPE,CELLID)
(In ,In ,Out)

INPUT

None

OUTPUT

None.

IKRGTIMT

Get start of included macro table of a component.

FUNCTION

Locate the first record of the included macro table and read it if there is one.

PARAMETERS

HFTYPE BIN(15)	/* HISTORY FILE TYPE	*/
	/* 1=PERMANENT, 2=TEMPORARY	*/
RECTYPE BIN(15)	/* HISTORY FILE RECORD TYPE	*/
	/* 1=SHORTERM, 2=LONGTERM	*/
INSTCELL	/* CELL-ID OF THE COMPONENT RECORD	*/
INSTADDR POINTER(32)	/* ADDRESS OF THE COMPONENT RECORD	*/
INSTRBA POINTER(24)	/* RBA OF THE COMPONENT RECORD	*/
CELLID	/* CELL-ID OF THE REQUESTED RECORD	*/
ADDRESS POINTER(32)	/* ADDRESS OF THE REQUESTED RECORD	*/
	/* NULL IF THERE IS NONE	*/
RBA POINTER(24)	/* RBA OF THE REQUESTED RECORD	*/
	/* UNDEF IF THERE IS NONE	*/

ENTRY-POINT

IKRGTIMT

CALLED VIA

CALL IKRGTIMT (HFTYPE,RECTYPE,CELLID,INSTCELL)
 (In ,In ,Out ,In)

INPUT

None

OUTPUT

None.

IKRGTINF

Get the first influenced member record of a PTF or local/APAR fix.

FUNCTION

Select the pointer to the first influenced member record of a PTF or a local/APAR fix entry and read it if there is one.

PARAMETERS

HFTYPE	BIN(15)	/* HISTORY FILE TYPE	*/
		/* 1=PERMANENT, 2=TEMPORARY	*/
RECTYPE	BIN(15)	/* HISTORY FILE RECORD TYPE	*/
		/* 1=SHORTERM, 2=LONGTERM	*/
BASECELL		/* CELL-ID OF THE PTF/APAR RECORD	*/
BASEADDR	POINTER(32)	/* ADDRESS OF THE PTF/APAR RECORD	*/
BASERBA	POINTER(24)	/* RBA OF THE PTF/APAR RECORD	*/
CELLID		/* CELL-ID OF THE REQUESTED RECORD	*/
ADDRESS	POINTER(32)	/* ADDRESS OF THE REQUESTED RECORD	*/
		/* NULL IF THERE IS NONE	*/
RBA	POINTER(24)	/* RBA OF THE REQUESTED RECORD	*/
		/* UNDEF IF THERE IS NONE	*/

ENTRY-POINT

IKRGTINF

CALLED VIA

CALL IKRGTINF (HFTYPE,RECTYPE,CELLID,BASECELL)
 (In ,In ,Out ,In)

INPUT

None

OUTPUT

None.

IKRGTMSH

LIOCS GET interface

FUNCTION

Called by LIOCS GET. IKRGTMSH will call the SEND appendage and then move the record to the DTF IOAREA.

PARAMETERS

DTF CHAR(L'DTFDI OR DTFCP) DTFDI or DTFCP

ENTRY-POINT

IKRGTMSH IKRGTMEN When returning from the SEND appendage

CALLED VIA

CALL IKRGTMSH (DTFPARAM)
 (In)

INPUT

None

OUTPUT

DTF IOAREA1 and eventually IOREG.

IKRGTPTF

Get the first PTF record of a component.

FUNCTION

Select the pointer to the first PTF record and, if there is one, read it.

PARAMETERS

HFTYPE	BIN(15)	/* HISTORY FILE TYPE	*/
		/* 1=PERMANENT, 2=TEMPORARY	*/
RECTYPE	BIN(15)	/* HISTORY FILE RECORD TYPE	*/
		/* 1=SHORTERM, 2=LONGTERM	*/
INSTCELL		/* CELL-ID OF THE PTF RECORD	*/
BASEADDR	POINTER(32)	/* ADDRESS OF THE PTF RECORD	*/
BASERBA	POINTER(24)	/* RBA OF THE PTF RECORD	*/
CELLID		/* CELL-ID OF THE REQUESTED RECORD	*/
ADDRESS	POINTER(32)	/* ADDRESS OF THE REQUESTED RECORD	*/
		/* NULL IF THERE IS NONE	*/
RBA	POINTER(24)	/* RBA OF THE REQUESTED RECORD	*/
		/* UNDEF IF THERE IS NONE	*/

ENTRY-POINT

IKRGTPTF

CALLED VIA

CALL IKRGTPTF (HFTYPE,RECTYPE,CELLID,INSTCELL)
 (In ,In ,Out ,In)

INPUT

None

OUTPUT

None.

IKRGTRC

Get a history file record.

FUNCTION

Read the history file record specified by its RBA.

PARAMETERS

1) HFTYPE BIN(15)	/* HISTORY FILE TYPE	*/
(IN)	/* 1 = PERMANENT	*/
	/* 2 = TEMPORARY	*/
2) RECTYPE BIN(15)	/* H.FILE RECORD TYPE	*/
(IN)	/* 1 = LONGTERM	*/
	/* 2 = SHORTERM	*/
3) CELLID	/* CELL-ID OF THE REQUESTED RECORD	*/
(IN/OUT)	/* POINTER(32) ITS ADDRESS	*/
	/* (NULL IF NONE)	*/
	/* POINTER(24) ITS RBA	*/
	/* (UNDEF IF NONE)	*/

ENTRY-POINT

IKRGTRC

CALLED VIA

CALL IKRGTRC (HFTYPE,RECTYPE,CELLID)
(In ,In ,In/Out)

INPUT

None

OUTPUT

None.

IKRGTTAP

Get a block from a backed up history file.

FUNCTION

Read the next sequential tape block from a backed up history file, and set the end-of-file indicator if there are no more blocks.

PARAMETERS

BLOCK CHAR(HFBLKLEN)	/*	BLOCK TO BE WRITTEN	*/
EOF BIT(1)	/*	ON = END OF FILE	*/
	/*	OFF = RECORD AVAILABLE	*/

ENTRY-POINT

IKRGTTAP

CALLED VIA

CALL IKRGTTAP (BLOCK,EOF)
 (In ,Out)

INPUT

None

OUTPUT

None.

IKRGWA

Give or release the available storage area.

FUNCTION

Get the address of the available work area and release the buffers for the work area.

PARAMETERS

1) FUNCTION BIN(15)	/* 1 = GET WORK AREA ADDRESSES	*/
	/* 2 = RELEASE USED BUFFERS	*/
	/* 3 = GET NUMBER OF	*/
	/* AVAILABLE BUFFERS	*/
2) WAADDR POINTER(32)	/* WORK AREA START ADDRESS	*/
3) WAEND POINTER(32)	/* WORK AREA END ADDRESS	*/
4) BUFAV BIN(16)	/* NUMBER OF AVAILABLE BUFFERS	*/

ENTRY-POINT

IKRGWA

CALLED VIA

FUNCTION 1	CALL IKRGWA (FUNCTION,WAADDR,WAEND)
	(In ,Out ,Out)
FUNCTION 2	CALL IKRGWA (FUNCTION,WAADDR)
	(In ,In)
FUNCTION 3	CALL IKRGWA (FUNCTION,BUFAV)
	(In ,Out)

INPUT

None

OUTPUT

None.

IKRHDBKT

Generate a backout PTF header.

FUNCTION

Generate the header of backout PTF; this consists of the JCL statements to invoke MSHP correctly and the function control statement REVOKE. Punch the statements by calling the I/O routine for backout generation.

PARAMETERS

```
1PARAM                               /* REVOKE statement information */
3 RVKCMP                             /* component-ID */
  4 RVKCMP1 CHAR(4)
  4 RVKCMP2 CHAR(2)
  4 RVKCMP3 CHAR(3)
  4 RVKCMP4 CHAR(3)
3 RVKPTF CHAR(LENGTH(PTFNO)) /* PTF number */
```

ENTRY-POINT

IKRHDBKT

CALLED VIA

```
CALL IKRHDBKT(PARAM)
      ( Out )
```

INPUT

None

OUTPUT

1. SYSPCH
2. The backout tape.

IKRHDLV2

Handle detail control statements.

FUNCTION

Fill all information given by the specified detail control statement into the actual internal representation block (IRP), indicate the presence of the statement, and return control to the caller.

Special handling is done for following statements:

- For DEFINE HISTORY a partition user label is created and written into the label area.
- Some statements require to read more data. This is also done by this module.

PARAMETERS

CMDNO CHAR(2)	/* command number	*/
10PENFL BIT(8)	/* sequence control flags	*/
	/* last processed command:	*/
2 AFFOPEN BIT(1)	/* 1:= AFFECTS	*/
2 REQOPEN BIT(1)	/* 1:= REQUIRES	*/
2 OROPEN BIT(1)	/* 1:= OR	*/
2 SCOPEN BIT(1)	/* 1:= SCAN	*/
FFOUND BIT(24)	/* presence flag byte	*/
1WRKBUFTB	/* work buffer info	*/
2 BFADRMX BIN(8)	/* maximum # of buffers	*/
2 BFADRACT BIN(8)	/* actually obtained buffers	*/
2 BFADRTBL(*) PTR	/* addr. of obtained buffers	*/
1FREE	/* IRP free space information*/	
2 FREEPTR PTR	/* free space pointer	*/
2 FREESP BIN	/* free space counter	*/
ZAPSL BIN(15)	/* # of statements to process*/	

ENTRY-POINT

IKRHDLV2

CALLED VIA

```
CALL IKRHDLV2(CMDNO,OPENFL,FFOUND,WRKBUFTB,FREE,ZAPSL)
           ( In , Out , Out , In/Out , In/Out, In)
```

INPUT

1. The SIT (Parser output) of the specified command, pointed to by CRGSITP.
2. The IRP of the function to which the specified command is a detail control statement (pointed to by CRGMATRP).

OUTPUT

The IRP of the function to which the specified command is a detail control statement (pointed to by CRGMATRP).

IKRHSLA

Build a user label record.

FUNCTION

Create a label record with 1 extent and add it to the user label area of the running partition.

PARAMETERS

1. LOGUN	BIT(16)	/* logical unit	*/
2. DLBL	CHAR(7)	/* label filename	*/
3) 1 EXTENT			
3 TRACKST	BIN(32)	/* start track of history file	*/
3 TRACKNO	BIN(32)	/* number of tracks in extent	*/
3 SPLIT	BIN(32)	/* split cylinder information	*/
4) IDENT	CHAR(44)	/* file identifier	*/

ENTRY-POINT

IKRHSLA

CALLED VIA

CALL IKRHSLA (LOGUN,DLBL,EXTENT,IDENT)

INPUT

HISTORY FILETYPE, FILENAME, START TRACK

OUTPUT

LABL CHAR(104) LABEL RECORD.

IKRIMSMT

Insert included macros into member-specific macro table in core.

FUNCTION

Receive the included macro sent by the calling program and insert it into the IRP (ENTIML - format). Create a new table if no specific one is available for the current generated member.

PARAMETERS

1 MEMBER		/* id of included macro	*/
2 MEMLB	CHAR(1)	/* library qualifier	*/
2 MEMCL	CHAR(1)	/* concerned class	*/
2 MEMNAME	CHAR(8)	/* macro name	*/
1 WRKBUFTB		/* work buffer control table	*/
2 BFADRMX	BIN(8)	/* maximum # of work buffers	*/
2 BFADRACT	BIN(8)	/* actual # of obtained buffers	*/
2 BFADRTAB(*)	POINTER	/* addresses of obtained buffers	*/

ENTRY-POINT

IKRIMSMT

CALLED VIA

CALL IKRIMSMT(MEMBER,WRKBUFTB)
(In ,In/Out)

INPUT

IKRXENT - Internal representation (IRP)

OUTPUT

IKRXENT - Internal representation (IRP).

IKRINMSH

Receive included macros from called system control program.

FUNCTION

Receive the name of the included macro as given via the interface macro MSHINCL, transform it into the internal member format (as in IRP), and invoke the routine which inserts it into the included-macro-table of the IRP.

PARAMETERS

1MEMBER	/* id of included macro	*/
2 MEMSUBL CHAR(1)	/* sublib qualifier	*/
2 MEMNAME CHAR(8)	/* macro name	*/

ENTRY-POINT

IKRINMSH

CALLED VIA

CALL IKRINMSH(MEMBER)
(In)

INPUT

IKRXENT - Internal representation block (IRP)

OUTPUT

None.

IKRINSTL

Install or restore products or SYSRES.

FUNCTION

This module handles the installation of product and system shipment tapes: If called for function RESTORE, the members are picked up from the tape and copied into the specified target libraries/sublibraries. The shipped history file is restored as auxiliary history file into workfile IJSYS02.

If called for function INSTALL, the shipped history file information is used to perform all needed checks prior to copying the members into the user's libraries/sublibraries and merging the corresponding history information into the user's system history.

More details under "SEQUENCE OF OPERATION".

PARAMETERS

None

ENTRY-POINT

IKRINSTL

CALLED VIA

CALL IKRINSTL

INPUT

1. Distribution medium (tape,disk)
2. System history file (INSTALL only)

OUTPUT

1. User's libraries
2. System history file (INSTALL only)
3. Workfile IJSYS02 (RESTORE only)

SEQUENCE OF OPERATION:

Get the shipment history file into workfile IJSYS02 which is used as auxiliary history file. For old format tapes, the LIBRARIAN function RESTORE SCAN is called to skip the libraries and to inform MSHP about which type of libraries have be found.

Position the tape to start of shipped libraries.

Determine which products are to be installed and in which libraries/sublibraries their members are shipped.

Ensure that the requested part of the product is on the shipment medium. In case where separation of production and generation part is requested, make sure they are separately shipped.

Determine the target libraries/sublibraries. In case where they were not fully specified by the user, MSHP default using history file information (system and auxiliary).

For function INSTALL following checks are performed:

- Downlevel-checking: Members on the target system to which service has been applied, cannot be overwritten by a version having a lower service level without user's confirmation.
- High-level checking: Installing the generation part only is not allowed if its service level is higher than the level of the installed production part.
- The generation part may not be installed prior to the production part.
- For the products which will be superseded by the requested installation,er is ask for confirmation (keep/delete).
- Products which are shipped in one sublibrary cannot be restored into different sublibraries.
- Products which supersede each other may not be installed into the same sublibraries.
- User's system history file must be big enough to obtain the new history file information.

Terminate in case an un-recoverable error was detected.

Inform the user and ask for confirmation if there has been any defaulting for the target libraries/sublibraries, or if something will be superseded or obsoleted.

Make sure the target libraries/sublibraries are accessible.

Copy now the data using the LIBRARIAN functions RESTORE (fromtape) or COPY (fromdisk).

Perform the final linking - if any requested.

Update the user's system history file (for INSTALL only).

IKRISLNK

Link linkbooks for PTF application.

FUNCTION

Link all linkbooks into their specified target sublibrary(ies).

In case where the IPL-ed system sublib SYSLIB is to be updated, an intermediate sublibrary \$\$MSHP is created in the same library. The links are done into \$\$MSHP, from where the created phases are moved into SYSLIB as soon as all links for SYSLIB were successful.

As soon as all link books of a PTF are successfully linked, the PTF entry in the history file is flagged as completely installed.

The intermediate sublibrary \$\$MSHP is deleted again.

The submitted sets are deleted.

PARAMETERS

LNKINFOS BIN(15)	/* set containing information */
	/* which link books are to be */
	/* linked into which sublibs. */
OUTINFOS BIN(15)	/* set containing information */
	/* which link books belong to */
	/* which PTFs. */
SYSRS CHAR(XLIBLN)	/* name of IPL-ed SYSLIB */

ENTRY-POINT

IKRISLNK

CALLED VIA

```
CALL IKRISLNK(LNKINFOS,OUTINFOS,SYSRS)
           ( In/Out , In/Out , In )
```

INPUT

None

OUTPUT

1. User libraries
2. System history file

Install and list from service tape or backout tape.

FUNCTION

This module handles the service file and the backout tape. It ensures that the input file is assigned as SYS006 and the output tape (if any) as SYS004. It creates always a temporary history on file IJSYS02.

Following three subfunctions are supported:

- List PTFs from service file:

Position the tape to the file which contains the PTF information. If a DISK file is used, the tape related functions are skipped.

Pick up all PTF information and archive them into the auxiliary history file.

Rewind and close the tape, if the service file is on tape.

Invoke function "RETRACE PTF" and "RETRACE APAR" to print the PTF information of all archived PTFs from the auxiliary history file.

- Install PTFs from service file:

If the service file is located on tape, position the service tape to the "EXCLUDE file", pick up all EXCLUDE commands, check them syntactically (using the parser) and semantically and fill the information into the I.R.P.

Open the system history file and copy all its components into the newly created auxiliary history file. If tape is used, position the tape to the file which contains the PTF information. Pick up all PTF informations and archive them into the auxiliary history file, except those which are excluded.

Determine which PTFs are needed for application and remove all others from the auxiliary history file. "Needed" for application are those which were included by the user and those which are required for pre- or co-req reasons.

Perform now the "upgrade" checking, which removes all PTFs which have to be rejected for specific reasons. All remaining PTFs in the auxiliary history file will be applied.

If PARM='PTFLIST' has been specified, member \$MSHP.REQ is cataloged in IJSYSRS.SYSLIB containing COPY-statements for all PTFs selected for application. This member may later on be used as input to utility SRVCPTF (copy PTFs from tape to DASD file).

Merge the history file information of the PTFs into the system history file, but flag them as incorrectly applied.

Pick up now the data from the PTF file.

Close and rewind the tape, if service tape is used.

Update the system history file. All PTFs are now flagged as correctly applied, but not yet linked.

Perform the final linking and update the history file information.

- Install PTFs from backout tape:

Open the backout tape.

Open the system history file and copy all its components into the newly created auxiliary history file.

Pick up all PTF informations and archive them into the auxiliary history file.

Reposition the tape to the start of the backout file. Determine which PTFs are needed for revoking and remove all others from the auxiliary history file.

"Needed" for revoking is the one which was selected by the user, those which have common members with another selected one and those, which needed another selected one as pre- or co-req.
Perform now the "upgrade" checking, which removes all PTFs which have to be rejected for specific reasons. All remaining PTFs in the auxiliary history file will be revoked.
Merge the history file information of those PTFs into the system history file, but flag them as incorrectly applied. Pick up now the data from the backout file.
Close and rewind the tape.
Update the system history file. All PTFs are now flagged as correctly applied, but not yet linked.
Perform the final linking and update the history file information.

PARAMETERS

None

ENTRY-POINT

IKRISVBK

CALLED VIA

CALL IKRISVBK

INPUT

1. System history file
2. Service tape or backout tape

OUTPUT

System history file.

IKRLAPA

FUNCTION

This routine scans all PTF records of a COMPONENT for an APAR- number. If the APAR is found, the respective PTF record is re- turned. If NEXT=NO, the COMPO- NENT is scanned from the beginning. If NEXT=YES, the PTFCELL contains the cell of a PTF containing one occurrence of the APAR. The routine then will scan for the next PTF with another occurrence of the same APAR.

PARAMETERS

```
HFTYPE BIN(15)           /* history file type      */
                          /* 1=perm 2=temp         */
RECTYPE BIN(15)          /* record type           */
                          /* 1=shortterm 2=longterm */
1ARGUMENT                /* search argument       */
2 COMPON CHAR(LENGTH(INSCMP)) /* component and level   */
2 APAR CHAR              /* APAR-number           */
2 NEXT BIT(1)            /* next occurrence       */

1PTFCELL,                /* PTF - CELL           */
2 PTFADR PTR(32),        /* stor. address of PTF record*/
2 PTFRBA PTR(24) BDY(WORD); /* RBA of PTF record     */
```

ENTRY-POINT

IKRLAPA

CALLED VIA

```
CALL IKRLAPA (HFTYPE,RECTYPE,ARGUMENT,PTFCELL)
              ( In , In , In ,In/Out )
```

INPUT

None

OUTPUT

None.

IKRLCMPR

Locate a comprised component.

FUNCTION

Determine which product comprises the specified component. In case where no cell-id was specified, the product queue is searched from the beginning.

If a cell-id was specified, parameter NEXT triggers how the search will be done: For NEXT=NO, the search will be started with the specified product; for NEXT=YES, the search is started for the next product in queue.

As soon as the component is located, the found product cell-id is return to the caller.

If the component cannot be located at all, NULL and UNDEF is returned.

PARAMETERS

HFTYPE BIN(15)	/* history file (1=sys, 2=aux)*/
RECTYPE BIN(15)	/* store type(1=short, 2=long)*/
1 PRODCELL	/* cell-id of product record */
2 PRODADR PTR(32)	/* storage address */
2 PRODRBA PTR(24) BDY(WORD)	/* RBA in history file */
1 ARGUMENT	/* search argument */
2 COMON CHAR(LENGTH(COFCMPID))	/* component-id */
2 LEVEL CHAR(LENGTH(PRDRFF))	/* level */
2 NEXT BIT(1)	/* next occurrence */

ENTRY-POINT

IKRLCMPR

CALLED VIA

```
CALL IKRLCMPR(HFTYPE,RECTYPE,PRODCELL,ARGUMENT)
           ( In , In , In/Out , In )
```

INPUT

The specified history file

OUTPUT

None.

IKRLGNRC

Locate generic members belonging to a component.

FUNCTION

Scan all generated members which belong to the specified component and locate the first one with a name with the same prefix as the submitted generic name. The located member is returned or "UNDEF" and "NULL" if none found.

If SHNEXT=NO, the component is scanned from the beginning.

If SHNEXT=YES, BEGCELL must contain the cell-id of a generated member; forward scanning is started from there for another member matching the generic name.

PARAMETERS

1) HFTYPE BIN(15)	/* type of history file:	*/
	/* 1:=system, 2:=auxiliary	*/
2) STORTYP BIN(15)	/* how to be stored (longterm,	*/
	/* shorterm)	*/
3) 1 BEGCELL	/* cellid for member record	*/
	/* if SHNEXT=1: member cell	*/
	/* if SHNEXT=0: component cell	*/
2 BEGADR PTR(32)	/* storage address	*/
2 BEGRBA PTR(24) BDY(BYTE)	/* RBA	*/
2 BEGADR PTR(32)	/* storage address of generated	*/
	/* member record or "NULL"	*/
2 BEGRBA PTR(24) BDY(BYTE)	/* RBA of generated member	*/
	/* record or "UNDEF"	*/
4) 1 SHARG	/* search argument	*/
2 SHCOMP CHAR(12)	/* component id	*/
2 SHMEMB CHAR(10)	/* generated member	*/
2 SHNEXT BIT(1)	/* 1:=next occurrence	*/

NOTE: IF SHNEXT=0 AND SHCOMP=' ' PARAMETER BEGCELL IS
NEEDED FOR INPUT.

ENTRY-POINT

IKRLGNRC

CALLED VIA

IKRLGNRC(HFTYPE,STORTYP,BEGCELL,SHARG)
(In ,In ,In/Out ,In)

INPUT

History file

OUTPUT

None.

IKRLIBST

LIBRARIAN interface module

FUNCTION

Uses LIBRARIAN second level interfaces in order to provide several functions for accessing sublibraries, members and records:

```
IKRLIBCA /* build ACCESS chain for component */
IKRLIBBA /* build requested ACCESS chain */
IKRLIBBL /* build complete LIBDEF as requested */
IKRLIBBC /* build LIBDEF OBJ,CATALOG as requested */
IKRLIBBS /* build LIBDEF OBJ,SEARCH as requested */
IKRLIBBM /* build CONNECT information */
IKRLIBGA /* get current ACCESS chain */
IKRLIBTA /* determine not accessible sublibraries */
IKRLIBCM /* connect member */
IKRLIBDM /* disconnect member */
IKRLIBUS /* update stow table */
IKRLIBGR /* get logical record */
IKRLIBPR /* put logical record */
```

PARAMETERS

1COMPKEY	/* component key	*/
2 KEYTYPE BIN(8)	/* type of component key:	*/
	/* 1:=name, 2:=cell	*/
2 COMPKEYD CHAR(¢)	/* component key info	*/
1NAMETYPE DEFINED(COMPKEYD)	/* component key is name:	*/
2 COMPID CHAR(9)	/* component id	*/
2 COMPLVL CHAR(3)	/* component level	*/
1CELLTYPE DEFINED(COMPKEYD)	/* component key is cell:	*/
2 COMPADR PTR(32) BDY(BYTE)	/* address of component record	*/
2 COMPRBA PTR(24) BDY(BYTE)	/* RBA in history file	*/
1FROM	/* source information	*/
2 FROMLIB CHAR(LL)	/* library name	*/
2 FROMSL CHAR(SLL)	/* sublib name	*/
1MEMBKEY	/* member key	*/
2 MEMBNM CHAR(MNL)	/* member name	*/
2 MEMBTY CHAR(MTL)	/* member type	*/
ACCSET BIN(15)	/* set of access info	*/
NOTACC BIN(15)	/* set of sublibs which	*/
	/* could not be accessed	*/
SYSRS CHAR(LL+SLL)	/* name of IPL-ed system sublib	*/
READBUF CHAR(¢)	/* data to be read	*/
LRBA PTR(32)	/* byte address (hex)	*/
	/* relative to member start	*/
MOVELEN BIN(15)	/* # of requested bytes	*/
MOVED BIN(31)	/* # of bytes actually read	*/
WRITEBUF CHAR(¢)	/* data to be written	*/
MBFOUND BIT(1)	/* 1:=member found	*/
ACCESSED BIN(15)	/* # of accessed sublibs	*/
MSGREQ BIT(1)	/* 1:=PRINT "SUBLIB NOT DEFINED"	*/

ENTRY-POINT

IKRLIBCA /* build ACCESS chain for component */

FUNCTION

Pre: RESIDENCE information for given component available.

Post: ACCESS chain for LIBRARIAN commands built containing those sublibraries in which the submitted component resides.

CALLED VIA

CALL IKRLIBCA(COMPKEY,ACCESSED)
(In , Out)

INPUT

System history file.

OUTPUT

ACCESS chain for LIBRARIAN established.

ENTRY-POINT

IKRLIBBA /* build requested ACCESS chain */

FUNCTION

Post: ACCESS chain for LIBRARIAN commands built containing submitted sublibraries.

CALLED VIA

CALL IKRLIBBA(ACCSET)
(In)

INPUT

None

OUTPUT

ACCESS chain for LIBRARIAN established.

ENTRY-POINT

IKRLIBBL /* build complete LIBDEF as requested */

FUNCTION

Post: A CATALOG chain for LNKEDT is built containing the submitted sublibraries.
Also a temporary SEARCH chain for OBJ is built, which is searched by
LNKEDT before any existing chain.

CALLED VIA

CALL IKRLIBBL(ACCSET)
(In)

INPUT

None

OUTPUT

LIBDEF OBJ,SEARCH=... for LNKEDT is established
LIBDEF PHASE,CATALOG=... for LNKEDT is established

ENTRY-POINT

IKRLIBBC /* build LIBDEF OBJ,CATALOG as requested */

FUNCTION

Post: A CATALOG chain for LNKEDT is built containing the submitted sublibraries.

CALLED VIA

```
CALL IKRLIBBC(ACCSET)
      ( In )
```

INPUT

None

OUTPUT

LIBDEF PHASE,CATALOG=... for LNKEDT is established

ENTRY-POINT

```
IKRLIBBS /* build LIBDEF OBJ,SEARCH as requested */
```

FUNCTION

Post: A SEARCH chain for LNKEDT is built containing the submitted sublibraries, which will be searched (not by fetch) before any existing chain.

CALLED VIA

```
CALL IKRLIBBS(ACCSET)
      ( In )
```

INPUT

None

OUTPUT

LIBDEF OBJ,SEARCH=... for LNKEDT is established

ENTRY-POINT

```
IKRLIBBM /* build CONNECT information */
```

FUNCTION

Post: CONNECT information is built according to the submitted source sublibrary and target sublibrary(ies) for any following MERGE or COPY command.

CALLED VIA

```
CALL IKRLIBBM(FROM,ACCSET)
      ( In , In )
```

INPUT

None

OUTPUT

CONNECT information is established

ENTRY-POINT

IKRLIBGA /* get current ACCESS chain */

FUNCTION

Pre: ACCESS chain has been established before.

Post: The current ACCESS chain information is returned to the caller.

CALLED VIA

CALL IKRLIBGA(ACCSET)
(Out)

INPUT

ACCESS chain

OUTPUT

None

ENTRY-POINT

IKRLIBTA /* determine not accessible sublibraries */

FUNCTION

Post: Try to access all requested sublibraries. Those which cannot be accessed (label info missing, library no online, sublibrary not defined) are returned to the caller. In case where the IPL-ed system sublib was among the accessed sublibs, its name is returned.

If messages printing is requested, the user is informed why it failed.

CALLED VIA

CALL IKRLIBTA(ACCSET,NOTACC,SYRSR,MSGREQ)
(In , Out , Out , In)

INPUT

None

OUTPUT

None

ENTRY-POINT

IKRLIBCM /* connect member */

FUNCTION

Pre: ACCESS chain has been established before.

Post: Member is searched following the chain of accessed sublibraries. If found, the member is connected, FOUND condition is returned and the names of the library and sublibrary where the member was found is returned in FROM; otherwise NOT FOUND is indicated.

CALLED VIA

```
CALL IKRLIBCM(MEMBKEY,MBFOUND,FROM)
           ( In , Out , Out)
```

INPUT

ACCESS chain

OUTPUT

AMB and RPL for the requested member built

ENTRY-POINT

IKRLIBDM /* disconnect member */

FUNCTION

Pre: A member has been connected before.

Post: The member is disconnected.

CALLED VIA

```
CALL IKRLIBDM
```

INPUT

AMB and RPL of the member

OUTPUT

None

ENTRY-POINT

IKRLIBGR /* get logical record */

FUNCTION

Pre: A member is connected.

Post: The requested number of bytes, starting at the submitted relative (to member begin) byte address is moved into the provided buffer. If LRBA+MOVELEN-1 exceeds the range of the connected member, only bytes within its range are moved. The number of bytes moved is always returned.

CALLED VIA

```
CALL IKRLIBGR(LRBA,MOVELEN,MOVED,READBUF)
           ( In , In , Out , Out )
```

INPUT

AMB and RPL of the member

The sublibrary member

OUTPUT

None

ENTRY-POINT

IKRLIBPR /* put logical record */

FUNCTION

Pre: A member is connected and (LRBA+MOVELEN-1) is still in its range.

Post: In the connected member, starting at the submitted relative (to member begin) byte address, the requested number of bytes are replaced by those provided in the buffer area.

CALLED VIA

CALL IKRLIBPR(LRBA,MOVELEN,WRITEBUF)
(In , In , In)

INPUT

AMB and RPL of the member

OUTPUT

The sublibrary member.

IKRLINCM

Locate macro in generated members.

FUNCTION

Scan all generated members which belong to the specified component and locate the first one which has the specified macro included.

If such a member is found, its cell-id is returned to the caller. Otherwise NULL and UNDEF is returned.

Parameter SHNEXT triggers where searching is to be started:

If SHNEXT=NO, the component is scanned from the beginning;

For SHNEXT=YES, the cell-id of a generated member is expected at entry time and the search is started for the next member in queue.

PARAMETERS

HFTYPE BIN(15)	/* history file (1=sys, 2=aux)*/
RECTYPE BIN(15)	/* store type(1=short, 2=long)*/
1CELLID	/* cell-id of search record */
2 ADDRESS PTR(32)	/* storage address */
2 RBA PTR(24) BDY(WORD)	/* RBA in history file */
1SHARG	/* search argument */
2 SHCOMP CHAR(LENGTH(INSCMP))	/* component-id */
2 SHMAC CHAR(LENGTH(IMTMNM)+2)	/* macro name */
2 SHNEXT BIT(1)	/* next occurrence */

ENTRY-POINT

IKRLINCM

CALLED VIA

CALL IKRLINCM(HFTYPE,RECTYPE,CELLID,SHARG)
(In , In ,In/Out, In)

INPUT

The specified history file

OUTPUT

None.

IKRLKUP

Look up the permanent history file for a specified entry.

FUNCTION

This routine scans the permanent history file for a specified entry and lists all relevant information on SYSLOG.

PARAMETERS

None

ENTRY-POINT

IKRLKUP

CALLED VIA

CALL IKRLKUP

INPUT

IKRXLKP (POINTED BY CRGMATRP)

OUTPUT

For LOOKUP PTF = PTF-NUMBER
For LOOKUP APAR = APAR-NUMBER
For LOOKUP PRODUCT = PRODUCT-NUMBER
For LOOKUP MODULE/MACRO/PHASE = NAME
For LOOKUP LATEST SERVICE.

IKRLMLF

Locate a module in a local APAR/fix.

FUNCTION

This routine scans the module-in-APAR records of all local APAR-fixes for a requested module/macro/phase and returns the APAR/fix-record in which the module was found.

If NEXT=NO, the component is scanned from the beginning.

If NEXT=YES, the APACELL contains the cell of a PTF containing one occurrence of the module. The routine then will scan for the next APAR with another occurrence of the same module.

PARAMETERS

HFTYPE BIN(15)	/* history file type	*/
	/* 1=perm 2=temp	*/
RECTYPE BIN(15)	/* record type	*/
	/* 1=shorterm 2=longterm	*/
1ARGUMENT	/* search argument	*/
2 COMPON CHAR	/* component and level	*/
2 MODULE CHAR	/* module-id	*/
2 NEXT BIT(1)	/* next occurrence	*/
1APACELL	/* APAR - CELL	*/
2 APAADR PTR(32)	/* stor.address of APAR record*/	
2 APARBA PTR(24)	/* RBA of APAR record	*/
1MIACELL	/* Module in APAR record	*/
2 MIAADR PTR(32)	/* Stor.address of module rec.*/	
2 MIARBA PTR(24)	/* RBA of module record	*/

ENTRY-POINT

IKRLMLF

CALLED VIA

CALL IKRLMLF (HFTYPE,RECTYPE,APACELL,MIACELL,ARGUMENT)
(In , In , Out , Out , In)

INPUT

None

OUTPUT

None.

IKRLMOD

Locate an MODULE-in-PTF and return the respective PTF record.

FUNCTION

This routine scans all PTF-RECORDS of a component for a MODULE-name. If the MODULE is found, the respective PTF record is returned.

If NEXT=NO the component is scanned from the beginning.

If NEXT=YES PTFCELL contains the cell of a PTF containing one occurrence of the MODULE. The routine then will scan for the next PTF with another occurrence of the same MODULE.

PARAMETERS

HFTYPE BIN(15)	/* history file type	*/
	/* 1=perm, 2=temp	*/
RECTYPE BIN(15)	/* record type	*/
	/* 1=shorterm, 2=longterm	*/
1PTFCELL	/* PTF - cell	*/
2 PTFADR PTR(32)	/* stor.address of PTF-record	*/
2 PTFRBA PTR(24)	/* RBA of PTF-record	*/
1ARGUMENT	/* search argument	*/
2 COMPON CHAR(LENGTH(INSCMP))	/* component and level	*/
2 MODULE CHAR	/* module-name	*/
2 NEXT BIT(1)	/* next occurrence	*/

ENTRY-POINT

IKRLMOD

CALLED VIA

CALL IKRLMOD (HFTYPE,RECTYPE,PTFCELL,ARGUMENT)
(in , in , out , in)

INPUT

None

OUTPUT

None.

IKRLOAD

Load a system control program.

FUNCTION

This module loads the requested system control program into the SCP-area of the MSHP partition, opens the logical interface with the program via the interface control table and transfers control to it.

The return code given back by the loaded system control program is transferred to the caller.

For more information see under entry points:

```
IKRLOAD          /* load the system control program      */
IKRLOAEJ         /* receive control from system controlprogram */
```

PARAMETERS

```
PROGNAME CHAR(8)          /* program to be loaded      */
SENDROUT POINTER(32)     /* address of caller's send  */
                          /* appendage routine. NULL  */
                          /* indicates there is none  */
RECVROUT POINTER(32)     /* address of caller's receive */
                          /* appendage routine. NULL  */
                          /* indicates there is none  */
RETCODE BIN(15)          /* return code                */
```

ENTRY-POINT

```
IKRLOAD /* load the system control program */
```

FUNCTION

Prepare the interface table, open the logical interface, load the specified system control program, and transfer control to the program.

CALLED VIA

```
CALL IKRLOAD(PROGNAME,SENDROUT,RECVROUT,RETCODE)
          ( In , In , In , Out )
```

INPUT

None

OUTPUT

None

ENTRY-POINT

```
IKRLOAEJ /* receive control from system controlprogram */
```

FUNCTION

Entry point after MSHEOJ macro processing of the called system control program.

CALLED VIA

```
CALL IKRLOAEJ(EOJRTCOD)
           (   In   )
```

INPUT

None

OUTPUT

None.

IKRLOFI

Edit the local fix information in a module-in-APAR record.

FUNCTION

This routine reads the local fix description from the module-in-APAR record and its associated FIX-COMMENT records and returns the local fix information one record at a time.

When invoked the first time, input parameter MIACELL must point to a module-in-APAR record read by the calling routine. The routine fills an input buffer, decodes the type of the first local fix record and returns. On second invocation, the second local fix record is returned. This is repeated until the last record has been returned. If no more records are found, the length field in the return parameter FIXREC contains '00'X.

If the input parameter MIACELL changes between two invocations, a new module-in-APAR record is expected. The first local fix record of the new APAR is returned.

Notes:

1. The calling routine must have read the respective modules-in-APAR record.
2. If the module-in-APAR record was read 'SHORTTERM', it might be not available anymore to the calling routine. For second invocations, the record need not be reread by the calling routine.

Local fix information

Type	Length	Data
------	--------	------

Type bin(8)

1 = Delete (macro)
2 = Replace (macro)
3 = Insert (macro)
4 = Data (macro)
5 = Address (phase, module) (alter)
6 = Verify (phase, module) (old text)
7 = Replace (phase, module) (new text)
8 = Estdid - id
9 = Estdid increment
B = Restart (macro)

Length bin(8)

Data char(*) max.length = 80

PARAMETERS

1MIACELL		/* cell of module-in-APAR rec.*/
2 MIAADR	PTR(32)	/* stor.address of module rec.*/
2 MIARBA	PTR(24)	/* RBA of module record */
1FIXREC		/* local fix record */
2 FIXLEN	BIT(8)	/* length of fix record */
2 FIXTYPE	CHAR(8)	/* decoded type */
2 FIXDATA	CHAR(80)	/* fix data */

ENTRY-POINT

IKRLOFI

CALLED VIA

CALL IKRLOFI (MIACELL, FIXREC)
(In , Out)

INPUT

None

OUTPUT

None.

IKRLSUP

Locate an superseding PTF and return the respective PTF record.

FUNCTION

This routine scans all PTF records within a specified component and their SUPERSEDED-PTF records to locate a superseding PTF. This routine scans all PTF records of a component for an APAR- number. If the APAR is found, the respective PTF-RECORD is returned.

If NEXT=NO, the component is scanned from the beginning.

If NEXT=YES, the PTFCELL contains the cell of a PTF containing one occurrence of the superseding PTF.

The routine then will scan for the next occurrence of a superseding PTF.

PARAMETERS

HFTYPE BIN(15)	/* history file type	*/
	/* 1=perm, 2=temp	*/
RECTYPE BIN(15)	/* record type	*/
	/* 1=shorterm, 2=longterm	*/
1PTFCELL	/* cell of component	*/
2 PTFADR PTR(32)	/* stor.address of PTF record	*/
2 PTFRBA PTR(24)	/* RBA of PTF record	*/
1ARGUMENT	/* search argument	*/
2 COMPON CHAR(LENGTH(INSCMP))	/* component and level	*/
2 SUPPTF CHAR(LENGTH(PTFNO))	/* PTF-# of superseded PTF	*/
2 NEXT BIT(1)	/* next occurrence	*/

ENTRY-POINT

IKRLSUP

CALLED VIA

```
CALL IKRLSUP (HFTYPE,RECTYPE,PTFCELL,ARGUMENT)
              ( In , In , Out , In )
```

INPUT

None

OUTPUT

None.

IKRMAIN

MSHP main entry routine

FUNCTION

This routine initially gets control from job control for any execution of MSHP.

It initializes the MSHP communication region IKRCOMRG, calculates the storage layout, opens the common files and the work buffer. It ensures that MSHP is not called in parallel from any other partition or any other processor.

It gets and clears (to x'00') a work buffer for the internal representation blocks (IRP) for all requested functions. It reads a function control statement and transfers control to the corresponding semantic check routine, which reads all associated detail control statements.

After the function is executed, control is given back to this module which handles the next function - if any - or returns control back to VSE.

PARAMETERS

None

ENTRY-POINT

IKRMAIN

CALLED VIA

(//) EXEC MSHP

INPUT

None

OUTPUT

None.

IKRMERGE

Main routine of the merge in MSHP

FUNCTION

Perform OPEN/CLOSE for both history files to be merged. The function routine is called to do the job.

PARAMETERS

None

ENTRY-POINT

IKRMERGE

CALLED VIA

CALL IKRMERGE

INPUT

None

OUTPUT

None.

IKRMOD

Set a history file buffer as 'to be rewritten'.

FUNCTION

Set the 'to be rewritten' bit in the buffer table for the respective buffer; this will force IKRREWR or IKRGTBUF to rewrite the history file block

PARAMETERS

BUFFADDR POINTER(32) address of the altered record

ENTRY-POINT

IKRMOD

CALLED VIA

CALL IKRMOD (BUFFADDR)

INPUT

None

OUTPUT

None.

IKRMRGIN

Internal merge routine

FUNCTION

Perform the internal merge from one history file into another. There are two types of actions: test only, or merge. Before starting to merge, test the necessary space in the target history file. If sufficient space is found, do the merge. Merging is done in two cycles: one for component records and dependent other records, and another for product records. A selective merge can be controlled by a list of components.

PARAMETERS

None

ENTRY-POINT

IKRMRGIN

CALLED VIA

CALL IKRMRGIN

INPUT

IKRXMRG - IRP for MERGE command (also used internally for BACKUP PRODUCT)
Source and target history files

OUTPUT

Updated target history file.

IKRNOTE

Calculate the 'NOTE' value of a history file block.

FUNCTION

Using the device dependent values located in the MSHP COMREG, compute the 'NOTE' value of a history file record as known by SDMODW.

PARAMETERS

1)	HFTYPE	BIN(15)	/* history file type	*/
	(INPUT)		/* 1 = permanent	*/
			/* 2 = temporary	*/
2)	BLOCKNBR	BIN(15)	/* history file block number	*/
	(INPUT)		/* relative to 1	*/
3)	NOTE	CHAR(4)	/* note value of that block	*/
	(OUTPUT)			
	A. FBA:			
		POINTER(24)	/* C.I. number	*/
		BIN(8)	/* record number	*/
	B. CKD:			
		BIN(16)	/* cylinder number	*/
		BIN(8)	/* track number	*/
		BIN(8)	/* record number	*/

ENTRY-POINT

IKRNOTE

CALLED VIA

CALL IKRNOTE (HFTYPE,BLOCKNBR,NOTE)

INPUT

None

OUTPUT

None.

IKROPBUF

Open the MSHP buffer pool.

FUNCTION

Create the buffer table in the buffer 0.

ENTRY-POINT

IKROPBUF

CALLED VIA

CALL IKROPBUF

INPUT

None

OUTPUT

None.

IKROPEN

Open up to 12 specified files.

FUNCTION

Select the DTF address of each specified file and open them.

PARAMETERS

FILE1, FILE2, FILE3, FILE4, FILE5, FILE6, FILE7,
FILE8, FILE9, FILEA, FILEB, FILEC, FILED, FILEE

(FROM 1 TO 14 FILE INDICES OF THE FORM:

IJSYSHF CONSTANT(1) BIN(15)
IJSYS02 CONSTANT(2) BIN(15)
TAPEHF CONSTANT(3) BIN(15)
CONSOLE CONSTANT(4) BIN(15)
IJSYSIN CONSTANT(5) BIN(15)
IJSYSLS CONSTANT(6) BIN(15)
IJSYSPH CONSTANT(7) BIN(15)
IJSYS04 CONSTANT(8) BIN(15)
IJSYS05 CONSTANT(9) BIN(15)
IJSYSXX CONSTANT(10) BIN(15)
IJSYSLK CONSTANT(11) BIN(15)
IJSYS01 CONSTANT(12) BIN(15)
IJSYS03 CONSTANT(13) BIN(15)
CUMFILE CONSTANT(14) BIN(15)
BKOUTAP CONSTANT(14) BIN(15))

ENTRY-POINT

IKROPEN

CALLED VIA

CALL IKROPEN (FILE1, FILE2, FILE3, FILE4, FILE5, FILE6, FILE7,
FILE8, FILE9, FILEA, FILEB, FILEC, FILED, FILEE)

INPUT

None

OUTPUT

None.

IKROPHF

Open a history file (permanent or temporary).

FUNCTION

Check the respective assignment, fill the history file table in the communication region, Select the correct DTFSD and OPEN it. If the opened history file is in old format, emulate it to new format.

PARAMETERS

HFTYPE BIN(15) /* History file type */

ENTRY-POINT

IKROPHF

CALLED VIA

CALL IKROPHF(HFTYPE)
(In)

INPUT

DTF of history file as denoted in IKRCOMRG

OUTPUT

History file table in IKRCOMRG.

IKROPLNK

Open the link file (MSHPLNK).

FUNCTION

This routine "opens" the link file and prepares it for being used by MSHP. The status of the link file triggers how this "opening" will be done. The status may be:

- not opened at all by MSHP,
- has been opened before, but is now closed,
- has been opened before and is still open.

If the link file has not been opened before by MSHP, a check is made that IJSYS01 as well as IJSYSLN are assigned and labels are available. A label with the file name MSHPLNK is created in the label area with the same DLBL and EXTENT information as the IJSYSLN file. The DTF skeleton is saved, then an open is performed.

If the link file has been opened before and is now closed, the DTF is re-freshed using the previously saved skeleton and an open is performed.

If the link file has been opened before and is still open, no open is performed, but the DTF is re-freshed with the status after the last open.

Positioning on the link file is triggered by two different entry points:

```
IKROPLNK /* open and position to start of link file */  
IKROPLNC /* open and keep position */
```

PARAMETERS

None

ENTRY-POINT

```
IKROPLNK /* open and position to start of link file */
```

FUNCTION

Open the link file as described above, and force positioning to the start of the link file by clearing the SYSLNK DIB. Write an ACTION card onto the file.

CALLED VIA

```
CALL IKROPLNK
```

INPUT

None

OUTPUT

1. Entry in the partition label area for IJSYSLK
2. SYSLNK DIB
3. Link file IJSYSLN

ENTRY-POINT

IKROPLNC /* open and keep position */

FUNCTION

Open the link file as described above and keep the current positioning.

CALLED VIA

CALL IKROPLNC

INPUT

None

OUTPUT

Entry in the partition label area for IJSYSLK.

IKROPMSH

MSHOPEN interface

FUNCTION

Sets a pseudo LIOCS module address in the corresponding DTF and sets the IOREG for 2IOAREAS OUTPUT DTFs.

PARAMETERS

DTF(L'DTFCP OR DTFDI), DTFCP, DTFDI

ENTRY-POINT

IKROPMSH

CALLED VIA

CALL IKROPMSH (DTF)

INPUT

None

OUTPUT

DTF LIOCS module address and IOREG.

IKRPAPAL

Print APAR list.

FUNCTION

Controls the printing the lines for 'RETRACE APARS'. The information is retrieved from the sort table. For each sort record the actual function is performed by a call to IKRPAPAL. This routine edits and writes out the information about the APAR, for whose sort record it is called.

For an APAR resolved by a PTF, one line is printed and contains: the APAR number, the affected component, the resolving PTF, and the date when the PTF was applied.

For a local/APAR fix the first line contains: The APAR number, the affected component, the word "LOCAL", and the date when the correction was made; this may be followed by further lines showing an affected module and what the correction for this module consisted of.

PARAMETERS

1 SORTVAL,	
2 ASORTST PTR(32),	/* SORT AREA START ADDRESS */
2 ARDTBLST PTR(32),	/* RECORD TABLE START ADDRESS */
2 ASQTBLST PTR(32),	/* SEQ. TABLE START ADDRESS */
2 ASORTND PTR(32),	/* SORT AREA END ADDRESS */
2 ARDTBLND PTR(32),	/* RECORD TABLE END ADDRESS */
2 ASQTBLND PTR(32),	/* SEQUENCE TABLE END ADDRESS */
2 INSRDN FIXED(15),	/* INSERTED RECORD NUMBER */
2 NINSRDN FIXED(15),	/* NOT INSERTED RECORD NUMBER */
2 LSRTRD FIXED(8),	/* LENGTH OF SORT RECORD */
2 LSF FIXED(8),	/* LENGTH OF SORT RECORD */
2 FSTCMPRB PTR(24),	/* FIRST COMPONENT RBA */
2 FSTFTRRB PTR(24),	/* FIRST FEATURE RBA */
2 NXCMPRB PTR(24);	/* NEXT COMPONENT RBA */

ENTRY-POINT

IKRPAPAL

CALLED VIA

CALL IKRPAPAL(SORTVAL)
(In/Out)

INPUT

Sort record table and sort sequence table

OUTPUT

None.

IKRPARSE

Parser interface routine

FUNCTION

This routine initializes the parser control block (CPCB) and calls the parser routine. After return, it analyses the parser return code and initiates the following actions:

1. Provides a continuation line.
2. Recognizes whether input line contains more than one command to be parsed.
3. Reads the next line in case of an empty line or a comment read.
4. Prints the input lines.
5. Stacks the input lines in case of functional control statements.

PARAMETERS

None

INPUT

MSHP statements

OUTPUT

Parser control block containing the correctly parsed input string.

IKRPCREQ

Queue pre/co-requisite and conflicting information records.

FUNCTION

Pick up all pre/co-requisite and conflicting informations, put them into pre-req records, and queue those records to the origin specified in the parameter list.

PARAMETERS

1OWNCELL	/* OWNER OF PRE-REQ INFORMATION */
2 OWNADR PTR	/* STORAGE ADDRESS OF OWNER */
2 OWNRBA PTR(24)	/* HISTORY FILE RBA OF OWNER */

ENTRY-POINT

IKRPCREQ

CALLED VIA

CALL IKRPCREQ(OWNCELL)
(In)

INPUT

IKRXENT - Interface to semantic checker

OUTPUT

None.

IKRPDZAP

Correct core image library phases.

FUNCTION

Alters an existing phase or reverts such an alteration. The area which corresponds to the change address is read, the verification is checked and the altered area is rewritten in place.

ASCAN function searches for a given argument and displays the area at the resulting location within the phase if found.

PARAMETERS

AFFPTR	PTR		/* pointer to IRP of affected phase */
1CNTLFLG	BIT(16)		/* control flags */
2 DOZAP	BIT(1)		/* 1:= correct phase */
2 UNDOZAP	BIT(1)		/* 1:= UNDO the change in phase */
2 OK	BIT(1)		/* 1:= ZAP was successful */
2 NEXTAFF	BIT(1)		/* 1:=next affects already read */
2 *	BIT(1)		
2 CATASTRO	BIT(1)		/* error condition */
2 *	BIT(10)		/* reserved */

/* FOLLOWING FOUR PARAMETERS ARE NEEDED FOR INVOCATION OF */
/* IKRCKLV2 AND IKRHDLV2. THEY ARE NOT TOUCHED IN THIS MODULE */

PARM2	BIT(8)		/* "OPENFL" in IKRECOR */
PARM3	BIT(16)		/* "FFOUND" in IKRECOR */
1WRKBUFTB			/* table of work buffers */
2 *	BIN(8)		
2 BFADRACT	BIN(8)		
2 BFADRTBL(*)	PTR		
1PARM5			/* "FREE" in IKRECOR */
2 *	PTR		
2 *	BIN		

ENTRY-POINT

IKRPDZAP

CALLED VIA

CALL IKRPDZAP(AFFPTR,CNTLFLG,PARM2,PARM3,WRKBUFTB,PARM5)
(In ,In/Out ,In ,In ,In ,In)

INPUT

IKRXENT - IRP for function CORRECT/UNDO
IKR#AFF - SIT for AFFECTS
IKR#ALT - SIT for ALTER
IKR#SCN - SIT for SCAN

OUTPUT

IKRXENT- IRP for function CORRECT/UNDO.

IKRPERS

Personalize permanent history file.

FUNCTION

Fill the header records of a permanent history file with user related personal data (customer name, address, phone number, system programmer name, environment).

PARAMETERS

None

ENTRY-POINT

IKRPERS

CALLED VIA

CALL IKRPERS

INPUT

1. IKRXPERS (pointed by CRGMATRP)
2. Permanent history file

OUTPUT

Modified permanent history file.

IKRPGENL

Print generated member list.

FUNCTION

Controls producing the list of generated members requested by 'RETRACE COMPONENT'.

IKRBGENL is called for each generated member. A generated member description produced by IKRBGENL generally consists of several print lines, in which the following information is arranged in columns:

GENERATED MEMBER|COMPONENT|DATE|STATUS|INCLUDED MACROS

The "STATUS" field shows if a generated member is "back level".

PARAMETERS

1 SORTVAL			
2 ASORTST	PTR(32)	/* sort area start address	*/
2 ARDTBLST	PTR(32)	/* record table start address	*/
2 ASQTLST	PTR(32)	/* sequence table start address	*/
2 ASORTND	PTR(32)	/* sort area end address	*/
2 ARDTBLND	PTR(32)	/* record table end address	*/
2 ASQTLND	PTR(32)	/* sequence table end address	*/
2 INSRDN	FIXED(15)	/* inserted record number	*/
2 NINSRDN	FIXED(15)	/* not inserted record number	*/
2 LSRTRD	FIXED(8)	/* length of sort record	*/
2 LSF	FIXED(8)	/* length of sort files	*/
2 FSTCMRBB	PTR(24)	/* first component RBA	*/
2 FSTFTRRB	PTR(24)	/* first feature RBA	*/

ENTRY-POINT

IKRPGENL

CALLED VIA

CALL IKRPGENL(SORTVAL)
(In/Out)

INPUT

Sort record table and sort sequence table.

OUTPUT

None.

IKRPHMSH

MSHPHASE interface

FUNCTION

Fills the interface table with the phase information and calls the RECEIVE Appendage.

PARAMETERS

PHASNAME CHAR(8) /* phase name being cataloged */

ENTRY-POINT

IKRPHMSH

CALLED VIA

CALL IKRPHMSH (PHASNAME)
(In)

INPUT

None

OUTPUT

None.

IKRPIO

MSHP parser I/O routine

FUNCTION

This routine performs the parser requested I/O during prompting and during error correction.

PARAMETERS

None

ENTRY-POINT

IKRPIO

CALLED VIA

CALL IKRPIO
ACCESSED VIA V-CONSTANT.

INPUT

None

OUTPUT

None.

IKRPPTFL

Print PTF list.

FUNCTION

Controls producing the list of PTFS requested by 'RETRACE PTF'. For each PTF, IKRBPTFL is called. A PTF description produced by IKRBPTFL generally consists of several print lines, in which the following information is arranged in columns:

PTF# | STATUS | COMPONENT | APAR# | AFFECTS | DATE |

REQUIRES | SUPERSEDES | SUPERSEDED |

The "STATUS" field shows if a PTF is incorrectly applied, or has been revoked.

"AFFECTS" means: modules affected.

"REQUIRES" lists COMPONENTS, FEATURES, and PTFS that are pre-required, co-required, or are required not to be applied.

PARAMETERS

ISORTVAL,

2 ASORTST PTR(32),	/* sort area start address */
2 ARDTBLST PTR(32),	/* record table start address */
2 ASQTBLST PTR(32),	/* seq. table start address */
2 ASORTND PTR(32),	/* sort area end address */
2 ARDTBLND PTR(32),	/* record table end address */
2 ASQTBLND PTR(32),	/* sequence table end address */
2 INSRDN FIXED(15),	/* inserted record number */
2 NINSRDN FIXED(15),	/* not inserted record number */
2 LSRTRD FIXED(8),	/* length of sort record */
2 LSF FIXED(8),	/* length of sort record */
2 FSTCMRFB PTR(24),	/* first component RBA */
2 FSTFTRRB PTR(24);	/* first feature RBA */

ENTRY-POINT

IKRPPTFL

CALLED VIA

CALL IKRPPTFL(SORTVAL)
(In/Out)

INPUT

Sort record table and sort sequence table.

OUTPUT

None.

IKRPRAPL

Prepare for PTF application.

FUNCTION

Prepare PTF's jobs from the service tape , reads them onto a workfile (SYS003) and analyze certain statements inside of a PTF .

The selection is triggered by information specified in the internal representation block (I.R.P.), by which specific TAILOR jobs can be included or excluded.

PARAMETERS

None

ENTRY-POINT

IKRPRAPL

CALLED VIA

CALL IKRPRAPL

INPUT

1) IKRXCUM - internal representation block (I.R.P.)

OUTPUT

1) IKRXCUM - internal representation block (I.R.P.)

IKRPRCS

Print command stack.

FUNCTION

Print function control statement stack. This routine is used to stack the lines of a function control statement. The function control statement is filled up from IKRPARSE. If printing should be suppressed, the stack is cleared.

PARAMETERS

None

ENTRY-POINT

IKRPRCS

CALLED VIA

CALL IKRPRCS

INPUT

None

OUTPUT

None.

IKRPSDMD

Pseudo I/O module

FUNCTION

Takes the place of an I/O module in a system program called by MSHP. It will transfer control to the corresponding MSHP routine, either to IKRGTMSH for a GET interface or to IKRPTMSH for a PUT interface.

PARAMETERS

None

ENTRY-POINT

IKRPSDMD + 8 for GET IKRPSDMD +12 for PUT

CALLED VIA

CALL IKRPSDMD

INPUT

REG 1 contains the DTF address

OUTPUT

REG 1 contains parameter list address.

IKRPTMSH

LIOCS PUT interface

FUNCTION

Fills the interface table with the output record address and calls the RECEIVE appendage.

PARAMETERS

DTF CHAR(L'DTFDI OR DTFCP), DTFDI, DTFCP

ENTRY-POINT

IKRPTMSH
IKRPTMEN AFTER RECEIVE APPENDAGE

CALLED VIA

CALL IKRPTMSH (DTF)

INPUT

None

OUTPUT

None.

IKRQUEUE

Queue a history file record to the corresponding chain.

FUNCTION

A given history file record is to be queued to another one. This is controlled by two record type tables, where an allowed attachment is represented by a pair of values at the same table position. A third table contains the displacement of the anchor entry in the basing record.

For extendable record types, the handling of the extension record is done automatically.

PARAMETERS

1NEWCELL	/* cell-id of the record to be queued	*/
	/* it must be already formatted	*/
2 NEWADD POINTER(32)	/* its address	*/
2 NEWRBA POINTER(24)	/* its RBA	*/
1BASECELL	/* cell-id of the base record to which	*/
	/* the new record must be queued	*/
2 BASEADD POINTER(32)	/* its address	*/
2 BASERBA POINTER(24)	/* its RBA	*/

ENTRY-POINT

IKRQUEUE

CALLED VIA

```
CALL IKRQUEUE(NEWCELL,BASECELL)
              (In      ,In      )
```

INPUT

None

OUTPUT

None.

IKRRACSV

RSERV receive appendage for the function 'CORRECT'.

FUNCTION

Put the punch output to the appropriate file.

PARAMETERS

None

ENTRY-POINT

IKRRACSV

CALLED VIA

CALL IKRRACSV

INPUT

Data given by a RSERV program via IKRRCV.

OUTPUT

End-card pointed to by CRGRECVP.

IKRRAESV

ESERV/ASSEMBLY receive appendage for the function 'CORRECT'.

FUNCTION

Put the de/re-edited version of macro on SYS004.

PARAMETERS

None

ENTRY-POINT

IKRRAESV

CALLED VIA

CALL IKRRAESV

INPUT

None

OUTPUT

CRGAPOK Bit in IKRCOMR (indicate error)

SYS004 De/re-edited macro.

IKRRALNK

Linkage editor receive appendage routine

FUNCTION

Receive the names of the phases sent by the linkage editor and indicate in the MSHP communication region, if MSHP itself has been linked.

If called under control of function 'TAILOR', ensure that the submitted phase name is valid.

PARAMETERS

None

ENTRY-POINT

IKRRALNK

CALLED VIA

CALL IKRRALNK

INPUT

IKRXENT - internal representation block for function 'TAILOR'

OUTPUT

IKRCOMRG - MSHP communication region.

IKRRASRV

XSERV receive appendage

FUNCTION

PUT to the appropriate file and PUNCH output.

PARAMETERS

None

ENTRY-POINT

IKRRASRV

CALLED VIA

CALL IKRRASRV

INPUT

Data given by a XSERV program via IKRRCV.

OUTPUT

None.

IKRRCV

Receive a record from a system program.

FUNCTION

Catches the output record and goes back to the receive appendage routine.

PARAMETERS

WORKA CHAR(80) /* received record */

ENTRY-POINT

IKRRCV
IKRRCVEN after LIOCS PUT

CALLED VIA

CALL IKRRCV (WORKA)
(Out)

INPUT

None

OUTPUT

None.

IKRRDCE

Perform residence command.

FUNCTION

Several services are offered to handle the RESIDENCE information, that means the library location of MSHP administered products and components:

- IKRRDCE - The RESIDENCE command is processed. For selected products the specified production/generation libraries are attached to the appertaining components.
- IKRRDCGN - The library/sublibrary names for a component are found (full format: library 7 bytes , sublib 8 bytes).
- IKRRDCGS - Only the generation lib/sublib for a component is returned.
- IKRRDCGP - The library/sublibrary names for a component are found (command format: library.sublibrary).
- IKRRDCTE - Test whether two components have same library information.
- IKRRDCPR - Locate components of a specified production lib/sublib. The search is either started at the first component or continued after the last identified one.

PARAMETERS

```
ACCSET BIN(15) /* set-id of access information */
HFTYPE BIN(15) /* type of history file */
COMPKEY CHAR(*) /* component key */
CMP1KEY CHAR(*) /* component key */
CMP2KEY CHAR(*) /* component key */

1KEYDSECT BASED /* layout of component key */
 2 KEYTYPE BIN(8) /* type of key: */
 /* 1:=component-name */
 /* 2:=component-cell */
 /* 3:=product-cell (not valid */
 /* for 'IKRRDCPR') */
 2 KEYDATA CHAR(*) /* component key info */

1NAMEKEY DEFINED(KEYDATA) /* component key is name */
 2 KEYID CHAR(9) /* component-id */
 2 KEYLVL CHAR(3) /* level info */

1CELLKEY DEFINED(KEYDATA) /* component key is cell */
 2 KEYADR PTR(32) BDY(BYTE) /* address of component record */
 2 KEYRBA PTR(24) BDY(BYTE) /* RBA in history file */
EQUAL BIT(1) /* lib.sublib names are equal */

1SHARG1 /* search argument for IKRRDCPR */
 2 PRLIB /* production lib. print format */
 3 PRLLG BIN(8) /* length of library name */
 3 PRLNM CHAR(16) /* library name in print-format */
 2 NEXT BIT(1) /* function indicator OFF means */
 /* start from beginning, ON to */
 /* continue */

PRTLBPTR PTR(32) /* base of lib.sublib info */
```

ENTRY-POINT

IKRRDCE

CALLED VIA

IKRRDCE(HFTYPE)
(In)

INPUT

IKRRDC - IRP for function 'RESIDENCE'
MSHP - permanent history file

OUTPUT

MSHP - permanent history file

ENTRY-POINT

IKRRDCGN

CALLED VIA

CALL IKRRDCGN(HFTYPE,COMPKEY,ACCSET)
(In , In , Out)

INPUT

MSHP - permanent history file

OUTPUT

None

ENTRY-POINT

IKRRDCGS

CALLED VIA

CALL IKRRDCGS(HFTYPE,COMPKEY,ACCSET)
(In , In , Out)

INPUT

MSHP - permanent history file

OUTPUT

None

ENTRY-POINT

IKRRDCGP

CALLED VIA

CALL IKRRDCGP(HFTYPE,COMPKEY,PRTLBPTR)
(In , In , Out)

INPUT

MSHP - permanent history file

OUTPUT

None

ENTRY-POINT

IKRRDCTE

CALLED VIA

CALL IKRRDCTE(HFTYPE,CMP1KEY,CMP2KEY,EQUAL)
(In , In , In , Out)

INPUT

MSHP - permanent history file

OUTPUT

None

ENTRY-POINT

IKRRDCPR

CALLED VIA

CALL IKRRDCPR(HFTYPE,COMPKEY,SHARG1,PRTLBPTR)
(In , Out , In , Out)

INPUT

MSHP - permanent history file

OUTPUT

None.

IKRRDLV2

Get a level 2 command.

FUNCTION

Get either from console or from reader a level 2 command (detail control statement). Console input is prompted. The parser is used to analyze the statement. Level 1 or incorrect commands are rejected.

PARAMETERS

EOB BIT(1) /* end of block indicator */

ENTRY-POINT

IKRRDLV2

CALLED VIA

CALL IKRRDLV2(EOB)
(Out)

INPUT

None

OUTPUT

None.

IKRRDNXT

Read the next data.

FUNCTION

Read the next data, either from the console or from SYSIPT, depending on the users choice.

PARAMETERS

IN CHAR(80) /* area containing the next data statement */
EOD BIT(1) /* indicates end of data */

ENTRY-POINT

IKRRDNXT

CALLED VIA

CALL IKRRDNXT (IN,EOD)

INPUT

None

OUTPUT

None.

IKRREGCK

Trace macro service on generated members.

FUNCTION

This module performs the "regression check" on the system history file and informs about the result. Following subfunctions are supported:

1. Create list for regressed generated members.
2. Locate all generated members which are affected by the submitted macro, flag them as being "regressed", and insert them into the list.
3. Locate all generated members that match the submitted (generic) name, flag them as being "regressed", and insert them into the list.
4. Print all regressed generated members as denoted in the list.
5. Delete list of regressed generated members.
6. Flag the specified member as not being "regressed" and remove it from the list.

PARAMETERS

ACTION	BIN(8)	/* requested function	*/
SERVCOMP	CHAR(LENGTH(INSCMP))	/* serviced component	*/
SERVMAC	CHAR(LENGTH(GENMN))	/* serviced macro	*/
INFLMEMB	CHAR(LENGTH(GENMN))	/* influenced member	*/

ENTRY-POINT

IKRREGCK

CALLED VIA

The requested subfunction determines which parameters are needed in which sequence:

CALL IKRREGCK(ACTION)	/* create list	*/
(In)		
CALL IKRREGCK(ACTION,SERVMAC)	/* insert for macros	*/
(In , In)		
CALL IKRREGCK(ACTION,SERVCOMP,INFLMEMB)	/* insert influenced	*/
(In , In , In)		
CALL IKRREGCK(ACTION)	/* print list	*/
(In)		
CALL IKRREGCK(ACTION)	/* delete list	*/
(In)		
CALL IKRREGCK(ACTION,SERVCOMP,SERVMAC)	/* remove from list	*/
(In , In , In)		

INPUT

System history file

OUTPUT

System history file.

IKRREQFR

Request for free records.

FUNCTION

This module checks whether the requested number of free records are still available in the denoted history file. If so, it ensures that enough free records are preformatted and can be dequeued from the free-record chain. Otherwise the user is informed that the history file is full.

PARAMETERS

```
HFTYPE BIN(15)           /* history file (1=system,2=aux) */
REQRECS BIN(15)         /* number of requested records */
1RETINFO                /* return information */
 2 RETREQ BIN(8)         /* requested action: */
                        /* 1=quit if fails, 2=return */
 2 RETCOD BIN(8)        /* return code (0=OK,1=FULL) */
```

ENTRY-POINT

IKRREQFR

CALLED VIA

```
CALL IKRREQFR(HFTYPE,REQRECS,RETINFO)
           ( In , In , In/Out)
```

INPUT

Specified history file

OUTPUT

Return information.

IKRREWR

Rewrite all modified history file blocks to their respective file.

FUNCTION

Scan the buffer table for the buffer having the 'TO BE REWRITTEN' bit and call IKRRWBUF to rewrite such buffers.

PARAMETERS

None

ENTRY-POINT

IKRREWR

CALLED VIA

CALL IKRREWR

INPUT

None

OUTPUT

None.

IKRRLZAP

Correct object module.

FUNCTION

Alter an existing module (member type 'OBJ') or revert such an alteration. For altering, the specified module is punched to an intermediate file using the PUNCH function of the librarian. The changes are converted into rep-cards and appended to the module. The module is catalogued back into the library by the catalog function of the librarian.

If the UNDO function was selected, the rep-cards are removed in a similar process and the module is re-catalogued again.

PARAMETERS

```
AFFPTR PTR                /* pointer to IRP of affected module */
1CNTLFLG BIT(16)         /* control flags */
2 DOZAP BIT(1)           /* 1:= correct in RL */
2 UNDOZAP BIT(1)        /* 1:= UNDO the change in RL */
2 OK BIT(1)              /* 1:= ZAP was successful */
2 NEXTAFF BIT(1)        /* 1:=next "AFFECT" read */
2 * BIT(1)               /* reserved */
2 CATASTRO BIT(1)       /* module destroyed */
2 PHCOREQ BIT(1)        /* phase of COREQ PTF to handle */
2 * BIT(9)               /* reserved */
/* Following four parameters are needed for invocation of
/* IKRCKLV2 and IKRHDLV2. They are not touched in this module
PARM2 BIT(8)             /* "OPENFL" OR
PARM3 BIT(16)            /* "FFOUND" in IKRECOR
1PARM4                   /* "WRKBUFTB" in IKRECOR
2 * BIN(8)
2 * PTR
1PARM5                   /* "FREE" in IKRECOR
2 * PTR
2 * BIN
```

ENTRY-POINT

IKRRLZAP

CALLED VIA

```
IKRRLZAP(AFFPTR,CNTFLG,PARM2,PARM3,PARM4,PARM5)
(In ,In/Out,In ,In ,In ,In )
```

INPUT

IKRXENT - IRP for FUNCTION CORRECT/UNDO
IKR#AFF - SIT for AFFECTS
IKR#ALT - SIT for ALTER

OUTPUT

IKRXENT - IRP for function CORRECT/UNDO.

IKRRL2

Retrace overview list.

FUNCTION

Controls producing a list for the function 'RETRACE OVERVIEW'. The following lists are printed subsequently:

- The PRODUCT list
- The PTF/APAR list
- The APAR to PTF cross reference list
- The module cross reference list.

PARAMETERS

```
1 SORTVAL
  2 ASORTST  PTR(32)          /* sort area start address */
  2 ARDTBLST PTR(32)          /* record table start address */
  2 ASQTBLS  PTR(32)          /* sequence table start address */
  2 ASORTND  PTR(32)          /* sort area end address */
  2 ARDTBLND PTR(32)          /* record table end address */
  2 ASQTBLND PTR(32)          /* sequence table end address */
  2 INSRDN   FIXED(15)        /* inserted record number */
  2 NINSRDN  FIXED(15)        /* not inserted record number */
  2 LSRTRD   FIXED(8)         /* length of sort record */
  2 LSF      FIXED(8)         /* length of sort field */
  2 FSTCMPRB PTR(24)          /* first component RBA */
  2 FSTFTRRB PTR(24)          /* first feature RBA */
  2 NXTCMPRB PTR(24)          /* next component RBA */
```

ENTRY-POINT

IKRRL2

CALLED VIA

```
CALL IKRRL2(SORTVAL)
      (In/Out )
```

INPUT

Record and sequence table

OUTPUT

Record and sequence table.

IKRRL3

Print component lists.

FUNCTION

Process RETRACE COMPONENTS:

1. A sort table is built with one sort record entry for each component reflected in the history file as installed and not yet obsoleted, or - if a specific component is requested - only for that component.
2. For each sort record of a component, the sort table is completed with sort records for all libraries chained to the component. From this table, a list of all libraries assigned to the component is printed.
3. For each sort record of a correctly installed component, the sort table is completed with sort records for all PTFs applied to the component. From this state of the sort table, and after a line identifying the component, the PTFs are listed.
4. Similarly, (after removal of the PTF sort records), the sort table is completed with sort records for all APARs resolved by PTFs applied to the component, and for all local/APAR corrections of all local/APAR corrections of the component. The APARs, then, are listed from the sort table. For completing the sort table and for listing the PTFs and APARs, the same routines are used as in processing 'RETRACE PTF' and 'RETRACE APAR' respectively.

PARAMETERS

```
1SORTVAL
  2 ASORTST PTR(32)          /* sort area start address */
  2 ARDTBLST PTR(32)        /* record table start address */
  2 ASQTBLST PTR(32)        /* seq. table start address */
  2 ASORTND PTR(32)         /* sort area end address */
  2 ARDTBLND PTR(32)        /* record table end address */
  2 ASQTBLND PTR(32)        /* sequence table end address */
  2 INSRDN FIXED(15)        /* inserted record number */
  2 NINSRDN FIXED(15)       /* not inserted record number */
  2 LSRTRD FIXED(8)         /* length of sort record */
  2 LSF FIXED(8)            /* length of sort field */
  2 FSTCMPRB PTR(24)        /* first component RBA */
  2 FSTFTRRB PTR(24)        /* first feature RBA */
  2 NXCMPRB PTR(24)         /* next component RBA */

1COMPINFO                    /* information about specific */
                             /* component */
  2 COMP$ BIT(1)             /* 1 = component-id present */
  2 COMPID CHAR(12) INIT(' ') /* component and level */
  3 CMPID CHAR(9)            /* component-id */
  3 CMLV CHAR(3)             /* level-id
```

ENTRY-POINT

IKRRL3

CALLED VIA

```
CALL IKRRL3(SORTVAL,COMPINFO)
           (In/Out , In   )
```

INPUT

Sort table

OUTPUT

Record and sequence table.

IKRRL4

Print product list.

FUNCTION

Lists the products installed. The routine is called to process 'RETRACE PRODUCT', or to include the product list in the overview listing.

First a table of product sort records (consisting of product id and date of installation) is built. If there is not enough space to build this table, the user is informed how many pages more have to be allocated and MSHP is terminated.

The actual list is produced from the table by repeatedly scanning it for the (in collating sequence) next record. The record picked is also removed from the table to speed up searching.

PARAMETERS

1	SORTVAL		
2	ASORTST	PTR(32)	/* sort area start address */
2	ARDTBLST	PTR(32)	/* record table start address */
2	ASQTBLST	PTR(32)	/* seq. table start address */
2	ASORTND	PTR(32)	/* sort area end address */
2	ARDTBLND	PTR(32)	/* record table end address */
2	ASQTBLND	PTR(32)	/* sequence table end address */
2	INSRDN	FIXED(15)	/* inserted record number */
2	NINSRDN	FIXED(15)	/* not inserted record number */
2	LSRTRD	FIXED(8)	/* length of sort record */
2	LSF	FIXED(8)	/* length of sort record */
2	FSTCMPRB	PTR(24)	/* first component RBA */
2	FSTFTRRB	PTR(24)	/* first product RBA */
2	NXTCMPRB	PTR(24)	/* next component RBA */

ENTRY-POINT

IKRRL4

CALLED VIA

CALL IKRRL4(SORTVAL)
(In/Out)

INPUT

None

OUTPUT

None.

IKRRL5

Sort and print PTF list.

FUNCTION

Controls producing a list of all PTFs in PTF number sequence. First a sort table is built with one entry for each PTF. This is done by calling IKRBPRDT for each component. If the sort table overflows, MSHP terminates after informing the user how much more space would have been required.

Then, the actual listing is printed from the sort table by calling IKRPPTFL.

PARAMETERS

1	SORTVAL		
2	ASORTST	PTR(32)	/* sort area start address */
2	ARDBLST	PTR(32)	/* record table start address */
2	ASQBLST	PTR(32)	/* sequence table start address */
2	ASORTND	PTR(32)	/* sort area end address */
2	ARDBLND	PTR(32)	/* record table end address */
2	ASQBLND	PTR(32)	/* sequence table end address */
2	INSRDN	FIXED(15)	/* inserted record number */
2	NINSRDN	FIXED(15)	/* not inserted record number */
2	LSRTRD	FIXED(8)	/* length of sort record */
2	LSF	FIXED(8)	/* length of sort field */
2	FSTCMPRB	PTR(24)	/* first component RBA */
2	FSTFTRRB	PTR(24)	/* first feature RBA */
2	NXTCMPRB	PTR(24)	/* next component RBA */

ENTRY-POINT

IKRRL5

CALLED VIA

CALL IKRRL5(SORTVAL)
(In/Out)

INPUT

None

OUTPUT

None.

IKRRL6

Prepare APAR list.

FUNCTION

Controls producing a list of all APARs in APAR number sequence. The APARs listed are the local APAR fixes and those resolved by PTFs, which are correctly applied and have neither been REVOKED nor has the whole component become obsolete.

First, the sort table is built with one entry for each APAR. This is done by calling IKRBARDT for each component.

If the sort table overflows, MSHP terminates after informing the user of how much more space would have been needed.

Then, the actual listing is printed from the sort table by calling IKRPAPAL.

PARAMETERS

1	SORTVAL		
2	ASORTST	PTR(32)	/* sort area start address */
2	ARDTBLST	PTR(32)	/* record table start address */
2	ASQTBLS	PTR(32)	/* sequence table start address */
2	ASORTND	PTR(32)	/* sort area end address */
2	ARDTBLND	PTR(32)	/* record table end address */
2	ASQTBLE	PTR(32)	/* sequence table end address */
2	INSRDN	FIXED(15)	/* inserted record number */
2	NINSRDN	FIXED(15)	/* not inserted record number */
2	LSRTRD	FIXED(8)	/* length of sort record */
2	LSF	FIXED(8)	/* length of sort field */
2	FSTCMRBA	PTR(24)	/* first component RBA */
2	FSTFTRBA	PTR(24)	/* first feature RBA */
2	NXTCMRBA	PTR(24)	/* next component RBA */

ENTRY-POINT

IKRRL6

CALLED VIA

CALL IKRRL6(SORTVAL)
(In/Out)

INPUT

None

OUTPUT

Sort record- and sequence table.

IKRRL7

Print module lists.

FUNCTION

Controls and partly handles the 'RETRACE MODULES' function. An alphabetical listing of module names is produced. This list shows which component the module belongs to, which PTF or local/APAR fix it is affected by, and the date when the referred-to fix was applied. The listing is subdivided into three parts: affected phases (if any), affected modules (if any), and affected macros (if any). The listing is produced from a sort table, which is built by calling IKRSRTMD. If the sort table overflows, MSHP informs the user as to how much space is required and then terminates.

PARAMETERS

1	SORTVAL		
2	ASORTST	PTR(32)	/* sort area start address */
2	ARDTBLST	PTR(32)	/* record table start address */
2	ASQTBLST	PTR(32)	/* sequence table start address */
2	ASORTND	PTR(32)	/* sort area end address */
2	ARDTBLND	PTR(32)	/* record table end address */
2	ASQTBLND	PTR(32)	/* sequence table end address */
2	INSRDN	FIXED(15)	/* inserted record number */
2	NINSRDN	FIXED(15)	/* not inserted record number */
2	LSRTRD	FIXED(8)	/* length of sort record */
2	LSF	FIXED(8)	/* length of sort field */
2	FSTCMPRB	PTR(24)	/* first component RBA */
2	FSTFTRRB	PTR(24)	/* first feature RBA */
2	NXTCMPRB	PTR(24)	/* next component RBA */

ENTRY-POINT

IKRRL7

CALLED VIA

CALL IKRRL7(SORTVAL)
(In/Out)

INPUT

None

OUTPUT

Sort record- and sequence table.

IKRRMGEN

Remove generated members.

FUNCTION

This module builds and manipulates the list of generated members which have to be removed from the system history file and removes them on request. The following subfunctions are supported:

1. Create a new list.
2. Insert submitted member into the list.
3. Remove from the list all members which belong to a certain PTF.
4. Remove all generated members denoted in the list from the system history file.
5. Delete the list.

PARAMETERS

ACTION	BIN(8)	/* requested subfunction	*/
COMP	CHAR(LENGTH(REMCMPID))	/* component id	*/
MEMB	CHAR(LENGTH(REMGEMM))	/* generated member	*/
PTF	CHAR(LENGTH(REMPTAP))	/* PTF number	*/

ENTRY-POINT

IKRRMGEN

CALLED VIA

The requested subfunction determines, which parameters have to be specified in which sequence:

CALL IKRRMGEN(ACTION)	/* CREATE LIST	*/
(In)		
CALL IKRRMGEN(ACTION,COMP,MEMB,PTF)	/* INSERT INTO LIST	*/
(In , In , In , In)		
CALL IKRRMGEN(ACTION,COMP,PTF)	/* REMOVE FROM LIST	*/
(In , In , In)		
CALL IKRRMGEN(ACTION)	/* REMOVE FROM HISTORY F.	*/
(In)		
CALL IKRRMGEN(ACTION)	/* DELETE LIST	*/
(In)		

INPUT

System history file

OUTPUT

System history file.

IKRRTC

Retrace the history file (permanent).

FUNCTION

This module handles actions that are common for the various RETRACE subfunctions: It prepares the sort table and prints the common header page.

Depending upon which retrace option was chosen (represented in the IRP IKRXENT), the corresponding processing routine is called. On return the storage allocated for the sort table is freed.

PARAMETERS

1SORTVAL

2 ASORTST	PTR(32)	/* sort area start address	*/
2 ARDTBLST	PTR(32)	/* record table start address	*/
2 ASQTBLST	PTR(32)	/* seq. table start address	*/
2 ASORTND	PTR(32)	/* sort area end address	*/
2 ARDTBLND	PTR(32)	/* record table end address	*/
2 ASQTBLND	PTR(32)	/* sequence table end address	*/
2 INSRDN	FIXED(15)	/* inserted record number	*/
2 NINSRDN	FIXED(15)	/* not inserted record number	*/
2 LSRTD	FIXED(8)	/* length of sort record	*/
2 LSF	FIXED(8)	/* length of sort field	*/
2 FSTCMPRB	PTR(24)	/* first component RBA	*/
2 FSTFTRRB	PTR(24)	/* first product RBA	*/
2 NXTCMPRB	PTR(24)	/* next component RBA	*/

ENTRY-POINT

IKRRTC

CALLED VIA

CALL IKRRTC(SORTVAL)
(In/Out)

INPUT

IKRXRTC - Internal representation (IRP)

OUTPUT

None.

IKRRTRVE

Retrieve generated member information from permanent history file.

FUNCTION

Information of generated members is retrieved from a history file. There are two functions: get the overall macro reference list (for a component) or get a generated member with its associated information back into storage in IRP format.

PARAMETERS

```
HFTYPE BIN FIXED(15)      /* kind of history file      */
                            */
1WRKBUFTB                 /* work buffer control table  */
                            */
  2 BFADRMX BIN(8)        /* maximum number of work buffers */
                            */
  2 BFADRACT BIN(8)       /* actual number of obtained buffers */
                            */
  2 BFADR TAB(*) POINTER  /* addresses of obtained buffers */
                            */

CONTROL BIN FIXED(8)      /* control variable for case:  */
                            */
                            /* 1 overall macro table,      */
                            */
                            /* 2 generated member information */
                            */
```

ENTRY-POINT

IKRRTRVE

CALLED VIA

```
CALL IKRRTRVE(HFTYPE,WRKBUFTB,CONTROL)
              (In      ,In/Out  ,In)
```

INPUT

IKRXENT - interface to semantic checker
Permanent history file.

OUTPUT

IKRXENT - completed with generated member information.

IKRRWBUF

Rewrite a specific altered buffer.

FUNCTION

Determine to which history file this buffer belongs (PERM or TEMP), point to its physical location, and write it back.

PARAMETERS

```
BUFNBR BIN(15)          /* number of buffer which contains */
                        /* the block to be written      */
```

ENTRY-POINT

IKRRWBUF

CALLED VIA

```
CALL IKRRWBUF (BUFNBR)
              (In   )
```

INPUT

None

OUTPUT

None.

IKRSAASM

Assembly send appendage for function 'CORRECT'.

FUNCTION

Give input to assembler.

PARAMETERS

None

ENTRY-POINT

IKRSAASM

CALLED VIA

CALL IKRSAASM

INPUT

Source deck on SYS004

OUTPUT

None.

IKRSALIB

Send appendage for program LIBR.

FUNCTION

This module gets control if the librarian (LIBR) program is loaded by MSHP and issues a GET request for SYSIPT.

It feeds the LIBR program with control statements according to the requested function and - finally - with a ' ' card which has the effect that the LIBR program gets control back and starts processing.

The requested function and all information needed to build the control statements is denoted in the interface IKRSLI which is pointed to by CRGSENDP.

PARAMETERS

None

ENTRY-POINTS:

IKRSALIB

CALLED VIA

CALL IKRSALIB

INPUT

IKRSLI - Send appendage interface, pointed to by CRGSENDP
Accessed libraries and sublibraries

OUTPUT

Accessed libraries and sublibraries.

IKRSAMNT

Send appendage for cataloging members.

FUNCTION

This module expects at entry that a LIBR control statement (CATALOG, CATALR, or CATALS) has been read into the input area.

It determines, using pointer CRGSENDP, whether the member is to be cataloged or not.

It feeds this control statement to the LIBR program as well as all cards found on the input device until an end-of-data indicator is found (/).

PARAMETERS

None

ENTRY-POINT

IKRSAMNT

CALLED VIA

CALL IKRSAMNT

INPUT

1. List of acceptable members pointed to by CRGSENDP
2. Data records read from SYSIPT or SYSLOG

OUTPUT

None.

IKRSAPL

Select for re-generation.

FUNCTION

Selects MSHP TAILOR jobs from the generation file on SYS005, reads them onto a workfile (SYS003) and invokes the TAILOR function to generate the requested member.

The selection is triggered by information specified in the internal representation block (IRP), by which specific TAILOR jobs can be included or excluded.

PARAMETERS

None

ENTRY-POINT

IKRSAPL

CALLED VIA

CALL IKRSAPL

INPUT

1. IKRXCUM - internal representation block (IRP)
2. Generation file on SYS005 (80 or 81 records), unblocked; if on disk, the filename 'GENFILE' is required.

OUTPUT

1. IKRXCUM - internal representation block (IRP)

IKRSEND

Send a record to a called SCP program.

FUNCTION

Fills the interface table with the sent information and calls IKRGTMSH.

PARAMETERS

WORKA CHAR(80) /* card image of a control statement */
 /* to send to the SCP program */

ENTRY-POINT

IKRSEND

CALLED VIA

CALL IKRSEND (WORKA)
 (In)

INPUT

None

OUTPUT

None.

IKRSET

Abstract data type - set

FUNCTION

This module implements an abstract data type "SET"; it allows to collect data (called "members" or "elements") and to operate them in a way which is typical for a set. Untypical for a set is the ability of an sequential retrieve: In order to perform a certain function for all elements of the set, you may get them one by one using the "get first" and the "get next" operations.

Caution: Any "delete member" operation changes the sequence in the set.

For more functional details see the descriptions of each entry point:

```
IKRSETCS /* create a set */
IKRSETFS /* delete a set */
IKRSETIM /* insert member */
IKRSETDM /* delete member */
IKRSETIS /* is element ? */
IKRSETCD /* cardinality ? */
IKRSETLN /* length of member ? */
IKRSETGF /* get first member */
IKRSETGN /* get next member */
IKRSETGK /* get member according to key */
```

PARAMETERS

```
SETID BIN(15) /* set identifier */
ELEMENTL BIN(8) /* length of element */
CARD BIN(15) /* cardinality of set */
MEMBER CHAR( ) /* element */
ISELEM BIT(1) /* 1:=is element */
EOS BIT(1) /* 1:=end of set */
1KEY /* search information */
  2 KEYS BIN(15) /* start position of key */
  2 KEYL BIN(15) /* length of key */
  2 KEYD CHAR( ) /* search argument */
```

ENTRY-POINT

```
IKRSETCS /* create a set */
```

FUNCTION

Pre: None

Post: A set is created for elements of the specified length. The set-identifier is returned.

CALLED VIA

CALL IKRSETCS(SETID,ELEMENTL)
(Out , In)

INPUT

None

OUTPUT

None

ENTRY-POINT

IKRSETFS /* delete a set */

FUNCTION

Pre: The identified set must exist.

Post: The identified set is deleted. Its space is freed.

CALLED VIA

CALL IKRSETFS(SETID)
(In)

INPUT

None

OUTPUT

None

ENTRY-POINT

IKRSETIM /* insert member */

FUNCTION

Pre: The identified set must exist.

Post: The specified member is element of the identified set.

CALLED VIA

CALL IKRSETIM(SETID,MEMBER)
(In , In)

INPUT

None

OUTPUT

None

ENTRY-POINT

IKRSETDM /* delete member */

FUNCTION

Pre: The identified set must exist.

Post: The specified member is not element of the identified set. The sequence of elements in the set is changed.

CALLED VIA

```
CALL IKRSETDM(SETID,MEMBER)
           ( In , In )
```

INPUT

None

OUTPUT

None

ENTRY-POINT

```
IKRSETIS /* is element ?          */
```

FUNCTION

Pre: The identified set must exist.

Post: An indication is returned whether the specified member is element of the identified set.

CALLED VIA

```
CALL IKRSETIS(SETID,MEMBER,ISELEM)
           ( In , In , Out )
```

INPUT

None

OUTPUT

None

ENTRY-POINT

```
IKRSETCD /* cardinality ?          */
```

FUNCTION

Pre: The identified set must exist.

Post: The cardinality (number of inserted elements) of the identified set is returned.

CALLED VIA

```
CALL IKRSETCD(SETID,CARD)
           ( In , Out)
```

INPUT

None

OUTPUT

None

ENTRY-POINT

IKRSETLN /* length of member ? */

FUNCTION

Pre: The identified set must exist.

Post: The length of the elements in the identified set is returned.

CALLED VIA

CALL IKRSETLN(SETID,ELEMENTL)
(In , Out)

INPUT

None

OUTPUT

None

ENTRY-POINT

IKRSETGF /* get first member */

FUNCTION

Pre: The identified set must exist.

Post: Get the element which is currently the "first" in set. The "first" is either the first inserted, or any other if the first inserted has been deleted. The member pointer is updated and points to the second member. EOS is set if the set was empty. No member is returned in this case.

CALLED VIA

CALL IKRSETGF(SETID,MEMBER,EOS)
(In , Out ,Out)

INPUT

None

OUTPUT

None

ENTRY-POINT

IKRSETGN /* get next member */

FUNCTION

Pre: The identified set must exist.

Post: Get the element which is pointed to by the element pointer. The member pointer is updated by 1. EOS is set in case where the member pointer is higher then the cardinality of the set. No member is returned in this case.

CALLED VIA

```
CALL IKRSETGN(SETID,MEMBER,EOS)
           ( In , Out ,Out)
```

INPUT

None

OUTPUT

None

ENTRY-POINT

```
IKRSETGK /* get member according to key */
```

FUNCTION

Pre: The identified set must exist.

Post: Get the element which matches the search arguments according to the specified key range. ISELEM indicates whether a match was found or not. In case where found, the corresponding element is returned. The member pointer is not updated.

CALLED VIA

```
CALL IKRSETGK(SETID,KEY,MEMBER,ISELEM)
           ( In , In, Out , Out )
```

INPUT

None

OUTPUT

None.

IKRSHAPR

Locate a LOCAL/APAR FIX RECORD in a component.

FUNCTION

Using the COMPONENT RECORD, read the APAR-FIX chain and stop when the APAR RECORD is found or at end of queue if not found. If RECTYPE=LONGTERM then the requested APAR RECORD will be fixed.

PARAMETERS

HFTYPE BIN(15)	/* HISTORY FILE TYPE	*/
	/* 1=PERM,2=TEMP	*/
RECTYPE BIN(15)	/* RECORD TYPE	*/
	/* 1=SHORTERM,2=LONGTERM	*/
1CELLID	/* CELL-ID OF THE RECORD	*/
3 ADDRESS POINTER(32)	/* ITS ADDRESS	*/
3 RBA POINTER(24)	/* ITS RBA	*/
1ARGUMENT	/* SEARCH ARGUMENT	*/
3 COMPON CHAR(12)	/* COMPONENT AND LEVEL -ID	*/
3 APAR CHAR(8)	/* APAR NUMBER	*/

ENTRY-POINT

IKRSHAPR

CALLED VIA

CALLER IKRSHAPR (HFTYPE,RECTYPE,CELLID,ARGUMENT)
(In , In , Out , In)

INPUT

None

OUTPUT

None.

IKRSHCMP

Search a specific COMPONENT record.

FUNCTION

Read the HEADER record, extract the COMPONENT record pointer and scan the COMPONENT queue to find a COMPONENT. Read the first COMPONENT record. If the first record isn't the requested one, the COMPONENT queue is searched till the requested one is found or the end of queue is reached. If RECTYPE = LONGTERM, the requested record is fixed.

PARAMETERS

HFTYPE BIN(15)	/* HISTORY FILE TYPE	*/
	/* 1=PERM,2=TEMP	*/
RECTYPE BIN(15)	/* HIST.FILE RECORD TYPE	*/
	/* 1=SHORTERM,2=LONGTERM	*/
1CELLID	/* CELL-ID OF THE REQ. RECORD	*/
3 ADDRESS POINTER(32)	/* ITS ADDRESS	*/
3 RBA POINTER(24)	/* ITS RBA	*/
1ARGUMENT	/* SEARCH ARGUMENT	*/
3 COMPON CHAR(9)	/* COMPONENT ID	*/

ENTRY-POINT

IKRSHCMP

CALLED VIA

CALL IKRSHCMP(HFTYPE,RECTYPE,CELLID,ARGUMENT)
(In , In , Out , In)

INPUT

None

OUTPUT

None.

IKRSHGEN

Locate a generated member record in a COMPONENT.

FUNCTION

Using the COMPONENT extension record, read the generation activity record chain and stop when the generated member record is found or at end of queue if not found.

PARAMETERS

HFTYPE BIN(15)	/* history file type	*/
	/* 1=perm,2=temp	*/
RECTYPE BIN(15)	/* record type	*/
	/* 1=shorterm,2=longterm	*/
1CELLID	/* cell-id of the req. record	*/
3 ADDRESS POINTER(32)	/* its address	*/
3 RBA POINTER(24)	/* its RBA	*/
1ARGUMENT	/* search argument	*/
3 COMPON CHAR(12)	/* component-id	*/
3 MEMBER CHAR(10)	/* generated member name	*/

ENTRY-POINT

IKRSHGEN

CALLED VIA

CALL IKRSHGEN (HFTYPE,RECTYPE,CELLID,ARGUMENT)
(In , In , Out , In)

INPUT

None.

OUTPUT

None.

IKRSHGRP

Search a specific product.

FUNCTION

Scan the PRODUCT queue and search for the requested PRODUCT. If RECTYPE=LONGTERM the requested RECORD will be fixed.

PARAMETERS

HFTYPE BIN(15)	/* history file type	*/
	/* 1=perm,2=temp	*/
RECTYPE BIN(15)	/* record type	*/
	/* 1=shorterm,2=longterm	*/
1CELLID	/* cell-id of record	*/
3 ADDRESS POINTER(32)	/* its address	*/
3 RBA POINTER(24)	/* its RBA	*/
1ARGUMENT	/* search argument	*/
3 PRODUCT CHAR(3)	/* product - id	*/

ENTRY-POINT

IKRSHGRP

CALLED VIA

CALL IKRSHGRP(HFTYPE,RECTYPE,CELLID,ARGUMENT)
(In , In , Out , In)

INPUT

None

OUTPUT

None.

IKRSHPRD

Search a specific product.

FUNCTION

Scan the PRODUCT queue and search for the requested PRODUCT. If RECTYPE=LONGTERM the requested RECORD will be fixed.

PARAMETERS

HFTYPE BIN(15)	/* history file type	*/
	/* 1=perm,2=temp	*/
RECTYPE BIN(15)	/* record type	*/
	/* 1=shorterm,2=longterm	*/
1CELLID	/* cell-id of record	*/
3 ADDRESS POINTER(32)	/* its address	*/
3 RBA POINTER(24)	/* its RBA	*/
1ARGUMENT	/* search argument	*/
3 PRODUCT CHAR(6)	/* product - id	*/

ENTRY-POINT

IKRSHGRP

CALLED VIA

CALL IKRSHPRD(HFTYPE,RECTYPE,CELLID,ARGUMENT)
(In , In , Out , In)

INPUT

None

OUTPUT

None.

IKRSHPTF

Locate a PTF record in a component.

FUNCTION

Using the component record, read the PTF chain and stop when the PTF record is found or at end of queue if not found. If RECTYPE=LONGTERM then the requested PTF RECORD will be fixed.

PARAMETERS

HFTYPE BIN(15)	/* HISTORY FILE TYPE	*/
	/* 1=PERM,2=TEMP	*/
RECTYPE BIN(15)	/* RECORD TYPE	*/
	/* 1=SHORTERM,2=LONGTERM	*/
1CELLID	/* CELL-ID OF THE RECORD	*/
3 ADDRESS POINTER(32)	/* ITS ADDRESS	*/
3 RBA POINTER(24)	/* ITS RBA	*/
1ARGUMENT	/* SEARCH ARGUMENT	*/
3 COMPON CHAR(12)	/* COMPONENT AND LEVEL- ID	*/
3 PTF CHAR(8)	/* PTF NUMBER	*/

ENTRY-POINT

IKRSHPTF

CALLED VIA

CALL IKRSHPTF (HFTYPE,RECTYPE,CELLID,ARGUMENT)
(In , In , Out , In)

INPUT

None

OUTPUT

None.

IKRSHREL

Search a component record with a specific level number.

FUNCTION

Read the 1st component record and scan the installation queue up to its end or to the requested component record.

PARAMETERS

HFTYPE BIN(15)	/* history file type	*/
	/* 1=perm,2=temp	*/
RECTYPE BIN(15)	/* record type	*/
	/* 1=shortterm,2=longterm	*/
1CELLID	/* cell-id of the record	*/
3 ADDRESS POINTER(32)	/* its address	*/
3 RBA POINTER(24)	/* its RBA	*/
1ARGUMENT	/* search argument	*/
3 COMPON CHAR(9)	/* component-id	*/
3 LEVEL CHAR(3)	/* level number	*/

ENTRY-POINT

IKRSHREL

CALLED VIA

CALL IKRSHREL (HFTYPE,RECTYPE,CELLID,ARGUMENT)
(In , In , Out , In)

INPUT

None

OUTPUT

None.

IKRSLA

Modify user label area.

FUNCTION

This module interfaces with the user label area and offers the following two functions:

1. Add new label(s):
 - Get the current user labels and store them into the SCP-area (pointed to by CRGSCPAD).
 - Clear the user label area.
 - Add the specified label(s) in front and write the current labels behind.
2. Remove label(s):
 - Get the current user labels and store them into the SCP-area (pointed to by CRGSCPAD).
 - Clear the user label area.
 - Rewrite all labels into the label area except those which have to be removed.

PARAMETERS

```
1REMLIST,                /* list of labels to be removed */
 3 REMIND BIT(8)         /* X'FE':=removing requested */
 3 REMNAME CHAR(7)      /* name of label to be removed */

ADDLABL CHAR(*)         /* label to be added */
```

ENTRY-POINT

IKRSLA

CALLED VIA

Depending on the requested function:

```
CALL IKRSLA(ADDLABL,ADDLABL,...) /* adding labels */
      ( In , In ,...)
CALL IKRSLA(REMLIST,REMLIST,...) /* removing labels */
      ( In , In ,...)
```

INPUT

User label area

OUTPUT

User label area.

IKRSLZAP

Correct source macro.

FUNCTION

Alter an existing source code macro or reinstall previous version. The requested changes are collected. Depending on the macro type (edited macros are assumed for all, except A-type) the updates are handled. For A-type macros, the update list is submitted to a librarian update function. For edited macros, the macro is de-edited together with the update input. The output is used as input for re-editing by the assembler, this output is again fed to a librarian CATALOG step. The In/Output file is used as SYS004.

In case of an UNDO operation, the old macro version is catalogued again and the APAR entry removed from the history file, providing no mismatching condition is recognized.

PARAMETERS

AFFPTR	PTR	/* pointer to the affected macro IRP */	*/
1CNTLFLG	BIT(16)	/* control flags	*/
2 DOZAP	BIT(1)	/* 1 = correction	*/
2 UNDOZAP	BIT(1)	/* 1 = correction removal	*/
2 OK	BIT(1)	/* 1 = correction successful	*/
2 NEXTAFF	BIT(1)	/* 1 = next affect read	*/
2 *	BIT(1)	/* reserved	*/
2 CATASTRO	BIT(1)	/* 1 = catastrophe during macro CATAL*/	*/
2 *	BIT(2)	/* reserved	*/
2 UNDOSL	BIT(1)	/* 1 = UNDO for SL macro created	*/
2 INCOM	BIT(1)	/* 1 = UNDO job is additional	*/
2 *	BIT(6)	/* reserved	*/
/* Following are parameters needed for IKRCKLV2 and IKRHDLV2			*/
/* invocation. They are not modified in this module			*/
P2	BIT(8)	/* "OPENFL" from IKRECOR	*/
P3	BIT(16)	/* "FFOUND" from IKRECOR	*/
1P4		/* "WRKBUEFTB" from IKRECOR	*/
2 *	BIN(8)		
2 *	PTR		
1P5		/* "FREE" from IKRECOR	*/
2 *	PTR		
2 *	BIN		

ENTRY-POINT

IKRSLZAP

CALLED VIA

IKRSLZAP(AFFPTR,CNTLFLG,P2,P3,P4,P5)
(In ,In/Out ,In,In,In,In)

INPUT

IKRXENT - IRP for function CORRECT/UNDO
IKR#AFF - SIT for AFFECTS
IKR#IST - SIT for INSERT
IKR#RPL - SIT for REPLACE
IKR#DEL - SIT for DELETE

OUTPUT

IKRXENT: IRP for function 'CORRECT/UNDO'.

IKRSPCLS

Close a specific file.

FUNCTION

Locate the DTF of the specific file and CLOSE it.

PARAMETERS

FILE BIN(15) /* Index in the file-table (IKRCOMRG) */
/* Determining which file to close */

ENTRY-POINT

IKRSPCLS

CALLED VIA

CALL IKRSPCLS (FILE)
(In)

INPUT

None

OUTPUT

None.

IKRSREPQ

Sort repeat question procedure.

FUNCTION

Checks if sort table was big enough to produce the required RETRACE listing. If not, the user is first informed how many more pages of 2K are required and then MSHP is terminated.

PARAMETERS

1	SORTVAL		
2	ASORTST PTR(32)	/* sort area start address	*/
2	ARDTBLST PTR(32)	/* record table start address	*/
2	ASQTBLS PTR(32)	/* sequence table start address	*/
2	ASORTND PTR(32)	/* sort area end address	*/
2	ARDTBLND PTR(32)	/* record table end address	*/
2	ASQTBLS PTR(32)	/* sequence table end address	*/
2	INSRDN FIXED(15)	/* inserted record number	*/
2	NINSRDN FIXED	/* not inserted record number	*/
2	LSRTRD FIXED(8)	/* length of sort record	*/
2	LSF FIXED(8)	/* length of sort record	*/
2	FSTCMPRB PTR(24)	/* first component RBA	*/
2	FSTFTRRB PTR(24)	/* first feature RBA	*/
2	NXTCMPRB PTR(24)	/* next component RBA	*/

ENTRY-POINT

IKRDREPQ

CALLED VIA

CALL IKRSREPQ (SORTVAL)
(In/Out)

INPUT

None

OUTPUT

None.

IKRSRTMD

Insert PTF/APAR module records.

FUNCTION

Builds a sort table with information extracted from all the history file records associated with the current component record. If the component has been correctly installed and is not obsolete, the routine IKRBMODR is called to build a sort record for each module-in-PTF, module-in-local/APAR, and generated member record found in the history file which is related to the current component. The sort table records are then inserted into the sort table by IKRBINS.

The sort table records can be built in either a short "cross reference" form or a "main" form depending on LISTTYPE.

The short form consists of a:

1. One byte type field which is used to distinguish among phases, modules, and macros
2. The phase, module, or macro name
3. 8 characters containing the PTF number, the APAR number, or the word GENERATE

The main form consists of a:

1. One byte type field which is used to distinguish among phases, modules, and macros
2. The phase, module, or macro name
3. The RBA of the component
4. The RBA of the PTF, the APAR, or the generated member.
5. The RBA of the module record which is needed to retrieve from it fix descriptions for local/APAR fixes.

If the sort table overflows, the user is informed as to how much additional space is needed to hold the additional sort table records and then MSHP will terminate.

PARAMETERS

1 SORTVAL		/* input/output	*/
2 ASORTST	PTR(32)	/* sort area start address	*/
2 ARDTBLST	PTR(32)	/* record table start address	*/
2 ASQTBLST	PTR(32)	/* sequence table start address	*/
2 ASORTND	PTR(32)	/* sort area end address	*/
2 ARDTBLND	PTR(32)	/* record table end address	*/
2 ASQTBLND	PTR(32)	/* sequence table end address	*/
2 INSRDN	FIXED(15)	/* inserted record number	*/
2 NINSRDN	FIXED(15)	/* not inserted record number	*/
2 LSRTRD	FIXED(8)	/* length of sort record	*/
2 LSF	FIXED(8)	/* length of sort record	*/
2 FSTCMPRB	PTR(24)	/* first component RBA	*/
2 FSTFTRRB	PTR(24)	/* first feature RBA	*/
2 NXTCMPRB	PTR(24)	/* next component RBA	*/
LISTTYPE	CHAR(1)	/* list type parameter	*/

ENTRY-POINT

IKRSRTMD

CALLED VIA

CALL IKRSRTMD (SORTVAL,LISTTYPE)
(In/Out ,In)

INPUT

Record and sequence table

OUTPUT

Record and sequence table.

IKRSTMTS

List statements on PTF file.

FUNCTION

Lists on SYSLST (requested by MSHP):

- The service tape documentation

Prints the service tape documentation, which contains information concerning preventive and corrective service from the particular service tape.

- The coverletter file

Prints coverletters of those PTFs which are specified in the detail control statement PTF. The following is printed for all requested PTFs.

JCL statements

JCL comments

MSHP function control statements

MSHP detail control statements

PARAMETERS

None

ENTRY-POINT

IKRSTMTS

CALLED VIA

CALL IKRSTMTS

INPUT

IKRXCUM - IRP of 'LIST' statement

PTF-file on SYS004

OUTPUT

None.

IKRSVCMD

Save and restore MSHP command.

FUNCTION

This module implements a stack for MSHP commands in the GETVIS area. It may be used when MSHP must reload itself in the partition in order to execute the command.

Following functions are supported:

1. Get virtual space and save the current MSHP commands as denoted in the command stack (IKRPRCS2). The address of the created stack in the GETVIS area is stored into the user area of the partition COMREG. Set next-command-pointer to first saved command.
2. Test if a stack in the GETVIS area has been built.
3. Return the command which the next-command-pointer denotes. In case where the stack is exhausted, set EOS. Update the next-command-pointer.
4. Free the used virtual space.

PARAMETERS

ACTION BIN(8)	/* requested function	*/
PRESENT BIT(1)	/* 1:=stack is present	*/
CMDLINE CHAR(132)	/* command line	*/
EOS BIT(1)	/* 1:=end of stack	*/

ENTRY-POINT

IKRSVCMD

CALLED VIA

It depends on the requested subfunction, which parameters have to be specified in which order:

CALL IKRSVCMD(ACTION)	/* create stack	*/
(In)		
CALL IKRSVCMD(ACTION,PRESENT)	/* test presence	*/
(In , Out)		
CALL IKRSVCMD(ACTION,CMDLINE,EOS)	/* get command line	*/
(In , Out ,Out)		
CALL IKRSVCMD(ACTION)	/* delete stack	*/
(In)		

INPUT

1. Partition communication region
2. IKRPRCS1 - number of commands in MSHP command stack
3. IKRPRCS2 - MSHP command stack

OUTPUT

Partition communication region.

IKRSWTHF

Switch history files.

FUNCTION

This module pages out all history file buffers and clears the history file control area in the control buffer.

If switches the history file information in the MSHP communication region (DTF and control flag): The former system history file is now the auxiliary, the former auxiliary is now the system history file.

PARAMETERS

None

ENTRY-POINT

IKRSWTHF

CALLED VIA

CALL IKRSWTHF

INPUT

1. MSHP communication region
2. Control buffer in buffer area

OUTPUT

1. MSHP communication region
2. Control buffer in buffer area.

IKRTAIL

Execution routine for TAILOR command.

FUNCTION

This module generates members as requested via the TAILOR or the SELECT GENFILE command.

Prior to any generation activity, following global checks are performed:

- Work files must be assigned (SYSLNK for generating phases, SYS004 for other members).
- The component to which the generated member belongs must have an entry in the system history file.
- Residence information for the component must be available.
- The component's sublibraries must be accessible.

The generation activity for a single member consists of two parts: The checking and the execution part.

In case where the checking fails, generation for that member is terminated and processing continues with the generation of the next requested member (if any).

Following member specific checks are performed:

- Protect members which are generated with the same name in the same sublibrary.
- Protect members which are affected by a PTF or a local/APAR fix.
- Ensure that the generation data is available.
- Ensure that there is space enough in the system history file.

For the execution part, open the work files and write the user specified data onto them.

In case where two system control programs are to be called, invoke the first one and append its output to the work file.

Invoke the final system program which picks up the data from the work file (SYSLNK for LNKEDT, SYS004 for LIBR/MAINT) and generates the requested member.

Update the system history file.

PARAMETERS

1WRKBUFTB	/* table of work buffers */
2 BFADRMX BIN(8)	/* maximum # of buffers in table */
2 BFADRACT BIN(8)	/* actual # of obtained buffers */
2 BFADRTBL(¢) PTR(32)	/* addresses of obtained buffers */

ENTRY-POINT

IKRTAIL

CALLED VIA

CALL IKRTAIL(WRKBUFTB)
(In/Out)

INPUT

1. IKRXENT - Internal representation block (IRP)
2. System history file
3. Generation data
4. User's libraries

OUTPUT

1. Updated system history file
2. Updated user's libraries

IKRTRLCP

Translate component-id.

FUNCTION

This module translates the internal representation of a component-id into the representation which is used externally for printing on SYSLST or displaying on the console:

12 characters of the format

XXXXXXXXXXXX (for example, 566630101H07)

14 characters of the format

XXX-XXXX-XXX (for example, 5666-30101-H07)

In case where the last three characters of the internal format are not specified (blank), the output format is

XXX-XXXX, e.g. 5666-30101

PARAMETERS

1CMPEXT		/* external format of component	*/
2 CMPE1 CHAR(4)		/* type number	*/
2 CMPEF1 CHAR(1)		/* dash	*/
2 CMPE2 CHAR(5)		/* program number	*/
2 CMPEF2 CHAR(1)		/* dash	*/
2 CMPE3 CHAR(3)		/* level number	*/
1CMPINT		/* internal representation	*/
2 CMPI1 CHAR(4)		/* type number	*/
2 CMPI2 CHAR(5)		/* program number	*/
2 CMPI3 CHAR(3)		/* level number	*/

ENTRY-POINT

IKRTRLCP

CALLED VIA

CALL IKRTRLCP(CMPEXT,CMPINT)
 (Out , In)

INPUT

None

OUTPUT

None.

IKRUIRP

Build IRP for UNDO local APAR/fix.

FUNCTION

As a reverse retrieve function, all information about an APAR is picked up from the history file and filled in core into the corresponding IRP. This is a preparation for the function 'UNDO'.

PARAMETERS

1	WRKBUFTB		/* input/output */
2	BFADRMX	BIN(8)	/* maximum number of work buffers */
2	BFADRACT	BIN(8)	/* actual no. of obtained buffers */
2	BFADRTBL(*)	PTR	/* addresses of obtained buffers */
	INVFIX	BIT(1)	/* 1:=non-removable fix (output) */
	ANYLINK	BIT(1)	/* 1:=LINKBOOK found (output) */
	SVLNKPTR	PTR	/* PTR to link book table (output)*/
	RBAMIA	PTR(24)	/* RBA of MIA record (input) */
1	FREE		/* free space in IRP info (output)*/
2	FREEPTR	PTR	/* free space pointer */
2	FREESP	BIN	/* free space counter */
	APARADDR	PTR	/* incore APAR record address */

ENTRY-POINT

IKRUIRP

CALLED VIA

IKRUIRP(WRKBUFTB,INVFIX,ANYLINK,SVLNKPTR,RBAMIA,FREE,APARADDR)
(In/Out ,Out ,Out ,Out ,In ,Out ,In)

INPUT

Permanent history file

OUTPUT

IKRXENT - IRP for 'UNDO'.

IKRUNDO

Undo local APAR/fix.

FUNCTION

An APAR fix is removed both from history file and from members. Build complete IRP and call for each affected phase/module/macro the corresponding reverting routine. Remove the APAR from the history file. Inform the user and the caller if 'UNDO' could not terminate successfully.

PARAMETERS

RETURNCD BIN(15) /* 1:= alright, 2:= failed */

ENTRY-POINT

IKRUNDO

CALLED VIA

CALL IKRUNDO(RETURNCD)
 (Out)

INPUT

IKRXENT - IRP for 'UNDO'

OUTPUT

IKRXENT - IRP for 'UNDO'.

IKRUOMT

Update table of overall included macros.

FUNCTION

Integrate the new macros into the overall macro table for the component, and build for them the member reference table. Count the macros which have been added to the overall table and return the corresponding number.

Update the work buffer control table for every obtained buffer.

Remove specific macro table from IRP and free those buffers.

PARAMETERS

MADD#	BIN(15)	/* number of macros added to the	*/
		/* overall macro table	*/
1WRKBUFTB		/* work buffer control table	*/
2 BFADRMX	BIN(8)	/* maximum number of work buffers	*/
2 BFADRACT	BIN(8)	/* actual no. of obtained buffers	*/
2 BFADRTAB(*)	PTR(32)	/* addresses of obtained buffers	*/

ENTRY-POINT

IKRUOMT

CALLED VIA

```
CALL IKRUOMT(MADD#,WRKBUFTB)
           (Out ,In/Out )
```

INPUT

IKRXENT - Internal representation (IRP)

OUTPUT

IKRXENT - Internal representation (IRP).

IKRUPGR

PTF application checking

FUNCTION

This module handles the PTF application. It always assumes that an auxiliary history file exists containing all PTFs to be applied.

Using this auxiliary file and the user's system history file, it performs the following three functions:

```
IKRUPGR    /* Check and select PTFs      */
IKRUPGRL   /* Link all linkbooks         */
IKRUPGRU   /* Update system history file */
```

For details see functional description for each entry point.

PARAMETERS

None

ENTRY-POINT

```
IKRUPGR    /* Check and select PTFs      */
```

FUNCTION

Using system and auxiliary history file information, this function determines the set of PTFs which can be applied.

A PTF might be rejected for following reasons:

- its component is not completely installed or does not have residence information,
- its component's sublibraries cannot be accessed,
- it is already installed (and not explicitly requested),
- it is superseded on user system,
- it has been previously revoked (and is not explicitly requested),
- its requirements are not met,
- it would overwrite a local/APAR fix and the user wants to keep the fix,
- it would overwrite a generated member the user wants to keep.

Print the set of PTF which are selected and ready for application and ask the user for confirmation.

Determine which members of which PTFs have to be picked up from the tape.

In case where REVOKABLE was specified, determine which PTFs have to be grouped together in order to create a single backout-PTF containing all common members and the names of the affected PTFs. Create the backout tape (function INSTALL) or job (function APPLY).

CALLED VIA

```
CALL IKRUPGR
```

INPUT

1. System history file
2. Auxiliary history file

OUTPUT

1. System history file
2. Auxiliary history file
3. Backout device (tape or SYSPCH)

ENTRY-POINT

IKRUPGRL /* Link all linkbooks */

FUNCTION

Determines all linkbooks of all PTFs which have not been linked.

It accesses the sublibraries into which the phases must be linked (according to system history file information), invokes the Linkage Editor to link them and updates the history file (PTFs are flagged as being linked).

CALLED VIA

CALL IKRUPGRL

INPUT

1. System history file
2. Auxiliary history file (not for RESTART)

OUTPUT

1. System history file
2. User's libraries

ENTRY-POINT

IKRUPGRU /* Update system history file */

FUNCTION

Update the user's system history file. Application of the PTFs as denoted on the auxiliary history file may have caused changes of the system history file status:

- Flag all members which are regressed by the applied service,
- remove the entries of local/APAR fixes which are overwritten,
- flag all invalidated local/APAR fixes,
- remove the entries of generated members which are overwritten.

Inform the user which generated members have to be re-generated since included macros are affected by the applied service.

CALLED VIA

CALL IKRUPGRU

INPUT

1. System history file
2. Auxiliary history file

OUTPUT
System history file

IKRWKF

Workfiles (SYS001,SYS003,SYS004) access module

FUNCTION

Provide facilities to OPEN, READ, or WRITE from or to the workfiles.

PARAMETERS

None

ENTRY-POINTS:

IKRWKF01	OPEN	SYS001
IKRWKF03	OPEN	SYS003
IKRWKF04	OPEN	SYS004
IKRWKFR1	GET	SYS001
IKRWKFR3	GET	SYS003
IKRWKFR4	GET	SYS004
IKRWKFW1	PUT	SYS001
IKRWKFW3	PUT	SYS003
IKRWKFW4	PUT	SYS004

CALLED VIA

CALL IKRWKF

INPUT

IKRWKF1W:	A) WORKA CHAR(80).....	data to be written
IKRWKF3W:	A) WORKA CHAR(80).....	data to be written
IKRWKF4W:	A) WORKA CHAR(80).....	data to be written

OUTPUT

IKRWKF1R:	A) WORKA CHAR(80).....	read data
	B) EOF BIT(1).....	ON if EOF is reached
IKRWKF3R:	A) WORKA CHAR(80).....	read data
	B) EOF BIT(1).....	ON if EOF is reached
IKRWKF4R:	A) WORKA CHAR(80).....	read data
	B) EOF BIT(1).....	ON if EOF is reached.

IKRWOF

Console interface module

FUNCTION

Issue message on the console and keep it on the screen.

PARAMETERS

1MESSAGE
2 MSGLEN BIN(8) /* message length */
2 MSG CHAR(*) /* message */

ENTRY-POINT

IKRWOF

CALLED VIA

CALL IKRWOF (MESSAGE)
(In)

INPUT

None

OUTPUT

None.

IKRWOR

Console interface module (Write and Read)

FUNCTION

Issue message on the console and read the answer.

PARAMETERS

(VLIST MUST BE SPECIFIED)

```
1MESSAGE
  2 MSGLEN BIN(8)           /* message length           */
  2 MSG    CHAR(*)         /* message                   */
GOTMSGGA  PTR(32)          /* This optional parameter is */
                                     /* used when the input field is */
                                     /* not expected in IKRSTATA.   */
```

ENTRY-POINT

IKRWOR

CALLED VIA

```
CALL IKRWOR (MESSAGE,GOTMSGGA)
              (In      ,In      )
```

INPUT

None

OUTPUT

None.

IKRWRSQ

History file block sequential write

FUNCTION

IKRWRSQ writes a history file block sequentially to a given history file (permanent, temporary, tape).

PARAMETERS

1) BLOCK CHAR(HFBLKLEN)	/* block to be written	*/
2) HFTYPE BIN(15)	/* history file type where the file is	*/
	/* to be written	*/
	/* 0001 for permanent history file	*/
	/* 0002 for temporary history file	*/
	/* 0003 for history file on tape	*/

ENTRY-POINT

IKRWRSQ

CALLED VIA

CALL IKRWRSQ (BLOCK,HFTYPE)
(In ,In)

INPUT

None

OUTPUT

None.

IKRWTO

Console interface module (Write to operator console)

FUNCTION

Obtain the correct message length and write the message to the operator console.

PARAMETERS

```
1MESSAGE  
3 MSGLEN BIN(8)           /* message length */  
3 MSG CHAR(*)             /* message          */
```

ENTRY-POINT

IKRWTO

CALLED VIA

```
CALL IKRWTO(MESSAGE)  
      ( Out  )
```

INPUT

None

OUTPUT

None.

IKRWTP

Print a line.

FUNCTION

Print a line, perform carriage control requested by the caller, and take care of the actual page size.

PARAMETERS

1) LINE		/* line to be printed	*/
CC	CHAR(1)	/* control character (standard)	*/
PLEN	BIN(8)	/* length of print area, 120 characters max.*/	
PDATA	CHAR(*)	/* data to be printed	*/
2) HEADER1		/* top header	*/
H1LEN	BIN(8)	/* length of 1st header, 120 characters max.*/	
H1DATA	CHAR(*)	/* text of 1st header	*/
3) HEADER		/* second header	*/
3 H2LEN	BIN(8)	/* length of 2nd header, 120 characters max.*/	
3 H2DATA	CHAR(*)	/* text of 2nd header	*/

ENTRY-POINT

IKRWTP

CALLED VIA

CALL IKRWTP (LINE,HEADER1,HEADER2)
 (In ,In ,In)

INPUT

None

OUTPUT

None.

Chapter 3: Organization Information

This chapter gives cross-reference information; for example, the modules that are called by a given module, or the MSHP COMREG fields used by a given module. Figure 12 is an overview and guide for finding the figure applicable to your information requirement.

Information Wanted	Information at Hand	Refer to Figure
Modules calling a given module	Module name	12-2
Modules called by a given module	Module name	12-3
Macros invoked by a given module	Module name	12-4
Modules that invoke a given macro	Macro name	12-5
MSHP communication region fields (CRGxxxx) used by a given module	Module name	12-6
Modules that use a given MSHP communication region field (CRGxxxx)	Field name of the MSHP COMREG	12-7

Figure 12. Information Retrieval Guide

Module-Module Interrelations

Figure 12-2: Sorted by Calling Modules

IKRAAPAR CALLS	IKRDATE IKRDQFR IKRFIX IKRFRBUF IKRGTBUF IKRGTREC IKRPCREQ IKRQUEUE IKRREQFR IKRREWR IKRSHAPR IKRSHREL IKRTRLCP
IKRACMP1 CALLS	IKRREQFR
IKRACOMP CALLS	IKRACMP1 IKRDATE IKRDIAGN IKRDQFR IKRGTHD IKRPCREQ IKRQUEUE IKRSHREL IKRTRLCP
IKRAGEN CALLS	IKRDATE IKRDEQ IKRDQFR IKRFRBUF IKRGTBUF IKRGTIMT IKRGTREC IKRQUEUE IKRREQFR IKRREWR IKRRTRVE IKRSHGEN IKRSHREL IKRUOMT
IKRAPPLY CALLS	IKRACOMP IKRAPTF IKRCKCMP IKRCLOSE IKRCREAT IKRDATE IKRDIAGN IKREMBDA IKREMBDY IKREXIT IKRFRBUF IKRGTBUF IKRGTINF IKRGTREC IKRMOD IKRMARGIN IKROPHF IKRRDNXT IKRREWR IKRSETCD IKRSETCS IKRSETFS IKRSETGF IKRSETGN IKRSETIM IKRSHCMP IKRSHPTF IKRSHREL IKRSWTHF IKRTRLCP IKRUPGR IKRUPGRL IKRUPGRU
IKRAPRD CALLS	IKRDATE IKRDIAGN IKRDQFR IKRGTHD IKRGTREC IKRLCMPR IKRPCREQ IKRQUEUE IKRREQFR IKRSHPRD IKRTRLCP
IKRAPTF CALLS	IKRCKPTF IKRDATE IKRDIAGN IKRDQFR IKRGTREC IKRMOD IKRPCREQ IKRQUEUE IKRSHPTF IKRSHREL IKRTRLCP
IKRARCH CALLS	IKRAAPAR IKRACOMP IKRAGEN IKRAPRD IKRAPTF IKRCLOSE IKROPHF IKRREWR
IKRBARDT CALLS	IKRBINS IKRGTREC IKRREWR
IKRBGRDT CALLS	IKRBINS IKRGTGEN IKRGTREC IKRREWR
IKRKBPR CALLS	IKRCKASS IKRCLOSE IKRCREAT IKRDIAGN IKREXIT IKRFRBUF IKRGTBUF IKRGTCMP IKRGTREC IKRLCMPR IKRLIBTA IKRLOAD IKRMOD IKRMARGIN IKROPHF IKRRDCGP IKRRDCPR IKRREWR IKRSALIB IKRSETCD IKRSETCS IKRSETFS IKRSETIM IKRSHPRD IKRSHREL IKRSLA IKRSPCLS IKRTRLCP IKRWTP
IKRBBKT CALLS	IKRLOAD IKRRASRV IKRSALIB
IKRBKUP CALLS	IKRCLOSE IKRCPIN IKRCVTLU IKRDIAGN IKROPBUF IKROPEN IKROPHF IKRSPCLS
IKRBPRDT CALLS	IKRBINS IKRGTREC IKRREWR
IKRCGT CALLS	IKRCKADI IKRCKASS IKROPEN
IKRCHECK CALLS	IKRDIAGN IKRFRBUF IKRGTBUF IKRGTREC IKRLSUP IKRSETIS IKRSHAPR IKRSHCMP IKRSHGRP IKRSHPRD IKRSHPTF IKRSHREL IKRWTP
IKRCKASS CALLS	IKRCVTLU
IKRCKCMP CALLS	IKRDIAGN IKREXIT IKRFIX IKRGTCMP IKRGTREC IKRSHREL
IKRCKCTL CALLS	IKRDIAGN IKRWTP
IKRCKFIX CALLS	IKRDIAGN IKRGTREC IKRLMLF IKRLOFI
IKRCKLV2 CALLS	IKRCKADI IKRSERR
IKRCKPTF CALLS	IKRREQFR

IKRCLOSE	CALLS	IKRSPCLS			
IKRCOPY	CALLS	IKRCKASS	IKRCLOSE	IKRCPIN	IKROPEN
IKRCPIN	CALLS	IKRCUTHF	IKREMU	IKRGTHD	IKRGTHDX
		IKRGTTAP	IKRMOD	IKRREWR	IKRWRSQ
IKRCREAT	CALLS	IKRFRBUF	IKRGTBUF	IKRGTFR	IKRGTHD
		IKRGTTAP	IKRMOD	IKRREWR	IKRWRSQ
IKRCUTHF	CALLS	IKRGTFR	IKRGTHDX	IKRGTTAP	IKRMOD
IKRDEBUG	CALLS	IKRCLOSE	IKRCUTHF	IKRGTTAP	IKRGTHD
IKRDEL	CALLS	IKRCLOSE	IKRDLIN	IKRGTTAP	IKRGTHDX
IKRDEQ	CALLS	IKRGTHD	IKRGTHDX	IKRGTTAP	IKRGTTAP
IKRDIAGN	CALLS	IKRSETCD	IKRSETGF	IKRSETGN	IKRSETLN
		IKRWTQ	IKRWTP	IKRSETLN	IKRWOR
IKRDLELE	CALLS	IKRDEQ	IKRFIX	IKRGTTAP	IKRGTTAP
IKRDLIN	CALLS	IKRDEQ	IKRDIAGN	IKRDLELE	IKRFIX
		IKRGTTAP	IKRGTTAP	IKRGTTAP	IKRGTTAP
		IKRSHGEN	IKRSHPRD	IKRSHREL	IKRSHREL
IKRDQFR	CALLS	IKRGTHD	IKRGTHDX	IKRGTTAP	IKRMOD
IKREAPL	CALLS	IKRAPPLY	IKRCKLV2	IKRDIAGN	IKRFRBUF
		IKRHDLV2	IKRPRCS	IKRRDLV2	IKRSERR
IKREARC	CALLS	IKRARCH	IKRCKCTL	IKRCKLV2	IKRDIAGN
		IKRGTTAP	IKRHDLV2	IKRPRCS	IKRRDLV2
IKREBCK	CALLS	IKRBKPR	IKRCKLV2	IKRCOPY	IKRDIAGN
		IKRPRCS	IKRRDLV2	IKRSERR	IKRHDLV2
IKRECOR	CALLS	IKRAAPAR	IKRBIO	IKRCHECK	IKRCKCMP
		IKRCLOSE	IKRDIAGN	IKRDLIN	IKRFRBUF
		IKRGTTAP	IKRGTTAP	IKRGTTAP	IKRGTTAP
		IKRLIBBL	IKRLIBCA	IKRLIBGA	IKRLMLF
		IKRLOAD	IKRMOD	IKROPHF	IKROPLNK
		IKRPRCS	IKRPTLNK	IKRRALNK	IKRRDCGN
		IKRREGCK	IKRREWR	IKRRLZAP	IKRSERR
		IKRSETCS	IKRSETDM	IKRSETFS	IKRSETGF
		IKRSETIM	IKRSHAPR	IKRSHGEN	IKRSHREL
		IKRTRLCP	IKRUIRP	IKRSHREL	IKRSLZAP
IKRECPY	CALLS	IKRCKLV2	IKRCOPY	IKRDIAGN	IKRHDLV2
		IKRRDLV2	IKRSERR	IKRHDLV2	IKRPRCS
IKRECRE	CALLS	IKRCKLV2	IKRCLOSE	IKRCREAT	IKRDIAGN
		IKROPHF	IKRPRCS	IKRRDLV2	IKRSERR
IKREDMP	CALLS	IKRCKLV2	IKRDEBUG	IKRDIAGN	IKRHDLV2
		IKRRDLV2	IKRSERR	IKRHDLV2	IKRPRCS
IKREINC	CALLS	IKRAPPLY	IKRCKLV2	IKRDIAGN	IKRFRBUF
		IKRPRCS	IKRRDLV2	IKRSERR	IKRHDLV2
IKRELKP	CALLS	IKRCKLV2	IKRDIAGN	IKRHDLV2	IKRLKUP
		IKRRDLV2	IKRSERR	IKRHDLV2	IKRPRCS
IKRELST	CALLS	IKRCLOSE	IKRDIAGN	IKRGWA	IKRHDLV2
		IKRPRCS	IKRRDLV2	IKRSETCS	IKRSETFS
IKREMBDY	CALLS	IKRACMP1	IKRACOMP	IKRAPRD	IKRCHECK
		IKRDATE	IKRDIAGN	IKRDLIN	IKRFIX
		IKRISLNK	IKRLIBBA	IKRLIBBL	IKRLIBTA
		IKRMOD	IKROPLNK	IKRPTLNK	IKRRALNK
		IKRRDCGN	IKRRDLV2	IKRRDNXT	IKRREWR
		IKRSAMNT	IKRSETCD	IKRSETCS	IKRSETFS
		IKRSETGN	IKRSETIM	IKRSHREL	IKRTRLCP
IKREMRG	CALLS	IKRCKLV2	IKRDIAGN	IKRHDLV2	IKRMERGE
		IKRRDLV2	IKRSERR	IKRHDLV2	IKRPRCS

		IKRRDLV2	IKRSERR			
IKREMU	CALLS	IKRCNVCP	IKRCNVFP	IKRCUTHF	IKRDATE	IKRDEQ
		IKRDEQUU	IKRDIAGN	IKRDLELE	IKRDLIN	IKRDQFR
		IKRFIX	IKRGTCMP	IKRGTRP	IKRGTHD	IKRGTTREC
		IKRLCMPR	IKRMOD	IKRQUEUE	IKRREQFR	IKRREWR
		IKRSHCMP	IKRSHPRD	IKRSHREL		
IKREPER	CALLS	IKRCKLV2	IKRCLOSE	IKRDIAGN	IKRHDLV2	IKROPHF
		IKRPERS	IKRPRCS	IKRRDLV2	IKRSERR	
IKRERDC	CALLS	IKRCKLV2	IKRCLOSE	IKRDIAGN	IKRHDLV2	IKROPHF
		IKRPRCS	IKRRDCE	IKRRDLV2	IKRSERR	IKRSETCS
		IKRSETFS	IKRSETIM			
IKRERMV	CALLS	IKRCKLV2	IKRDEL	IKRDIAGN	IKRHDLV2	IKRPRCS
		IKRRDLV2	IKRSERR			
IKRERST	CALLS	IKRCKLV2	IKRCOPY	IKRDIAGN	IKRHDLV2	IKRINSTL
		IKRPRCS	IKRRDLV2	IKRSERR	IKRSETCS	IKRSETFS
		IKRSETGF	IKRSETIM			
IKRERTC	CALLS	IKRCKLV2	IKRCLOSE	IKRDIAGN	IKRHDLV2	IKROPHF
		IKRPRCS	IKRRDLV2	IKRRTC	IKRSERR	
IKRERVK	CALLS	IKRAPPLY	IKRCKLV2	IKRDIAGN	IKRHDLV2	IKRPRCS
		IKRRDLV2	IKRSERR			
IKRESEL	CALLS	IKRCKASS	IKRCKLV2	IKRCLOSE	IKRDIAGN	IKRHDLV2
		IKROPHF	IKRPRCS	IKRRDLV2	IKRSAPL	IKRSERR
		IKRSETCD	IKRSETCS	IKRSETFS		
IKRESTL	CALLS	IKRCKLV2	IKRDIAGN	IKRGWA	IKRHDLV2	IKRINSTL
		IKRISVBK	IKRPRCS	IKRRDLV2	IKRSERR	IKRSETCS
		IKRSETFS	IKRSETGF	IKRSETIM	IKRUPGRL	
IKRETLR	CALLS	IKRCKCTL	IKRCKLV2	IKRDIAGN	IKRFRBUF	IKRGTTBUF
		IKRHDLV2	IKRPRCS	IKRRDLV2	IKRSERR	IKRTAIL
IKREUND	CALLS	IKRCKLV2	IKRCLOSE	IKRDIAGN	IKRHDLV2	IKROPHF
		IKRPRCS	IKRRDLV2	IKRREWR	IKRSERR	IKRUNDO
IKREXIT	CALLS	IKRCLOSE	IKRDIAGN	IKRREWR	IKRSPCLS	IKRWTO
		IKRWTP				
IKRGOGO	CALLS	IKRCNVCP	IKRSETCD	IKRSETDM	IKRSETGF	IKRSETGK
		IKRSETGN	IKRSETIM	IKRSETIS		
IKRGTAPR	CALLS	IKRGTTREC				
IKRGTTBUF	CALLS	IKROPBUF	IKRRWBUF			
IKRGTTCMP	CALLS	IKRGTTHD	IKRGTTREC			
IKRGTTFR	CALLS	IKRGTTHD	IKRGTTREC			
IKRGTTGEN	CALLS	IKRGTTREC				
IKRGTTGRP	CALLS	IKRGTTHD	IKRGTTREC			
IKRGTTHD	CALLS	IKRGTTREC				
IKRGTTIMT	CALLS	IKRGTTREC				
IKRGTTINF	CALLS	IKRGTTREC				
IKRGTTPTF	CALLS	IKRGTTREC				
IKRGTTREC	CALLS	IKRFIX	IKRGTTBUF	IKRNOTE		
IKRHDBKT	CALLS	IKRBIO				
IKRHDLV2	CALLS	IKRCNVFP	IKRCVTLU	IKRDIAGN	IKRGTTBUF	IKRHSLA
		IKRRDNXT	IKRSETCS	IKRSETIM	IKRWTO	IKRWTP
IKRHELP	CALLS	IKRDIAGN	IKRSETGF	IKRSETGN	IKRWTO	
IKRHSLA	CALLS	IKRSLA				
IKRIMSMT	CALLS	IKRGTTBUF				
IKRINMSH	CALLS	IKRIMSMT				
IKRINSTL	CALLS	IKRCHECK	IKRCKASS	IKRCLOSE	IKRCPIN	IKRDEQ
		IKRDIAGN	IKRDIAGS	IKRDLIN	IKRFRBUF	IKRGTTBUF

	IKRGTGMP	IKRGTGRP	IKRGTREC	IKRLCMPR	IKRLIBBL
	IKRLIBTA	IKRLOAD	IKRMOD	IKRMRGIN	IKROPEN
	IKROPHF	IKROPLNK	IKRPTLNK	IKRRALNK	IKRRDCE
	IKRRDCGN	IKRRDCGS	IKRREWR	IKRSALIB	IKRSETCD
	IKRSETCS	IKRSETDM	IKRSETFS	IKRSETGF	IKRSETGK
	IKRSETGN	IKRSETIM	IKRSETIS	IKRSHAPR	IKRSHPRD
	IKRSHPTF	IKRWTO	IKRWTP		
IKRISLNK	CALLS	IKRDIAGN	IKRLIBBC	IKRLIBBL	IKRLIBBS
		IKRLOAD			
		IKRMOD	IKROPLNK	IKRPTLNK	IKRRALNK
		IKRREWR			
		IKRSALIB	IKRSETCD	IKRSETCS	IKRSETDM
		IKRSETFS			
		IKRSETGF	IKRSETGK	IKRSETGN	IKRSETIM
		IKRSETIS			
		IKRSHPTF			
IKRISVBK	CALLS	IKRCGT	IKRCGTOP	IKRCKASS	IKRCKLV2
		IKRCLOSE			
		IKRCREAT	IKRDIAGN	IKRDLIN	IKREAPL
		IKRERVK			
		IKREXIT	IKRFRBUF	IKRGOGCP	IKRGOGXP
		IKRGTBUF			
		IKRGTGMP	IKRGTREC	IKRHDLV2	IKRLMOD
		IKRLOAD			
		IKRMOD	IKRMRGIN	IKROPHF	IKRPARSE
		IKRPRAPL			
		IKRPRCS	IKRRDIPT	IKRRDLV2	IKRREWR
		IKRRTC			
		IKRSALIB	IKRSETCD	IKRSETCS	IKRSETDM
		IKRSETFS			
		IKRSETGF	IKRSETGK	IKRSETGN	IKRSETIM
		IKRSETIS			
		IKRSHPTF	IKRSHREL	IKRSPCLS	IKRSWTHF
		IKRTRLCP			
		IKRUPGR	IKRUPGRL	IKRUPGRU	IKRWKF3C
		IKRWKF3O			
		IKRWKF3W	IKRWTP		
IKRLAPA	CALLS	IKRFIX	IKRGTREC	IKRSHREL	
IKRLCMPR	CALLS	IKRGTGRP	IKRGTREC		
IKRLGNRC	CALLS	IKRGTGEN	IKRGTREC	IKRSHREL	
IKRLIBST	CALLS	IKRDIAGN	IKRFRBUF	IKRGTBUF	IKRGWA
		IKRRDCGN			
		IKRSETCD	IKRSETCS	IKRSETDM	IKRSETFS
		IKRSETGF	IKRSETGK	IKRSETGN	IKRSETIM
		IKRSETIS			
		IKRSHPTF	IKRSHREL	IKRSPCLS	IKRSWTHF
		IKRTRLCP			
		IKRUPGR	IKRUPGRL	IKRUPGRU	IKRWKF3C
		IKRWKF3O			
		IKRWKF3W	IKRWTP		
IKRLINCM	CALLS	IKRGTGEN	IKRGTIMT	IKRGTREC	IKRSHREL
IKRLKMSH	CALLS	IKRLIBST			
IKRLKUP	CALLS	IKRCLOSE	IKRDATE	IKRDIAGN	IKRGTGMP
		IKRGTIMT			
		IKRGTINF	IKRGTREC	IKRLAPA	IKRLINCM
		IKRLMLF			
		IKRLMOD	IKRLOFI	IKRLSUP	IKROPHF
		IKRRDCGP			
		IKRREWR	IKRSHAPR	IKRSHCMP	IKRSHGEN
		IKRSHPRD			
		IKRSHPTF	IKRSHREL	IKRTRLCP	IKRWOF
IKRLMLF	CALLS	IKRFIX	IKRGTREC	IKRSHREL	
IKRLMOD	CALLS	IKRFIX	IKRGTREC	IKRSHREL	
IKRLOAD	CALLS	IKRDIAGN	IKRFRBUF	IKRGTBUF	IKROPEN
		IKRSPCLS			
		IKRWTP			
IKRLOFI	CALLS	IKRFRBUF	IKRGTBUF	IKRGTREC	
IKRLSUP	CALLS	IKRFIX	IKRGTREC	IKRSHREL	
IKRMAIN	CALLS	IKRDIAGN	IKRGTBUF	IKROPEN	IKRPARSE
		IKRPRCS			
		IKRRDIPT	IKRSPCLS	IKRSVCMD	IKRWTO
		IKRWTP			
IKRMERGE	CALLS	IKRCLOSE	IKRMRGIN	IKROPHF	
IKRMRGIN	CALLS	IKRAGEN	IKRDATE	IKRDIAGN	IKRDQFR
		IKRFIX			
		IKRFRBUF	IKRGTAPR	IKRGTBUF	IKRGTGMP
		IKRGTGEN			
		IKRGTGRP	IKRGTHTD	IKRGTPTF	IKRGTREC
		IKRIMSMT			
		IKRLSUP	IKRMOD	IKRQUEUE	IKRREQFR
		IKRREWR			
		IKRRTRVE	IKRSHAPR	IKRSHGEN	IKRSHPRD
		IKRSHPTF			
		IKRSHREL	IKRTRLCP		
IKROPEN	CALLS	IKRGTGMP			
IKROPHF	CALLS	IKRCKADI	IKRDIAGN	IKREMU	IKRNOTE
IKROPLNK	CALLS	IKRCKADI	IKRCKASS	IKROPEN	IKRPTLNK
		IKRSLA			

IKRPAPAL	CALLS	IKRDATE	IKRGTRC	IKRLOFI	IKRTRLCP	IKRWTP
IKRPARSE	CALLS	IKRDIAGN	IKRRDIPT	IKRSVCMD	IKRWTP	
IKRPCREQ	CALLS	IKRDQFR	IKRQUEUE			
IKRPDZAP	CALLS	IKRCKFIX	IKRCKLV2	IKRDIAGN	IKRFRBUF	IKRGTBUF
		IKRHDLV2	IKRLIBCM	IKRLIBDM	IKRLIBGR	IKRLIBPR
		IKRLOAD	IKRRDLV2	IKRSALIB	IKRSETCS	IKRSETFS
		IKRSETIM				
IKRPERS	CALLS	IKRDATE	IKRDQFR	IKRGTHD	IKRGTRC	IKRMOD
		IKRQUEUE	IKRREWR			
IKRPGENL	CALLS	IKRDATE	IKRFRBUF	IKRGTBUF	IKRGTRC	IKRRTRVE
		IKRTRLCP	IKRWTP			
IKRPIO	CALLS	IKRRDIPT	IKRWOR	IKRWTO		
IKRPPTFL	CALLS	IKRDATE	IKRGTRC	IKRTRLCP	IKRWTP	
IKRPRAPL	CALLS	IKRCGT	IKRDIAGN	IKREXIT	IKRFRBUF	IKRGTBUF
		IKRGWA	IKRWKF1C	IKRWKF10	IKRWKF1R	IKRWKF1W
		IKRWKF3C	IKRWKF30	IKRWKF3W		
IKRPRCS	CALLS	IKRWTP				
IKRQUEUE	CALLS	IKRDQFR	IKRGTRC	IKRMOD		
IKRRACSV	CALLS	IKRRCV	IKRWKF1W			
IKRRAESV	CALLS	IKRCKCTL	IKRRCV	IKRWKF4W		
IKRRALNK	CALLS	IKRCKCTL	IKRRCV			
IKRRASRV	CALLS	IKRBIO	IKRRCV	IKRWKF5W		
IKRRDCE	CALLS	IKRDEQ	IKRDIAGN	IKRDQFR	IKRFRBUF	IKRGTBUF
		IKRGTCMP	IKRGTRC	IKRQUEUE	IKRREQFR	IKRREWR
		IKRSETCD	IKRSETCS	IKRSETGF	IKRSETGN	IKRSETIM
		IKRSHPRD	IKRSHREL	IKRTRLCP		
IKRRDIPT	CALLS	IKRWKF3R				
IKRRDLV2	CALLS	IKRDIAGN	IKRHELP	IKRPARSE	IKRPRCS	IKRRDIPT
		IKRWTO	IKRWTP			
IKRRDNXT	CALLS	IKRDIAGN	IKRRDIPT			
IKRREGCK	CALLS	IKRDIAGN	IKRFRBUF	IKRGTBUF	IKRGTCMP	IKRGTRC
		IKRLGNRC	IKRLINCM	IKRMOD	IKRSETCS	IKRSETFS
		IKRSETGF	IKRSETGN	IKRSETIM	IKRSHGEN	IKRSHREL
		IKRTRLCP				
IKRREQFR	CALLS	IKRFRBUF	IKRGTBUF	IKRGTFR	IKRGTHD	IKRGTHDX
		IKRGTRC	IKRMOD	IKRNOTE	IKRREWR	
IKRREWR	CALLS	IKRRWBUF				
IKRRLZAP	CALLS	IKRCKFIX	IKRCKLV2	IKRDATE	IKRDIAGN	IKRHDLV2
		IKRLIBCM	IKRLIBDM	IKRLOAD	IKROPLNK	IKRPTLNK
		IKRRACSV	IKRRALNK	IKRRDLV2	IKRSALIB	IKRSETCS
		IKRSETFS	IKRSETIM	IKRSPCLS	IKRWKF1C	IKRWKF10
		IKRWKF1R	IKRWKF4C	IKRWKF40	IKRWKF4R	IKRWKF4W
IKRRL2	CALLS	IKRBINS	IKRDATE	IKRGTRC	IKRLOFI	IKRREWR
		IKRRL4	IKRSREPQ	IKRSRTMD	IKRTRLCP	IKRWTP
IKRRL3	CALLS	IKRBARDT	IKRBGRDT	IKRBPRDT	IKRDATE	IKRGTRC
		IKRPAPAL	IKRPGENL	IKRPPTFL	IKRRDCGP	IKRSHCMP
		IKRSHREL	IKRSREPQ	IKRTRLCP	IKRWTP	
IKRRL4	CALLS	IKRDATE	IKRGTRC	IKRRDCGP	IKRSREPQ	IKRTRLCP
		IKRWTP				
IKRRL5	CALLS	IKRBPRDT	IKRPPTFL	IKRSREPQ	IKRWTP	
IKRRL6	CALLS	IKRBARDT	IKRPAPAL	IKRSREPQ	IKRWTP	

IKRRL7	CALLS	IKRDATE	IKRGTRC	IKRLOFI	IKRSREPQ	IKRSRTMD
		IKRTRLCP	IKRWTP			
IKRRMGEN	CALLS	IKRDLIN	IKRFRBUF	IKRGTBUF		
IKRRTC	CALLS	IKRGTHD	IKRGTHDX	IKRGTRC	IKRRL2	IKRRL3
		IKRRL4	IKRRL5	IKRRL6	IKRRL7	IKRWTP
IKRRTRVE	CALLS	IKRGTBUF	IKRGTIMT	IKRGTRC	IKRSHGEN	IKRSHREL
IKRRWBUF	CALLS	IKRNOTE				
IKRSAASM	CALLS	IKRSEND	IKRWKF4C	IKRWKF40	IKRWKF4R	
IKRSALIB	CALLS	IKRLIBBM	IKRSEND	IKRSETCD	IKRSETGF	IKRSETGN
		IKRWKF4R				
IKRSAMNT	CALLS	IKRDIAGN	IKRRDNXT	IKRSEND	IKRSETIS	
IKRSAPL	CALLS	IKRCGT	IKRCGTPS	IKRDIAGN	IKRETLR	IKRFRBUF
		IKRGOGMB	IKRGTBUF	IKRPARSE	IKRPRCS	IKRRDIPT
		IKRSETCD	IKRSETGF	IKRSETGN	IKRTRLCP	IKRWKF3C
		IKRWKF30	IKRWKF3W	IKRWTP		
IKRSEND	CALLS	IKRGTMEN				
IKRSERR	CALLS	IKRDIAGN				
IKRSET	CALLS	IKRFRBUF	IKRGTBUF			
IKRSHAPR	CALLS	IKRFIX	IKRGTRC	IKRSHREL		
IKRSHCMP	CALLS	IKRFIX	IKRGTCMP	IKRGTRC		
IKRSHGEN	CALLS	IKRFIX	IKRGTRC	IKRSHREL		
IKRSHGRP	CALLS	IKRFIX	IKRGTRC	IKRSHREL		
IKRSHPRD	CALLS	IKRFIX	IKRGTRC	IKRSHREL		
IKRSHPTF	CALLS	IKRFIX	IKRGTRC	IKRSHREL		
IKRSHREL	CALLS	IKRFIX	IKRGTCMP	IKRGTRC		
IKRSLA	CALLS	IKRFRBUF	IKRGTBUF			
IKRSLZAP	CALLS	IKRBIO	IKRCKASS	IKRCKFIX	IKRCKLV2	IKRDATE
		IKRDEQ	IKRDIAGN	IKRDLIN	IKRFIX	IKRFRBUF
		IKRGTBUF	IKRGTRC	IKRGWA	IKRHDLV2	IKRLIBCM
		IKRLIBDM	IKRLMLF	IKRLMOD	IKRLOAD	IKRMOD
		IKROPEN	IKRRAESV	IKRRASRV	IKRRDLV2	IKRRDNXT
		IKRREWR	IKRSAASM	IKRSALIB	IKRSAMNT	IKRSETCS
		IKRSETFS	IKRSETGF	IKRSETGN	IKRSETIM	IKRSHAPR
		IKRWKF4C	IKRWKF40			
IKRSPCLS	CALLS	IKRFRBUF				
IKRSREPQ	CALLS	IKRDIAGN				
IKRSRTMD	CALLS	IKRBINS	IKRGTRC	IKRGTRC	IKRREWR	
IKRSTMTS	CALLS	IKRCGT	IKRCGTOP	IKRCGTPS	IKRDIAGN	IKRGOGO
		IKRSETGF	IKRSETGN	IKRSPCLS	IKRWTP	
IKRSWTHF	CALLS	IKRREWR				
IKRTAIL	CALLS	IKRAGEN	IKRCKADI	IKRCKCMP	IKRCLOSE	IKRDATE
		IKRDIAGN	IKRDLIN	IKRFRBUF	IKRGTBUF	IKRGTCMP
		IKRGTRC	IKRGNRC	IKRLIBBA	IKRLIBBL	
		IKRLMLF	IKRLMOD	IKRLOAD	IKRMOD	IKROPHF
		IKROPLNC	IKROPLNK	IKRPTLNK	IKRRAESV	IKRRALNK
		IKRRDCGN	IKRRDCTE	IKRREWR	IKRRMGEN	IKRRTRVE
		IKRSAASM	IKRSALIB	IKRSETCD	IKRSHAPR	IKRSHGEN
		IKRSHREL	IKRSPCLS	IKRTRLCP	IKRWKF4C	IKRWKF40
		IKRWKF4W				
IKRUIRP	CALLS	IKRGTBUF	IKRGTRC	IKRGTRC	IKRSETCS	IKRSETIM

IKRUNDO	CALLS	IKRAAPAR	IKRCKCMP	IKRDIAGN	IKRDLIN	IKRFRBUF
		IKRGTBUF	IKRGWA	IKRLIBBL	IKRLIBCA	IKRLIBGA
		IKRLMOD	IKRLOAD	IKRMOD	IKROPLNK	IKRPDZAP
		IKRPTLNK	IKRRALNK	IKRRDCGN	IKRRDLV2	IKRRDNXT
		IKRREGCK	IKRREWR	IKRRLZAP	IKRSERR	IKRSETCD
		IKRSETDM	IKRSETFS	IKRSETGF	IKRSETGN	IKRSHAPR
		IKRSHGEN	IKRSHREL	IKRSLZAP	IKRTRLCP	IKRUIRP
IKRUOMT	CALLS	IKRGTBUF				
IKRUPGR	CALLS	IKRBIO	IKRBKT	IKRCHECK	IKRCLOSE	IKRDIAGN
		IKRDLIN	IKRFIX	IKRFRBUF	IKRGTAPR	IKRGTBUF
		IKRGTCMP	IKRGTCMP	IKRGTCMP	IKRGTPTF	IKRGTTREC
		IKRHDBKT	IKRISLNK	IKRLIBBA	IKRLIBGA	IKRLIBTA
		IKRLMOD	IKRLSUP	IKRMOD	IKRMRGIN	IKROPEN
		IKRPHF	IKRRDCGN	IKRREGCK	IKRREWR	IKRRMGEN
		IKRRTC	IKRSETCD	IKRSETCS	IKRSETDM	IKRSETFS
		IKRSETGF	IKRSETGK	IKRSETGN	IKRSETIM	IKRSETIS
		IKRSHAPR	IKRSHGEN	IKRSHPTF	IKRSHREL	IKRSPCLS
		IKRSWTHF	IKRTRLCP	IKRWTP		
IKRWKF	CALLS	IKRCKADI	IKRGTBUF	IKROPEN		
IKRWOR	CALLS	IKRWTO				

Figure 12-3: Sorted by Called Modules

IKRAAPAR IS CALLED BY	IKRARCH IKRECOR IKRUNDO
IKRACMP1 IS CALLED BY	IKRACOMP IKREMBDY
IKRACOMP IS CALLED BY	IKRAPPLY IKRARCH IKREMBDY
IKRAGEN IS CALLED BY	IKRARCH IKRMRGIN IKRTAIL
IKRAPPLY IS CALLED BY	IKREAPL IKREINC IKRERVK
IKRAPRD IS CALLED BY	IKRARCH IKREMBDY
IKRAPTF IS CALLED BY	IKRAPPLY IKRARCH
IKRARCH IS CALLED BY	IKREARC
IKRBARDT IS CALLED BY	IKRRL3 IKRRL6
IKRBGRDT IS CALLED BY	IKRRL3
IKRBINS IS CALLED BY	IKRBARDT IKRBGRDT IKRBRPDT IKRRL2
IKRBIO IS CALLED BY	IKRSRTMD
IKRBKPR IS CALLED BY	IKRECOR IKRHDBKT IKRRASRV IKRSLZAP
IKRBKT IS CALLED BY	IKRUPGR
IKRBRPDT IS CALLED BY	IKREBCK
IKRCGT IS CALLED BY	IKRUPGR
IKRCGTOP IS CALLED BY	IKRRL3 IKRRL5
IKRCGTPS IS CALLED BY	IKRISVBK IKRPRAPL IKRSAPL IKRSTMTS
IKRCHECK IS CALLED BY	IKRISVBK IKRSTMTS
IKRCKADI IS CALLED BY	IKRSAPL IKRSTMTS
IKRCKASS IS CALLED BY	IKRECOR IKREMBDY IKRINSTL IKRUPGR
IKRCKCMP IS CALLED BY	IKRCGT IKRCKLV2 IKROPHF IKROPLNK
IKRCKCTL IS CALLED BY	IKRTAIL IKRWKF
IKRCKFIX IS CALLED BY	IKRBKPR IKRCGT IKRCOPY IKRESEL
IKRCKLV2 IS CALLED BY	IKRINSTL IKRISVBK IKROPLNK IKRSLZAP
IKRCKPTF IS CALLED BY	IKRAPPLY IKRECOR IKRTAIL IKRUNDO
IKRCLOSE IS CALLED BY	IKREARC IKRETLR IKRRAESV IKRRALNK
IKRCNVCP IS CALLED BY	IKRPDZAP IKRRLZAP IKRSLZAP
IKRCNVFP IS CALLED BY	IKREAPL IKREARC IKREBCK IKRECOR
IKRCOPY IS CALLED BY	IKRECPY IKRECRE IKREDMP IKREINC
IKRCPIN IS CALLED BY	IKRELKP IKREMRG IKREPER IKRERDC
IKRCREAT IS CALLED BY	IKRERMV IKRERST IKRERTC IKRERVK
IKRCUTHF IS CALLED BY	IKRESEL IKRESTL IKRETLR IKREUND
IKRCVTLU IS CALLED BY	IKRISVBK IKRPDZAP IKRRLZAP IKRSLZAP
IKRDATE IS CALLED BY	IKRAPTF
IKRCNVCP IS CALLED BY	IKRAPPLY IKRARCH IKRBKPR IKRBKUP
IKRCNVFP IS CALLED BY	IKRCOPY IKRDEBUG IKRDEL IKRECOR
IKRCOPY IS CALLED BY	IKRECRE IKRELST IKREPER IKRERDC
IKRCPIN IS CALLED BY	IKRERTC IKRESEL IKREUND IKREXIT
IKRCREAT IS CALLED BY	IKRINSTL IKRISVBK IKRLKUP IKRMERGE
IKRCUTHF IS CALLED BY	IKRTAIL IKRUPGR
IKRCVTLU IS CALLED BY	IKREMBDY IKREMU IKRGOGO
IKRDATE IS CALLED BY	IKREMU IKRHDLV2
IKRCNVCP IS CALLED BY	IKREBCK IKRECPY IKRERST
IKRCNVFP IS CALLED BY	IKRBKUP IKRCOPY IKRINSTL
IKRCOPY IS CALLED BY	IKRAPPLY IKRBKPR IKRECRE IKRISVBK
IKRCPIN IS CALLED BY	IKRCPIN IKRDEBUG IKREMU
IKRCREAT IS CALLED BY	IKRBKUP IKRCKASS IKRHDLV2
IKRCUTHF IS CALLED BY	IKRAAPAR IKRACOMP IKRAGEN IKRAPPLY
IKRCVTLU IS CALLED BY	IKRAPRD IKRAPTF IKREMBDY IKREMU
IKRDATE IS CALLED BY	IKRLKUP IKRMRGIN IKRPAPAL IKRPERS
	IKRPGENL IKRPPTFL IKRRLZAP IKRRL2

	IKRRL3	IKRRL4	IKRRL7	IKRSLZAP
	IKRTAIL			
IKRDEBUG IS CALLED BY	IKREDMP			
IKRDEL IS CALLED BY	IKRERMV			
IKRDEQ IS CALLED BY	IKRAGEN	IKRDLELE	IKRDLIN	IKREMU
	IKRINSTL	IKRRDCE	IKRSLZAP	
IKRDEQU IS CALLED BY	IKREMU			
IKRDIAGN IS CALLED BY	IKRACOMP	IKRAPPLY	IKRAPRD	IKRAPTF
	IKRBKPR	IKRBKUP	IKRCHECK	IKRCKCMP
	IKRCKCTL	IKRCKFIX	IKRDLIN	IKREAPL
	IKREARC	IKREBCK	IKRECOR	IKRECPY
	IKRECRE	IKREDMP	IKREINC	IKRELKP
	IKRELST	IKREMBDY	IKREMRG	IKREMU
	IKREPER	IKRERDC	IKRERMV	IKRERST
	IKRERTC	IKRERVK	IKRESEL	IKRESTL
	IKRETLR	IKREUND	IKREXIT	IKRHDLV2
	IKRHELP	IKRINSTL	IKRISLNK	IKRISVBK
	IKRLIBST	IKRLKUP	IKRLOAD	IKRMAIN
	IKRMRGIN	IKROPHF	IKRPARSE	IKRPDZAP
	IKRPRAPL	IKRRDCE	IKRRDLV2	IKRRDNXT
	IKRREGCK	IKRRLZAP	IKRSAMNT	IKRSAPL
	IKRSERR	IKRSLZAP	IKRSREPQ	IKRSTMTS
	IKRTAIL	IKRUNDO	IKRUPGR	
IKRDIAGS IS CALLED BY	IKRINSTL			
IKRDLELE IS CALLED BY	IKRDLIN	IKREMU		
IKRDLIN IS CALLED BY	IKRDEL	IKRECOR	IKREMBDY	IKREMU
	IKRINSTL	IKRISVBK	IKRRMGEN	IKRSLZAP
	IKRTAIL	IKRUNDO	IKRUPGR	
IKRDQFR IS CALLED BY	IKRAAPAR	IKRACOMP	IKRAGEN	IKRAPRD
	IKRAPTF	IKREMU	IKRMRGIN	IKRPREQ
	IKRPERS	IKRQUEUE	IKRRDCE	
IKREAPL IS CALLED BY	IKRISVBK			
IKREMBDA IS CALLED BY	IKRAPPLY			
IKREMBDY IS CALLED BY	IKRAPPLY			
IKREMU IS CALLED BY	IKRCPIN	IKROPHF		
IKRERVK IS CALLED BY	IKRISVBK			
IKRETLR IS CALLED BY	IKRSAPL			
IKREXIT IS CALLED BY	IKRAPPLY	IKRBKPR	IKRCKCMP	IKRISVBK
	IKRPRAPL			
IKRFIX IS CALLED BY	IKRAAPAR	IKRCKCMP	IKRDLELE	IKRDLIN
	IKREMBDY	IKREMU	IKRGTRC	IKRLAPA
	IKRLMLF	IKRLMOD	IKRLSUP	IKRMRGIN
	IKRSHAPR	IKRSHCMP	IKRSHGEN	IKRSHGRP
	IKRSHPRD	IKRSHPTF	IKRSHREL	IKRSLZAP
	IKRUPGR			
IKRFRBUF IS CALLED BY	IKRAAPAR	IKRAGEN	IKRAPPLY	IKRBKPR
	IKRCHECK	IKRCREAT	IKREAPL	IKREARC
	IKRECOR	IKREINC	IKRETLR	IKRINSTL
	IKRISVBK	IKRLIBST	IKRLOAD	IKRLOFI
	IKRMRGIN	IKRPDZAP	IKRPGENL	IKRPRAPL
	IKRRDCE	IKRREGCK	IKRREQFR	IKRRMGEN
	IKRSAPL	IKRSET	IKRSLA	IKRSLZAP
	IKRSPCLS	IKRTAIL	IKRUNDO	IKRUPGR
IKRGOGCP IS CALLED BY	IKRISVBK			

IKRGOGMB IS CALLED BY	IKRSAPL
IKRGOGO IS CALLED BY	IKRSTMTS
IKRGOGXP IS CALLED BY	IKRISVBK
IKRGTAPR IS CALLED BY	IKRMRGIN IKRUPGR
IKRGTBUF IS CALLED BY	IKRAAPAR IKRAGEN IKRAPPLY IKRBKPR
	IKRCHECK IKRCREAT IKREAPL IKREARC
	IKRECOR IKRETLR IKRGTREC IKRHDLV2
	IKRIMSMT IKRINSTL IKRISVBK IKRLIBST
	IKRLOAD IKRLOFI IKRMAIN IKRMRGIN
	IKRPDZAP IKRPGENL IKRPRAPL IKRRDCE
	IKRREGCK IKRREQFR IKRRMGEN IKRRTRVE
	IKRSAPL IKRSET IKRSLA IKRSLZAP
	IKRTAIL IKRUIRP IKRUNDO IKRUOMT
	IKRUPGR IKRWKF
IKRGTCMP IS CALLED BY	IKRBKPR IKRCKCMP IKRDLIN IKRECOR
	IKREMU IKRINSTL IKRISVBK IKRLKUP
	IKRMRGIN IKRRDCE IKRREGCK IKRSHCMP
	IKRSHREL IKRTAIL IKRUPGR
IKRGTFR IS CALLED BY	IKRCREAT IKRCUTHF IKRREQFR
IKRGTGEN IS CALLED BY	IKRBBGRDT IKRDLIN IKRLGNRC IKRLINCM
	IKRMRGIN IKRSRTMD IKRTAIL IKRUPGR
IKRGTGRP IS CALLED BY	IKRDLIN IKREMU IKRINSTL IKRLCMPR
	IKRMRGIN IKRSHGRP IKRSHPRD
IKRGTHD IS CALLED BY	IKRACOMP IKRAPRD IKRCPIN IKRCREAT
	IKRDEQ IKRDQFR IKREMU IKRGTCMP
	IKRGTFR IKRGTGRP IKRMRGIN IKRPERS
	IKRREQFR IKRRTC
IKRGTDX IS CALLED BY	IKRCPIN IKRCREAT IKRCUTHF IKRDEQ
	IKRDQFR IKRREQFR IKRRTC
IKRGTIMT IS CALLED BY	IKRAGEN IKRDLIN IKRLINCM IKRLKUP
	IKRRTRVE
IKRGTINF IS CALLED BY	IKRAPPLY IKRLKUP IKRUIRP IKRUPGR
IKRGTMEN IS CALLED BY	IKRSEND
IKRGTPTF IS CALLED BY	IKRMRGIN IKRUPGR
IKRGTREC IS CALLED BY	IKRAAPAR IKRAGEN IKRAPPLY IKRAPRD
	IKRAPTF IKRBARDT IKRBBGRDT IKRBKPR
	IKRBBPRDT IKRCHECK IKRCKCMP IKRCKFIX
	IKRCPIN IKRCREAT IKRCUTHF IKRDEBUG
	IKRDEQ IKRDLELE IKRDLIN IKRDQFR
	IKRECOR IKREMBDY IKREMU IKRGTAPR
	IKRGTCMP IKRGTFR IKRGTGEN IKRGTGRP
	IKRGTIMT IKRGTINF IKRGTHD IKRGTPTF
	IKRINSTL IKRISVBK IKRLAPA IKRLCMPR
	IKRLGNRC IKRLINCM IKRLKUP IKRLMLF
	IKRLMOD IKRLOFI IKRLSUP IKRMRGIN
	IKRPAPAL IKRPERS IKRPGENL IKRPPTFL
	IKRQUEUE IKRRDCE IKRREGCK IKRREQFR
	IKRRL2 IKRRL3 IKRRL4 IKRRL7
	IKRRTC IKRRTRVE IKRSHAPR IKRSHCMP
	IKRSHGEN IKRSHGRP IKRSHPRD IKRSHPTF
	IKRSHREL IKRSLZAP IKRSRTMD IKRTAIL
	IKRUIRP IKRUPGR
IKRGTAP IS CALLED BY	IKRCPIN IKROPEN
IKRGTWA IS CALLED BY	IKRECOR IKRELST IKRESTL IKRLIBST

IKRHDBKT IS CALLED BY	IKRPRAPL	IKRSLZAP	IKRUNDO	
IKRHDLV2 IS CALLED BY	IKRUPGR			
	IKREAPL	IKREARC	IKREBCK	IKRECOR
	IKRECPY	IKRECRE	IKREDMP	IKREINC
	IKRELKP	IKRELST	IKREMRG	IKREPER
	IKRERDC	IKRERMV	IKRERST	IKRERTC
	IKRERVK	IKRESEL	IKRESTL	IKRETLR
	IKREUND	IKRISVBK	IKRPDZAP	IKRRLZAP
	IKRSLZAP			
IKRHELP IS CALLED BY	IKRRDLV2			
IKRHSLA IS CALLED BY	IKRHDLV2			
IKRIMSMT IS CALLED BY	IKRINMSH	IKMRGIN		
IKRINSTL IS CALLED BY	IKRERST	IKRESTL		
IKRISLNK IS CALLED BY	IKREMBDY	IKRUPGR		
IKRISVBK IS CALLED BY	IKRELST	IKRESTL		
IKRLAPA IS CALLED BY	IKRECOR	IKRLKUP		
IKRLCMPR IS CALLED BY	IKRAPRD	IKRBKPR	IKREMU	IKRINSTL
IKRLGNRC IS CALLED BY	IKRREGCK	IKRTAIL		
IKRLIBBA IS CALLED BY	IKREMBDY	IKRTAIL	IKRUPGR	
IKRLIBBC IS CALLED BY	IKRISLNK			
IKRLIBBL IS CALLED BY	IKRECOR	IKREMBDY	IKRINSTL	IKRISLNK
	IKRTAIL	IKRUNDO		
IKRLIBBM IS CALLED BY	IKRSALIB			
IKRLIBBS IS CALLED BY	IKRISLNK			
IKRLIBCA IS CALLED BY	IKRECOR	IKRUNDO		
IKRLIBCM IS CALLED BY	IKRPDZAP	IKRRLZAP	IKRSLZAP	
IKRLIBDM IS CALLED BY	IKRPDZAP	IKRRLZAP	IKRSLZAP	
IKRLIBGA IS CALLED BY	IKRECOR	IKRUNDO	IKRUPGR	
IKRLIBGR IS CALLED BY	IKRPDZAP			
IKRLIBPR IS CALLED BY	IKRPDZAP			
IKRLIBST IS CALLED BY	IKRLKMSH			
IKRLIBTA IS CALLED BY	IKRBKPR	IKREMBDY	IKRINSTL	IKRUPGR
IKRLINCM IS CALLED BY	IKRLKUP	IKRREGCK		
IKRLKUP IS CALLED BY	IKRELKP			
IKRLMLF IS CALLED BY	IKRCKFIX	IKRECOR	IKRLKUP	IKRSLZAP
	IKRTAIL			
IKRLMOD IS CALLED BY	IKRECOR	IKRISVBK	IKRLKUP	IKRSLZAP
	IKRTAIL	IKRUNDO	IKRUPGR	
IKRLOAD IS CALLED BY	IKRBKPR	IKRBKT	IKRECOR	IKREMBDY
	IKRINSTL	IKRISLNK	IKRISVBK	IKRPDZAP
	IKRRLZAP	IKRSLZAP	IKRTAIL	IKRUNDO
IKRLOFI IS CALLED BY	IKRCKFIX	IKRLKUP	IKRPAPAL	IKRRL2
	IKRRL7			
IKRLSUP IS CALLED BY	IKRCHECK	IKRLKUP	IKMRGIN	IKRUPGR
IKRMERGE IS CALLED BY	IKREMRG			
IKRMOD IS CALLED BY	IKRAPPLY	IKRAPTF	IKRBKPR	IKRCPIN
	IKRCREAT	IKRCUTHF	IKRDEQ	IKRDQFR
	IKRECOR	IKREMBDY	IKREMU	IKRINSTL
	IKRISLNK	IKRISVBK	IKMRGIN	IKRPERS
	IKRQUEUE	IKRREGCK	IKRREQFR	IKRSLZAP
	IKRTAIL	IKRUNDO	IKRUPGR	
IKMRGIN IS CALLED BY	IKRAPPLY	IKRBKPR	IKRINSTL	IKRISVBK

IKRNOTE IS CALLED BY	IKRMERGE	IKRUPGR		
IKROPBUF IS CALLED BY	IKRGTRC	IKROPHF	IKRREQFR	IKRRWBUF
IKROPEN IS CALLED BY	IKRBKUP	IKRGTBUF		
	IKRBKUP	IKRCGT	IKRCOPY	IKRINSTL
	IKRLOAD	IKRMAIN	IKROPHF	IKROPLNK
	IKRSLZAP	IKRUPGR	IKRWKF	
IKROPHF IS CALLED BY	IKRAPPLY	IKRARCH	IKRBKPR	IKRBKUP
	IKRCOPY	IKRDEBUG	IKRDEL	IKRECOR
	IKRECRE	IKREPER	IKRERDC	IKRERTC
	IKRESEL	IKREUND	IKRINSTL	IKRISVBK
	IKRLKUP	IKRMERGE	IKRTAIL	IKRUPGR
	IKRTAIL			
IKROPLNC IS CALLED BY	IKRECOR	IKREMBDY	IKRINSTL	IKRISLNK
IKROPLNK IS CALLED BY	IKRRLZAP	IKRTAIL	IKRUNDO	
	IKRRL3	IKRRL6		
IKRPAPAL IS CALLED BY	IKRISVBK	IKRMAIN	IKRRDLV2	IKRSAPL
IKRPARSE IS CALLED BY	IKRAAPAR	IKRACOMP	IKRAPRD	IKRAPTF
IKRPCREQ IS CALLED BY	IKRECOR	IKRUNDO		
IKRPDZAP IS CALLED BY	IKREPER			
IKRPERS IS CALLED BY	IKRRL3			
IKRPGENL IS CALLED BY	IKRRL3	IKRRL5		
IKRPPTFL IS CALLED BY	IKRISVBK			
IKRPRAPL IS CALLED BY	IKREAPL	IKREARC	IKREBCK	IKRECOR
IKRPRCS IS CALLED BY	IKRECPY	IKRECRE	IKREDMP	IKREINC
	IKRELKP	IKRELST	IKREMRG	IKREPER
	IKRERDC	IKRERMV	IKRERST	IKRERTC
	IKRERVK	IKRESEL	IKRESTL	IKRETLR
	IKREUND	IKRISVBK	IKRMAIN	IKRRDLV2
	IKRSAPL			
IKRPTLNK IS CALLED BY	IKRECOR	IKREMBDY	IKRINSTL	IKRISLNK
	IKROPLNK	IKRRLZAP	IKRTAIL	IKRUNDO
IKRQUEUE IS CALLED BY	IKRAAPAR	IKRACOMP	IKRAGEN	IKRAPRD
	IKRAPTF	IKRDEQ	IKREMU	IKMRGIN
	IKRPCREQ	IKRPERS	IKRRDCE	
	IKRRLZAP			
IKRRACSV IS CALLED BY	IKRSLZAP	IKRTAIL		
IKRRAESV IS CALLED BY	IKRECOR	IKREMBDY	IKRINSTL	IKRISLNK
IKRRALNK IS CALLED BY	IKRRLZAP	IKRTAIL	IKRUNDO	
	IKRBKT	IKRSLZAP		
IKRRASRV IS CALLED BY	IKRRACSV	IKRRAESV	IKRRALNK	IKRRASRV
IKRRCV IS CALLED BY	IKREMBDY	IKRERDC	IKRINSTL	
IKRRDCE IS CALLED BY	IKRECOR	IKREMBDY	IKRINSTL	IKRLIBST
IKRRDCGN IS CALLED BY	IKRTAIL	IKRUNDO	IKRUPGR	
	IKRBKPR	IKRLKUP	IKRRL3	IKRRL4
IKRRDCGP IS CALLED BY	IKRINSTL			
IKRRDCGS IS CALLED BY	IKRBKPR			
IKRRDCPR IS CALLED BY	IKRTAIL			
IKRRDCTE IS CALLED BY	IKRISVBK	IKRMAIN	IKRPARSE	IKRPIO
IKRRDIPT IS CALLED BY	IKRRDLV2	IKRRDNXT	IKRSAPL	
	IKREAPL	IKREARC	IKREBCK	IKRECOR
IKRRDLV2 IS CALLED BY	IKRECPY	IKRECRE	IKREDMP	IKREINC
	IKRELKP	IKRELST	IKREMBDY	IKREMRG
	IKREPER	IKRERDC	IKRERMV	IKRERST
	IKRERTC	IKRERVK	IKRESEL	IKRESTL

	IKRETLR	IKREUND	IKRISVBK	IKRPDZAP
	IKRRLZAP	IKRSLZAP	IKRUNDO	
IKRRDNXT IS CALLED BY	IKRAPPLY	IKREMBDY	IKRHDLV2	IKRSAMNT
	IKRSLZAP	IKRUNDO		
IKRREGCK IS CALLED BY	IKRECOR	IKRUNDO	IKRUPGR	
IKRREQFR IS CALLED BY	IKRAAPAR	IKRACMP1	IKRAGEN	IKRAPRD
	IKRCKPTF	IKREMU	IKRMRGIN	IKRRDCE
IKRREWR IS CALLED BY	IKRAAPAR	IKRAGEN	IKRAPPLY	IKRARCH
	IKRBARDT	IKRBGRDT	IKRBKPR	IKRBRPDT
	IKRCPIN	IKRCREAT	IKRDEL	IKRDLIN
	IKRECOR	IKREMBDY	IKREMU	IKREUND
	IKREXIT	IKRINSTL	IKRISLNK	IKRISVBK
	IKRLKUP	IKRMRGIN	IKRPERS	IKRRDCE
	IKRREQFR	IKRRL2	IKRSLZAP	IKRSRTMD
	IKRSWTHF	IKRTAIL	IKRUNDO	IKRUPGR
IKRRLZAP IS CALLED BY	IKRECOR	IKRUNDO		
IKRRL2 IS CALLED BY	IKRRTC			
IKRRL3 IS CALLED BY	IKRRTC			
IKRRL4 IS CALLED BY	IKRRL2	IKRRTC		
IKRRL5 IS CALLED BY	IKRRTC			
IKRRL6 IS CALLED BY	IKRRTC			
IKRRL7 IS CALLED BY	IKRRTC			
IKRRMGEN IS CALLED BY	IKRTAIL	IKRUPGR		
IKRRTC IS CALLED BY	IKRRTC	IKRISVBK	IKRUPGR	
IKRRTRVE IS CALLED BY	IKRAGEN	IKRMRGIN	IKRPGENL	IKRTAIL
IKRRWBUF IS CALLED BY	IKRGTBUF	IKRREWR		
IKRSAASM IS CALLED BY	IKRSLZAP	IKRTAIL		
IKRSALIB IS CALLED BY	IKRBKPR	IKRBKT	IKREMBDY	IKRINSTL
	IKRISLNK	IKRISVBK	IKRPDZAP	IKRRLZAP
	IKRSLZAP	IKRTAIL		
IKRSAMNT IS CALLED BY	IKREMBDY	IKRSLZAP		
IKRSAPL IS CALLED BY	IKRESEL			
IKRSEND IS CALLED BY	IKRSAASM	IKRSALIB	IKRSAMNT	
IKRSERR IS CALLED BY	IKRCKLV2	IKREAPL	IKREARC	IKREBCK
	IKRECOR	IKRECPY	IKRECRE	IKREDMP
	IKREINC	IKRELKP	IKREMRG	IKREPER
	IKRERDC	IKRERMV	IKRERST	IKRERTC
	IKRERVK	IKRESEL	IKRESTL	IKRETLR
	IKREUND	IKRUNDO		
IKRSETCD IS CALLED BY	IKRAPPLY	IKRBKPR	IKRDIAGN	IKRECOR
	IKREMBDY	IKRESEL	IKRGOGO	IKRINSTL
	IKRISLNK	IKRISVBK	IKRLIBST	IKRRDCE
	IKRSALIB	IKRSAPL	IKRTAIL	IKRUNDO
	IKRUPGR			
IKRSETCS IS CALLED BY	IKRAPPLY	IKRBKPR	IKRECOR	IKRELST
	IKREMBDY	IKRERDC	IKRERST	IKRESEL
	IKRESTL	IKRHDLV2	IKRINSTL	IKRISLNK
	IKRISVBK	IKRLIBST	IKRPDZAP	IKRRDCE
	IKRREGCK	IKRRLZAP	IKRSLZAP	IKRUIRP
	IKRUPGR			
IKRSETDM IS CALLED BY	IKRECOR	IKRGOGO	IKRINSTL	IKRISLNK
	IKRISVBK	IKRLIBST	IKRUNDO	IKRUPGR
IKRSETFS IS CALLED BY	IKRAPPLY	IKRBKPR	IKRECOR	IKRELST
	IKREMBDY	IKRERDC	IKRERST	IKRESEL

	IKRESTL	IKRINSTL	IKRISLNK	IKRISVBK
	IKRLIBST	IKRPDZAP	IKRREGCK	IKRRLZAP
	IKRSLZAP	IKRUNDO	IKRUPGR	
IKRSETGF IS CALLED BY	IKRAPPLY	IKRDIAGN	IKRECOR	IKREMBDY
	IKRERST	IKRESTL	IKRGOGO	IKRHELP
	IKRINSTL	IKRISLNK	IKRISVBK	IKRLIBST
	IKRRDCE	IKRREGCK	IKRSALIB	IKRSAPL
	IKRSLZAP	IKRSTMTS	IKRUNDO	IKRUPGR
IKRSETGK IS CALLED BY	IKRGOGO	IKRINSTL	IKRISLNK	IKRISVBK
	IKRUPGR			
IKRSETGN IS CALLED BY	IKRAPPLY	IKRDIAGN	IKRECOR	IKREMBDY
	IKRGOGO	IKRHELP	IKRINSTL	IKRISLNK
	IKRISVBK	IKRLIBST	IKRRDCE	IKRREGCK
	IKRSALIB	IKRSAPL	IKRSLZAP	IKRSTMTS
	IKRUNDO	IKRUPGR		
IKRSETIM IS CALLED BY	IKRAPPLY	IKRDKPR	IKRECOR	IKREMBDY
	IKRERDC	IKRERST	IKRESTL	IKRGOGO
	IKRHDLV2	IKRINSTL	IKRISLNK	IKRISVBK
	IKRLIBST	IKRPDZAP	IKRRDCE	IKRREGCK
	IKRRLZAP	IKRSLZAP	IKRUIRP	IKRUPGR
IKRSETIS IS CALLED BY	IKRCHECK	IKRGOGO	IKRINSTL	IKRISLNK
	IKRISVBK	IKRSAMNT	IKRUPGR	
IKRSETLN IS CALLED BY	IKRDIAGN			
IKRSHAPR IS CALLED BY	IKRAAPAR	IKRCHECK	IKRECOR	IKRINSTL
	IKRLKUP	IKRMARGIN	IKRSLZAP	IKRTAIL
	IKRUNDO	IKRUPGR		
IKRSHCMP IS CALLED BY	IKRAPPLY	IKRCHECK	IKREMU	IKRLKUP
	IKRRL3			
IKRSHGEN IS CALLED BY	IKRAGEN	IKRDLIN	IKRECOR	IKRLKUP
	IKRMARGIN	IKRREGCK	IKRRTRVE	IKRTAIL
	IKRUNDO	IKRUPGR		
IKRSHGRP IS CALLED BY	IKRCHECK			
IKRSHPRD IS CALLED BY	IKRAPRD	IKRDKPR	IKRCHECK	IKRDLIN
	IKREMU	IKRINSTL	IKRLKUP	IKRMARGIN
	IKRRDCE			
IKRSHPTF IS CALLED BY	IKRAPPLY	IKRAPTF	IKRCHECK	IKRINSTL
	IKRISLNK	IKRISVBK	IKRLKUP	IKRMARGIN
	IKRUPGR			
IKRSHREL IS CALLED BY	IKRAAPAR	IKRACOMP	IKRAGEN	IKRAPPLY
	IKRAPTF	IKRDKPR	IKRCHECK	IKRCKCMP
	IKRDLIN	IKRECOR	IKREMBDY	IKREMU
	IKRISVBK	IKRLAPA	IKRLGNRC	IKRLINCM
	IKRLKUP	IKRLMLF	IKRLMOD	IKRLSUP
	IKRMARGIN	IKRRDCE	IKRREGCK	IKRRL3
	IKRRTRVE	IKRSHAPR	IKRSHGEN	IKRSHPTF
	IKRTAIL	IKRUNDO	IKRUPGR	
IKRSLA IS CALLED BY	IKRDKPR	IKRHSLA	IKROPLNK	
IKRSLZAP IS CALLED BY	IKRECOR	IKRUNDO		
IKRSPCLS IS CALLED BY	IKRDKPR	IKRDKUP	IKRCLOSE	IKREXIT
	IKRISVBK	IKRLOAD	IKRMAIN	IKRRLZAP
	IKRSTMTS	IKRTAIL	IKRUPGR	
IKRSREPQ IS CALLED BY	IKRRL2	IKRRL3	IKRRL4	IKRRL5
	IKRRL6	IKRRL7		
IKRSRTMD IS CALLED BY	IKRRL2	IKRRL7		

IKRSTMTS	IS CALLED BY	IKRELST	
IKRSVCMD	IS CALLED BY	IKRMAIN	IKRPARSE
IKRSWTHF	IS CALLED BY	IKRAPPLY	IKRISVBK IKRUPGR
IKRTAIL	IS CALLED BY	IKRETLR	
IKRTRLCP	IS CALLED BY	IKRAAPAR	IKRACOMP IKRAPPLY IKRAPRD
		IKRAPTF	IKRBKPR IKRECOR IKREMBDY
		IKRISVBK	IKRLKUP IKRMRGIN IKRPAPAL
		IKRPGENL	IKRPPTFL IKRRDCE IKRREGCK
		IKRRL2	IKRRL3 IKRRL4 IKRRL7
		IKRSAPL	IKRTAIL IKRUNDO IKRUPGR
		IKRECOR	IKRUNDO
		IKREUND	
		IKRAGEN	
IKRUIRP	IS CALLED BY	IKRAPPLY	IKRISVBK
IKRUNDO	IS CALLED BY	IKRAPPLY	IKRESTL IKRISVBK
IKRUOMT	IS CALLED BY	IKRAPPLY	IKRISVBK
IKRUPGR	IS CALLED BY	IKRPRAPL	IKRRLZAP
IKRUPGRL	IS CALLED BY	IKRPRAPL	IKRRLZAP
IKRUPGRU	IS CALLED BY	IKRPRAPL	IKRRLZAP
IKRWKF1C	IS CALLED BY	IKRPRAPL	IKRRACSV
IKRWKF10	IS CALLED BY	IKRISVBK	IKRPRAPL IKRSAPL
IKRWKF1R	IS CALLED BY	IKRISVBK	IKRPRAPL IKRSAPL
IKRWKF1W	IS CALLED BY	IKRRDIPT	
IKRWKF3C	IS CALLED BY	IKRISVBK	IKRPRAPL IKRSAPL
IKRWKF30	IS CALLED BY	IKRRLZAP	IKRSAASM IKRSLZAP IKRTAIL
IKRWKF3R	IS CALLED BY	IKRRLZAP	IKRSAASM IKRSLZAP IKRTAIL
IKRWKF3W	IS CALLED BY	IKRRLZAP	IKRSAASM IKRSALIB
IKRWKF4C	IS CALLED BY	IKRRAESV	IKRRLZAP IKRTAIL
IKRWKF40	IS CALLED BY	IKRRASRV	
IKRWKF4R	IS CALLED BY	IKRLKUP	
IKRWKF4W	IS CALLED BY	IKRDIAGN	IKRPIO
IKRWKF5W	IS CALLED BY	IKRCPIN	IKRCREAT
IKRWOF	IS CALLED BY	IKRDIAGN	IKREXIT IKRHDLV2 IKRHELP
IKRWOR	IS CALLED BY	IKRINSTL	IKRLIBST IKRMAIN IKRPIO
IKRWRSQ	IS CALLED BY	IKRRDLV2	IKRWOR
IKRWTO	IS CALLED BY	IKRBKPR	IKRCHECK IKRCKCTL IKRDEBUG
		IKRDIAGN	IKREXIT IKRHDLV2 IKRINSTL
		IKRISVBK	IKRLIBST IKRLOAD IKRMAIN
		IKRPAPAL	IKRPARSE IKRPGENL IKRPPTFL
		IKRPRCS	IKRRDLV2 IKRRL2 IKRRL3
		IKRRL4	IKRRL5 IKRRL6 IKRRL7
		IKRRTC	IKRSAPL IKRSTMTS IKRUPGR

Module-Macro Interrelations

Figure 12-4: Sorted by Invoking Modules

IKRAAPAR	INVOKES	IKRAPA	IKRCMD	IKRCOM	IKRINS	IKRMIA
		IKRMIP	IKRPRE	IKRPROC	IKRQUIT	IKRTYPE
		IKRXENT				
IKRACMP1	INVOKES	IKRPRE	IKRPROC	IKRTYPE	IKRXENT	
IKRACOMP	INVOKES	IKRINS	IKRPROC	IKRTYPE	IKRXENT	
IKRAGEN	INVOKES	IKRCMD	IKRCOM	IKRINS	IKRPROC	IKRQUIT
		IKRTYPE	IKRXENT	IKRIMT	IKRIMR	IKRGEN
IKRAPPLY	INVOKES	IKRAIP	IKRCMD	IKRINS	IKRMIP	IKRNSTAT
		IKRPRE	IKRPROC	IKRPTF	IKRSUP	IKRTYPE
		IKRXCUM	IKRXENT	IKRXMRG		
IKRAPRD	INVOKES	IKRCOF	IKRCOM	IKRMIP	IKRPRD	IKRPRE
		IKRPROC	IKRQUIT	IKRTYPE	IKRXENT	
IKRAPTF	INVOKES	IKRAIP	IKRCMD	IKRCOM	IKRINS	IKRMIP
		IKRPROC	IKRPTF	IKRQUIT	IKRSUP	IKRTYPE
		IKRXENT				
IKRARCH	INVOKES	IKRPROC	IKRTYPE	IKRXENT		
IKRBARDT	INVOKES	IKRAIP	IKRAPA	IKRINS	IKRPROC	IKRPTF
		IKRTYPE				
IKRBGRDT	INVOKES	IKRINS	IKRPROC	IKRTYPE	IKRGEN	
IKRBINS	INVOKES	IKRPROC				
IKRBIO	INVOKES	IKRCMD	IKRPROC			
IKRBKPR	INVOKES	IKRCOF	IKRINS	IKRLIB	IKRLSI	IKRPRD
		IKRPROC	IKRQUIT	IKRTYPE	IKRXCPY	IKRXCUM
		IKRXMRG				
IKRBKT	INVOKES	IKRCMD	IKRLSI	IKRPROC	IKRTYPE	
IKRBKUP	INVOKES	IKRCMD	IKRNSTAT	IKRPROC	IKRQUIT	IKRXCPY
IKRBPRDT	INVOKES	IKRINS	IKRPROC	IKRPTF	IKRSUP	IKRTYPE
IKRCGT	INVOKES	IKRCMD	IKRDTFBS	IKRDTFSD	IKRPROC	IKRQUIT
		IKRTYPE	IKRXCUM			
IKRCHECK	INVOKES	IKRAPA	IKRCMD	IKRINS	IKRPROC	IKRPTF
		IKRQUIT	IKRTYPE	IKRXENT	IKRXSTL	
IKRCKASS	INVOKES	IKRCMD	IKRPROC	IKRQUIT		
IKRCKCMP	INVOKES	IKRINS	IKRNSTAT	IKRPROC	IKRXENT	
IKRCKCTL	INVOKES	IKRPROC	IKRTYPE	IKRXENT		
IKRCKD	INVOKES	IKRCMD	IKRPROC	IKRQUIT	IKRXCPY	
IKRCKFIX	INVOKES	IKRAPA	IKRCMD	IKRMIA	IKRPROC	IKRTYPE
		IKRXENT				
IKRCKLV2	INVOKES	IKR#AFF	IKR#ALT	IKR#CIS	IKR#CPT	IKR#DEL
		IKR#DFN	IKR#ICL	IKR#IST	IKR#RPL	IKR#RQR
		IKR#RSV	IKR#SCN	IKR#VER	IKRCMD	IKRPROC
		IKRTYPE	IKRXCUM	IKRXENT	IKRXSTL	IKR#INF
		IKR#EXE				
IKRCKPTF	INVOKES	IKRAIP	IKRCOM	IKRMIP	IKRPRE	IKRPROC
		IKRSUP	IKRTYPE	IKRXENT		
IKRCLMSH	INVOKES	IKRDTF	IKRIFCTB	IKRPROC	IKRQUIT	
IKRCLOSE	INVOKES	IKRBFRO	IKRCMD	IKRPROC		
IKRCNVRT	INVOKES	IKRPROC				
IKRCOMR	INVOKES	IKRCOMRX	IKRPROC			
IKRCOPY	INVOKES	IKRPROC	IKRXCPY			

IKRCPC	INVOKES	IKRLCPCB	IKRPROC			
IKRCPIN	INVOKES	IKRFRE	IKRHD1	IKRHDX	IKRPROC	IKRQUIT
IKRCREAT	INVOKES	IKRFRE	IKRHD1	IKRHD2	IKRHD3	IKRHDX
		IKRPROC	IKRXCRE			
IKRCUTHF	INVOKES	IKRFRE	IKRHDX	IKRPROC		
IKRCVTLU	INVOKES	IKRPROC				
IKRDATE	INVOKES	IKRPROC				
IKRDEBUG	INVOKES	IKRPROC	IKRXDMP			
IKRDEL	INVOKES	IKRPROC				
IKRDEQ	INVOKES	IKRFRE	IKRHD3	IKRHDX	IKRPROC	IKRQUIT
IKRDIAGN	INVOKES	IKREBC	IKRNSTAT	IKRPROC	IKRQUIT	
IKRDLELE	INVOKES	IKRAPA	IKRCOF	IKRCOM	IKRMIA	IKRMIP
		IKRPROC				
IKRDLIN	INVOKES	IKRAPA	IKRCMD	IKRCOF	IKRHD1	IKRINS
		IKRLIB	IKRMIP	IKRPRD	IKRPRE	IKRPROC
		IKRPTF	IKRQUIT	IKRTYPE	IKRXRMV	IKRINX
		IKRIMT	IKRIMR	IKRGEN		
IKRDQFR	INVOKES	IKRFRE	IKRHD1	IKRHD3	IKRHDX	IKRPROC
IKREAPL	INVOKES	IKR#APL	IKRCMD	IKRPROC	IKRQUIT	IKRTYPE
		IKRXENT				
IKREARC	INVOKES	IKR#ARC	IKRCMD	IKRPROC	IKRTYPE	IKRXENT
IKREBCK	INVOKES	IKR#BCK	IKRCMD	IKRPROC	IKRTYPE	IKRXCPY
IKRECOR	INVOKES	IKR#AFF	IKR#COR	IKR#DFN	IKR#IVL	IKRAPA
		IKRCMD	IKRINS	IKRMIP	IKRNSTAT	IKRPRE
		IKRPROC	IKRPTF	IKRQUIT	IKRTYPE	IKRXENT
		IKRXRMV	IKRGEN			
IKRECPY	INVOKES	IKR#CPY	IKRCMD	IKRPROC	IKRXCPY	
IKRECRE	INVOKES	IKR#CRE	IKRCMD	IKRPROC	IKRXCRE	
IKREDMP	INVOKES	IKR#DMP	IKRCMD	IKRPROC	IKRXDMP	
IKREINC	INVOKES	IKR#INC	IKRCMD	IKRNSTAT	IKRPROC	IKRTYPE
		IKRXENT				
IKRELKP	INVOKES	IKR#LKP	IKRCMD	IKRPROC	IKRTYPE	IKRXLKP
IKRELST	INVOKES	IKR#LIST	IKRCMD	IKRPROC	IKRQUIT	IKRTYPE
		IKRXCUM				
IKREMBDY	INVOKES	IKRCMD	IKRINS	IKRLSI	IKRNSTAT	IKRPROC
		IKRQUIT	IKRTYPE	IKRXENT	IKRXRDC	IKRXRMV
IKREMRG	INVOKES	IKR#MRG	IKRCMD	IKRPROC	IKRXMRG	
IKREMU	INVOKES	IKRAPA	IKRCMD	IKRCOF	IKRCOM	IKRFRE
		IKRHD1	IKRHD3	IKRHDX	IKRINS	IKRLIB
		IKRMIA	IKRMIP	IKRPRD	IKRPRE	IKRPROC
		IKRPTF	IKRTYPE	IKRXCPY	IKRXRMV	
IKREPER	INVOKES	IKR#PER	IKRCMD	IKRPROC	IKRXPER	
IKRERDC	INVOKES	IKR#RDC	IKRCMD	IKRPROC	IKRQUIT	IKRTYPE
		IKRXRDC				
IKRERMV	INVOKES	IKR#RMV	IKRCMD	IKRPROC	IKRTYPE	IKRXRMV
IKRERST	INVOKES	IKR#RST	IKRCMD	IKRPROC	IKRTYPE	IKRXCPY
		IKRXSTL				
IKRERTC	INVOKES	IKR#RTC	IKRCMD	IKRPROC	IKRTYPE	IKRXRTC
IKRERVK	INVOKES	IKR#RVK	IKRCMD	IKRPROC	IKRTYPE	IKRXENT
IKRESEL	INVOKES	IKR#DFN	IKR#SEL	IKRCMD	IKRPROC	IKRTYPE
		IKRXCUM				
IKRESTL	INVOKES	IKR#DFN	IKR#STL	IKRCMD	IKRPROC	IKRQUIT

		IKRTYPE	IKRXCUM	IKRXSTL		
IKRETLR	INVOKES	IKRCMD	IKRPROC	IKRTYPE	IKRXENT	IKR#TLR
IKREUND	INVOKES	IKR#UND	IKRCMD	IKRNSTAT	IKRPROC	IKRTYPE
		IKRXENT				
IKREXIT	INVOKES	IKRCMD	IKRPROC			
IKRFBA	INVOKES	IKRCMD	IKRPROC	IKRQUIT	IKRXCYP	
IKRFIX	INVOKES	IKRBFRO	IKRPROC			
IKRFRBUF	INVOKES	IKRBFRO	IKRPROC			
IKRGOGO	INVOKES	IKRPROC	IKRTYPE	IKRXCUM		
IKRGTAPR	INVOKES	IKRINS	IKRPROC			
IKRGTBUF	INVOKES	IKRBFRO	IKRPROC	IKRQUIT		
IKRGTCMP	INVOKES	IKRHD1	IKRPROC			
IKRGTFR	INVOKES	IKRHD1	IKRPROC			
IKRGTGEN	INVOKES	IKRINS	IKRPROC	IKRINX		
IKRGTGRP	INVOKES	IKRHD1	IKRPROC			
IKRGTHD	INVOKES	IKRHD3	IKRHDX	IKRPROC		
IKRGTIMT	INVOKES	IKRINS	IKRPROC	IKRINX		
IKRGTINF	INVOKES	IKRAPA	IKRPROC	IKRPTF	IKRQUIT	IKRPTX
IKRGTMSH	INVOKES	IKRDTFX	IKRIFCTB	IKRPROC		
IKRGTPTF	INVOKES	IKRINS	IKRPROC			
IKRGTREC	INVOKES	IKRBFRO	IKRPROC			
IKRGTTAP	INVOKES	IKRPROC				
IKRGWA	INVOKES	IKRBFRO	IKRPROC			
IKRHDBKT	INVOKES	IKRINS	IKRPROC	IKRPTF		
IKRHDLV2	INVOKES	IKR#AFF	IKR#ALT	IKR#CIS	IKR#CPT	IKR#DEL
		IKR#DFN	IKR#ICL	IKR#IST	IKR#IVL	IKR#PTF
		IKR#RPL	IKR#RQR	IKR#RSTA	IKR#RSV	IKR#SCN
		IKR#SUP	IKR#VER	IKRAPA	IKRCMD	IKRCOM
		IKRNSTAT	IKRPROC	IKRQUIT	IKRTYPE	IKRXCUM
		IKRXENT	IKRXSTL	IKR#INF	IKR#GEN	IKR#EXE
IKRHELP	INVOKES	IKRCMD	IKRPROC			
IKRHSLA	INVOKES	IKRPROC				
IKRIMSMT	INVOKES	IKRPROC	IKRTYPE	IKRXENT		
IKRINMSH	INVOKES	IKRPROC	IKRTYPE	IKRXENT		
IKRINSTL	INVOKES	IKRAIP	IKRAPA	IKRCMD	IKRCOF	IKRINS
		IKRLSI	IKRMIA	IKRMIP	IKRNSTAT	IKRPRD
		IKRPRE	IKRPROC	IKRPTF	IKRQUIT	IKRSUP
		IKRTYPE	IKRXCYP	IKRXENT	IKRXMRG	IKRXRDC
		IKRXRMV	IKRXSTL			
IKRISLTK	INVOKES	IKRLSI	IKRPROC	IKRPTF	IKRQUIT	IKRXCUM
IKRISVBK	INVOKES	IKR#APL	IKRBFRO	IKRCMD	IKRINS	IKRMIP
		IKRNCPCB	IKRNMSG	IKRNSTAT	IKRPRE	IKRPROC
		IKRPTF	IKRTYPE	IKRXCRC	IKRXCUM	IKRXMRG
		IKRXRMV	IKRXRTC			
IKRLAPA	INVOKES	IKRAIP	IKRINS	IKRPROC	IKRPTF	
IKRLCMPR	INVOKES	IKRCOF	IKRPRD	IKRPROC		
IKRLGNRC	INVOKES	IKRINS	IKRPROC	IKRQUIT	IKRGEN	
IKRLIBST	INVOKES	IKRPROC	IKRQUIT	IKRXRDC		
IKRLINCM	INVOKES	IKRINS	IKRPROC	IKRQUIT	IKRIMT	IKRIMR
		IKRGEN				
IKRLKMSH	INVOKES	IKRPROC				
IKRLKUP	INVOKES	IKRAIP	IKRAPA	IKRCOF	IKRCOM	IKRINS
		IKRMIA	IKRMIP	IKRPRD	IKRPRE	IKRPROC
		IKRPTF	IKRSUP	IKRTYPE	IKRXLKP	IKRPTX

		IKRIMT	IKRIMR	IKRGEN		
IKRLMLF	INVOKES	IKRAPA	IKRINS	IKRMIA	IKRPROC	IKRPTF
IKRLMOD	INVOKES	IKRINS	IKRMIP	IKRPROC	IKRPTF	
IKRLOAD	INVOKES	IKRIFCTX	IKRPROC			
IKRLOFI	INVOKES	IKRCOM	IKRMIA	IKRPROC	IKRQUIT	IKRTYPE
IKRLSUP	INVOKES	IKRINS	IKRPROC	IKRPTF	IKRSUP	
IKRMAIN	INVOKES	IKRBFRO	IKRNMSG	IKRNSTAT	IKRPROC	IKRQUIT
IKRMERGE	INVOKES	IKRCOM	IKRPROC	IKRXMRG		
IKRMOD	INVOKES	IKRBFRO	IKRPROC			
IKRMRGIN	INVOKES	IKRAPA	IKRCOF	IKRCOM	IKRHD1	IKRINS
		IKRLIB	IKRMIA	IKRMIP	IKRPRD	IKRPRE
		IKRPROC	IKRPTF	IKRSUP	IKRTYPE	IKRXENT
		IKRXMRG	IKRXRMV	IKRPTX	IKRINX	IKRIMT
		IKRIMR	IKRGEN			
IKRMSG	INVOKES	IKRLMSG	IKRPROC			
IKRNOTE	INVOKES	IKRPROC				
IKROPBUF	INVOKES	IKRBFRO	IKRPROC			
IKROPEN	INVOKES	IKRCMD	IKRDTFBS	IKRHD1	IKRPROC	IKRQUIT
IKROPHF	INVOKES	IKRBFRO	IKRCMD	IKRDTFSD	IKRHD1	IKRNSTAT
		IKRPROC	IKRQUIT	IKRXCPY		
IKROPLNK	INVOKES	IKRDTFSD	IKRPROC	IKRQUIT		
IKROPMSH	INVOKES	IKRDTF	IKRIFCTB	IKRPROC	IKRQUIT	
IKRPAPAL	INVOKES	IKRAIP	IKRAPA	IKRINS	IKRMIA	IKRPRE
		IKRPROC	IKRPTF	IKRTYPE	IKRXRTC	
IKRPARSE	INVOKES	IKRCMD	IKRNCPCB	IKRNMSG	IKRNSTAT	IKRPROC
		IKRQUIT	IKRTYPE			
IKRPCREQ	INVOKES	IKRPRE	IKRPROC	IKRTYPE	IKRXENT	
IKRPDZAP	INVOKES	IKR#AFF	IKR#ALT	IKR#SCN	IKRAPA	IKRCMD
		IKRLSI	IKRMIA	IKRNSTAT	IKRPROC	IKRTYPE
		IKRXENT				
IKRPERS	INVOKES	IKRAPA	IKRHD1	IKRHD2	IKRHD3	IKRHDX
		IKRINS	IKRPRD	IKRPROC	IKRPTF	IKRQUIT
		IKRXPER				
IKRPGENL	INVOKES	IKRINS	IKRPROC	IKRTYPE	IKRXENT	IKRIMR
		IKRGEN				
IKRPHMSH	INVOKES	IKRIFCTB	IKRPROC			
IKRPIO	INVOKES	IKRNCPCB	IKRPROC			
IKRPPTFL	INVOKES	IKRAIP	IKRINS	IKRMIP	IKRPRE	IKRPROC
		IKRPTF	IKRSUP	IKRTYPE	IKRXRTC	
IKRPRAPL	INVOKES	IKRPROC	IKRTYPE	IKRXENT		
IKRPRCS	INVOKES	IKRPROC				
IKRPSDMD	INVOKES	IKRIFCTX	IKRPROC			
IKRPTLNK	INVOKES	IKRPROC				
IKRPTMSH	INVOKES	IKRDTF	IKRIFCTB	IKRPROC		
IKRQUEUE	INVOKES	IKRAIP	IKRAPA	IKRBFRO	IKRCOF	IKRCOM
		IKRFRE	IKRHD1	IKRHD3	IKRHDX	IKRINS
		IKRLIB	IKRMIA	IKRMIP	IKRPRD	IKRPRE
		IKRPROC	IKRPTF	IKRQUIT	IKRSUP	IKRPTX
		IKRINX	IKRIMT	IKRIMR	IKRGEN	
IKRRACSV	INVOKES	IKRPROC				
IKRRAESV	INVOKES	IKRCMD	IKRPROC	IKRQUIT	IKRTYPE	IKRXENT
IKRRALNK	INVOKES	IKRCMD	IKRPROC	IKRQUIT	IKRXENT	

IKRRASRV	INVOKES	IKRCMD	IKRPROC				
IKRRCV	INVOKES	IKRIFCTB	IKRPROC				
IKRRDCE	INVOKES	IKRCMD	IKRCOF	IKRINS	IKRLIB	IKRPRD	
		IKRPROC	IKRTYPE	IKRXRDC			
IKRRDIPT	INVOKES	IKRPROC					
IKRRDLV2	INVOKES	IKRCMD	IKRNCPCB	IKRNMSG	IKRNSTAT	IKRPROC	
		IKRQUIT					
IKRRDNXT	INVOKES	IKRNSTAT	IKRPROC				
IKRREGCK	INVOKES	IKRINS	IKRPROC	IKRQUIT	IKRTYPE	IKRGEN	
IKRREQFR	INVOKES	IKRCMD	IKRFRE	IKRHD1	IKRHDX	IKRPROC	
		IKRQUIT					
IKRREWR	INVOKES	IKRBFRO	IKRPROC				
IKRRLZAP	INVOKES	IKR#AFF	IKR#ALT	IKRCMD	IKRLSI	IKRNSTAT	
		IKRPROC	IKRTYPE	IKRXENT			
IKRRL2	INVOKES	IKRAIP	IKRAPA	IKRCOM	IKRINS	IKRMIA	
		IKRMIP	IKRPROC	IKRPTF	IKRTYPE	IKRPTX	
IKRRL3	INVOKES	IKRINS	IKRPROC	IKRTYPE			
IKRRL4	INVOKES	IKRCOF	IKRCOM	IKRPRD	IKRPROC	IKRTYPE	
		IKRXRTC					
IKRRL5	INVOKES	IKRPROC					
IKRRL6	INVOKES	IKRPROC					
IKRRL7	INVOKES	IKRAPA	IKRINS	IKRMIA	IKRPROC	IKRPTF	
		IKRTYPE	IKRGEN				
IKRRMGEN	INVOKES	IKRPROC	IKRQUIT	IKRXRMV			
IKRRTC	INVOKES	IKRCMD	IKREBC	IKRFRE	IKRHD1	IKRHD2	
		IKRHD3	IKRHDX	IKRINS	IKRPROC	IKRTYPE	
		IKRXRTC					
IKRRTRVE	INVOKES	IKRCMD	IKRCOM	IKRINS	IKRPROC	IKRQUIT	
		IKRTYPE	IKRXENT	IKRIMT	IKRIMR	IKRGEN	
IKRRWBUF	INVOKES	IKRBFRO	IKRPROC				
IKRSAASM	INVOKES	IKRCMD	IKRPROC	IKRQUIT	IKRTYPE	IKRXENT	
IKRSALIB	INVOKES	IKRLSI	IKRPROC				
IKRSAMNT	INVOKES	IKRCMD	IKRNSTAT	IKRPROC	IKRTYPE	IKRXENT	
IKRSAPL	INVOKES	IKRBFRO	IKRCMD	IKRNCPCB	IKRNMSG	IKRNSTAT	
		IKRPROC	IKRTYPE	IKRXCUM	IKR#TLR		
IKRSEND	INVOKES	IKRIFCTB	IKRPROC				
IKRSERR	INVOKES	IKRCMD	IKRPROC	IKRQUIT			
IKRSET	INVOKES	IKRPROC	IKRQUIT				
IKRSHAPR	INVOKES	IKRAPA	IKRINS	IKRPROC			
IKRSHCMP	INVOKES	IKRINS	IKRPROC				
IKRSHGEN	INVOKES	IKRINS	IKRPROC	IKRINX	IKRGEN		
IKRSHGRP	INVOKES	IKRPRD	IKRPROC				
IKRSHPRD	INVOKES	IKRPRD	IKRPROC				
IKRSHPTF	INVOKES	IKRINS	IKRPROC	IKRPTF			
IKRSHREL	INVOKES	IKRINS	IKRPROC				
IKRSLA	INVOKES	IKRPROC	IKRQUIT				
IKRSLZAP	INVOKES	IKR#AFF	IKR#DEL	IKR#IST	IKR#RPL	IKRAPA	
		IKRCMD	IKRCOM	IKRLSI	IKRMIA	IKRNSTAT	
		IKRPROC	IKRPTF	IKRQUIT	IKRTYPE	IKRXENT	
		IKRXRMV					
IKRSPCLS	INVOKES	IKRPROC					
IKRSREPQ	INVOKES	IKRPROC	IKRQUIT				
IKRSRTMD	INVOKES	IKRAIP	IKRAPA	IKRINS	IKRMIA	IKRMIP	
		IKRPROC	IKRPTF	IKRTYPE	IKRGEN		

IKRSTAT	INVOKES	IKRLSTAT	IKRPROC				
IKRSTMTS	INVOKES	IKRPROC	IKRXCUM				
IKRSVCMD	INVOKES	IKRPROC	IKRQUIT				
IKRSWTHF	INVOKES	IKRBFRO	IKRPROC				
IKRTAIL	INVOKES	IKRAPA	IKRINS	IKRLSI	IKRMIA	IKRNSTAT	
		IKRPROC	IKRPTF	IKRQUIT	IKRTYPE	IKRXENT	
		IKRXRMV	IKRGEN				
IKRTPMSH	INVOKES	IKRPROC	IKRQUIT	IKRXSTL			
IKRTRLCP	INVOKES	IKRPROC					
IKRUIRP	INVOKES	IKRAPA	IKRCOM	IKRMIA	IKRMIP	IKRPRE	
		IKRPROC	IKRQUIT	IKRTYPE	IKRXENT		
IKRUNDO	INVOKES	IKRAPA	IKRCMD	IKRCOM	IKRMIA	IKRNSTAT	
		IKRPROC	IKRPTF	IKRQUIT	IKRTYPE	IKRXENT	
		IKRXRDC	IKRXRMV	IKRGEN			
IKRUOMT	INVOKES	IKRPROC	IKRXENT				
IKRUPGR	INVOKES	IKRAIP	IKRAPA	IKRCMD	IKRINS	IKRMIA	
		IKRMIP	IKRNSTAT	IKRPRE	IKRPROC	IKRPTF	
		IKRQUIT	IKRTYPE	IKRXCUM	IKRXMRG	IKRXRMV	
		IKRXRTC	IKRGEN				
IKRWKF	INVOKES	IKRDTFSD	IKRPROC				
IKRWOF	INVOKES	IKRPROC					
IKRWOR	INVOKES	IKRNSTAT	IKRPROC				
IKRWRSQ	INVOKES	IKRPROC	IKRQUIT				
IKRWTO	INVOKES	IKRPROC					
IKRWTP	INVOKES	IKREBC	IKRPROC				

Figures 12-5: Sorted by Invoked Macros

IKR#AFF	IS INVOKED BY	IKRCKLV2 IKRECOR IKRHDLV2 IKRPDZAP IKRRLZAP IKRSLZAP
IKR#ALT	IS INVOKED BY	IKRCKLV2 IKRHDLV2 IKRPDZAP IKRRLZAP
IKR#APL	IS INVOKED BY	IKREAPL IKRISVBK
IKR#ARC	IS INVOKED BY	IKREARC
IKR#BCK	IS INVOKED BY	IKREBCK
IKR#CIS	IS INVOKED BY	IKRCKLV2 IKRHDLV2
IKR#COR	IS INVOKED BY	IKRECOR
IKR#CPT	IS INVOKED BY	IKRCKLV2 IKRHDLV2
IKR#CPY	IS INVOKED BY	IKRECPY
IKR#CRE	IS INVOKED BY	IKRECRE
IKR#DEL	IS INVOKED BY	IKRCKLV2 IKRHDLV2 IKRSLZAP
IKR#DFN	IS INVOKED BY	IKRCKLV2 IKRECOR IKRESEL IKRESTL IKRHDLV2
IKR#DMP	IS INVOKED BY	IKREDMP
IKR#EXE	IS INVOKED BY	IKRCKLV2 IKRHDLV2
IKR#GEN	IS INVOKED BY	IKRHDLV2
IKR#ICL	IS INVOKED BY	IKRCKLV2 IKRHDLV2
IKR#INC	IS INVOKED BY	IKREINC
IKR#INF	IS INVOKED BY	IKRCKLV2 IKRHDLV2
IKR#IST	IS INVOKED BY	IKRCKLV2 IKRHDLV2 IKRSLZAP
IKR#IVL	IS INVOKED BY	IKRECOR IKRHDLV2
IKR#LIST	IS INVOKED BY	IKRELST
IKR#LKP	IS INVOKED BY	IKRELKP
IKR#MRG	IS INVOKED BY	IKREMRG
IKR#PER	IS INVOKED BY	IKREPER
IKR#PTF	IS INVOKED BY	IKRHDLV2
IKR#RDC	IS INVOKED BY	IKRERDC
IKR#RMV	IS INVOKED BY	IKRERMV
IKR#RPL	IS INVOKED BY	IKRCKLV2 IKRHDLV2 IKRSLZAP
IKR#RQR	IS INVOKED BY	IKRCKLV2 IKRHDLV2
IKR#RST	IS INVOKED BY	IKRERST
IKR#RSTA	IS INVOKED BY	IKRHDLV2
IKR#RSV	IS INVOKED BY	IKRCKLV2 IKRHDLV2
IKR#RTC	IS INVOKED BY	IKRERTC
IKR#RVK	IS INVOKED BY	IKRERVK
IKR#SCN	IS INVOKED BY	IKRCKLV2 IKRHDLV2 IKRPDZAP
IKR#SEL	IS INVOKED BY	IKRESEL
IKR#STL	IS INVOKED BY	IKRESTL
IKR#SUP	IS INVOKED BY	IKRHDLV2
IKR#TLR	IS INVOKED BY	IKRETLR IKRSAPL
IKR#UND	IS INVOKED BY	IKREUND
IKR#VER	IS INVOKED BY	IKRCKLV2 IKRHDLV2
IKRAIP	IS INVOKED BY	IKRAPPLY IKRAPTF IKRBARDT IKRCKPTF IKRINSTL IKRLAPA IKRLKUP IKRPAPAL IKRPPTFL IKRQUEUE IKRRL2 IKRSRTMD IKRUPGR
IKRAPA	IS INVOKED BY	IKRAAPAR IKRBARDT IKRCHECK IKRCKFIX IKRDLELE IKRDLIN IKRECOR IKREMU IKRGTINF IKRHDLV2 IKRINSTL IKRLKUP IKRLMLF IKRMRGIN IKRPAPAL IKRPDZAP IKRPERS IKRQUEUE IKRRL2 IKRRL7

		IKRSHAPR	IKRSLZAP	IKRSRTMD	IKRTAIL
		IKRUIRP	IKRUNDO	IKRUPGR	
IKRBFRO	IS INVOKED BY	IKRCLOSE	IKRFIX	IKRFRBUF	IKRGTBUF
		IKRGTRC	IKRGWA	IKRISVBK	IKRMAIN
		IKRMOD	IKROPBUF	IKROPHF	IKRQUEUE
		IKRREWR	IKRRWBUF	IKRSAPL	IKRSWTHF
IKRCMD	IS INVOKED BY	IKRAAPAR	IKRAGEN	IKRAPPLY	IKRAPTF
		IKRBIO	IKRBKT	IKRBKUP	IKRCGT
		IKRCHECK	IKRCKASS	IKRCKD	IKRCKFIX
		IKRCKLV2	IKRCLOSE	IKRDLIN	IKREAPL
		IKREARC	IKREBCK	IKRECOR	IKRECPY
		IKRECRE	IKREDMP	IKREINC	IKRELKP
		IKRELST	IKREMBDY	IKREMRG	IKREMU
		IKREPER	IKRERDC	IKRERMV	IKRERST
		IKRERTC	IKRERVK	IKRESEL	IKRESTL
		IKRETLR	IKREUND	IKREXIT	IKRFBA
		IKRHDLV2	IKRHELP	IKRINSTL	IKRISVBK
		IKROPEN	IKROPHF	IKRPARSE	IKRPDZAP
		IKRRAESV	IKRRALNK	IKRRASRV	IKRRDCE
		IKRRDLV2	IKRREQFR	IKRRLZAP	IKRRTC
		IKRRTRVE	IKRSAASM	IKRSAMNT	IKRSAPL
		IKRSERR	IKRSLZAP	IKRUNDO	IKRUPGR
IKRCOF	IS INVOKED BY	IKRAPRD	IKRBKPR	IKRDLELE	IKRDLIN
		IKREMU	IKRINSTL	IKRLCMPR	IKRLKUP
		IKRMARGIN	IKRQUEUE	IKRRDCE	IKRRL4
IKRCOM	IS INVOKED BY	IKRAAPAR	IKRAGEN	IKRAPRD	IKRAPTF
		IKRCKPTF	IKRDLELE	IKREMU	IKRHDLV2
		IKRLKUP	IKRLOFI	IKRMERGE	IKRMARGIN
		IKRQUEUE	IKRRL2	IKRRL4	IKRRTRVE
		IKRSLZAP	IKRUIRP	IKRUNDO	
IKRCOMRX	IS INVOKED BY	IKRCOMR			
IKRDTF	IS INVOKED BY	IKRCLMSH	IKROPMSH	IKRPTMSH	
IKRDTFBS	IS INVOKED BY	IKRCGT	IKROPEN		
IKRDTFSD	IS INVOKED BY	IKRCGT	IKROPHF	IKROPLNK	IKRWKF
IKRDTFX	IS INVOKED BY	IKRGTMSH			
IKREBC	IS INVOKED BY	IKRDIAGN	IKRRTC	IKRWTP	
IKRFRE	IS INVOKED BY	IKRCPIN	IKRCREAT	IKRCUTHF	IKRDEQ
		IKRDQFR	IKREMU	IKRQUEUE	IKRREQFR
		IKRRTC			
IKRFUNC	IS INVOKED BY	IKRCKLV2	IKRCNVPC	IKREUND	IKRGOGO
		IKRMARGIN	IKRPERS	IKRPHMSH	IKRQUEDM
		IKRRDLV2	IKRRMGEN	IKRSETDM	IKRSHCMP
		IKRSHPRD	IKRWKF10	IKRWKF4R	IKRWKF5W
		IKWRSQ	IKRWTOFL		
IKRGEN	IS INVOKED BY	IKRAGEN	IKRBGRDT	IKRDLIN	IKRECOR
		IKRLGNRC	IKRLINCM	IKRLKUP	IKRMARGIN
		IKRPGENL	IKRQUEUE	IKRREGCK	IKRRL7
		IKRRTRVE	IKRSHGEN	IKRSRTMD	IKRTAIL
		IKRUNDO	IKRUPGR		
IKRHDX	IS INVOKED BY	IKRCPIN	IKRCREAT	IKRCUTHF	IKRDEQ
		IKRDQFR	IKREMU	IKRGTHD	IKRPERS
		IKRQUEUE	IKRREQFR	IKRRTC	
IKRHD1	IS INVOKED BY	IKRCPIN	IKRCREAT	IKRDLIN	IKRDQFR
		IKREMU	IKRGTCMP	IKRGTFR	IKRGTGRP
		IKRMARGIN	IKROPEN	IKROPHF	IKRPERS

		IKRQUEUE	IKRREQFR	IKRRTC	
IKRHD2	IS INVOKED BY	IKRCREAT	IKRPERS	IKRRTC	
IKRHD3	IS INVOKED BY	IKRCREAT	IKRDEQ	IKRDQFR	IKREMU
		IKRGTHD	IKRPERS	IKRQUEUE	IKRRTC
IKRIFCTB	IS INVOKED BY	IKRCLMSH	IKRGTMSH	IKROPMSH	IKRPHMSH
		IKRPTMSH	IKRRCV	IKRSEND	
IKRIFCTX	IS INVOKED BY	IKRLOAD	IKRPSDMD		
IKRIMR	IS INVOKED BY	IKRAGEN	IKRDLIN	IKRLINCM	IKRLKUP
		IKRMGIN	IKRPGENL	IKRQUEUE	IKRRTRVE
IKRIMT	IS INVOKED BY	IKRAGEN	IKRDLIN	IKRLINCM	IKRLKUP
		IKRMGIN	IKRQUEUE	IKRRTRVE	
IKRINS	IS INVOKED BY	IKRAAPAR	IKRACOMP	IKRAGEN	IKRAPPLY
		IKRAPTF	IKRBARDT	IKRBRDRT	IKRBKPR
		IKRBRDRT	IKRCHECK	IKRCKCMP	IKRDLIN
		IKRECOR	IKREMBDY	IKREMU	IKRGTAPR
		IKRGTGEN	IKRGTIMT	IKRGTPTF	IKRHDBKT
		IKRINSTL	IKRISVBK	IKRLAPA	IKRLGNRC
		IKRLINCM	IKRLKUP	IKRLMLF	IKRLMOD
		IKRLSUP	IKRMGIN	IKRPAPAL	IKRPERS
		IKRPGENL	IKRPPTFL	IKRQUEUE	IKRRDCE
		IKRREGCK	IKRRL2	IKRRL3	IKRRL7
		IKRRTC	IKRRTRVE	IKRSHAPR	IKRSHCMP
		IKRSHGEN	IKRSHPTF	IKRSHREL	IKRSRTMD
		IKRTAIL	IKRUPGR		
IKRINX	IS INVOKED BY	IKRDLIN	IKRGTGEN	IKRGTIMT	IKRMGIN
		IKRQUEUE	IKRSHGEN		
IKRLCPCB	IS INVOKED BY	IKRCPC			
IKRLIB	IS INVOKED BY	IKRBKPR	IKRDLIN	IKREMU	IKRMGIN
		IKRQUEUE	IKRRDCE		
IKRLMSG	IS INVOKED BY	IKRMSG			
IKRLSI	IS INVOKED BY	IKRBKPR	IKRBKT	IKREMBDY	IKRINSTL
		IKRISLNK	IKRPDZAP	IKRRLZAP	IKRSALIB
		IKRSLZAP	IKRTAIL		
IKRLSTAT	IS INVOKED BY	IKRSTAT			
IKRMIA	IS INVOKED BY	IKRAAPAR	IKRCKFIX	IKRDLELE	IKREMU
		IKRINSTL	IKRLKUP	IKRLMLF	IKRLOFI
		IKRMGIN	IKRPAPAL	IKRPDZAP	IKRQUEUE
		IKRRL2	IKRRL7	IKRSLZAP	IKRSRTMD
		IKRTAIL	IKRUIRP	IKRUNDO	IKRUPGR
IKRMIP	IS INVOKED BY	IKRAAPAR	IKRAPPLY	IKRAPRD	IKRAPTF
		IKRCKPTF	IKRDLELE	IKRDLIN	IKRECOR
		IKREMU	IKRINSTL	IKRISVBK	IKRLKUP
		IKRLMOD	IKRMGIN	IKRPPTFL	IKRQUEUE
		IKRRL2	IKRSRTMD	IKRUIRP	IKRUPGR
IKRNCPCB	IS INVOKED BY	IKRISVBK	IKRPARSE	IKRPIO	IKRRDLV2
		IKRSAPL			
IKRNMSG	IS INVOKED BY	IKRISVBK	IKRMAIN	IKRPARSE	IKRRDLV2
		IKRSAPL			
IKRNSTAT	IS INVOKED BY	IKRAPPLY	IKRBKUP	IKRCKCMP	IKRDIAGN
		IKRECOR	IKREINC	IKREMBDY	IKREUND
		IKRHDLV2	IKRINSTL	IKRISVBK	IKRMAIN
		IKROPHF	IKRPARSE	IKRPDZAP	IKRRDLV2
		IKRRDNXT	IKRRLZAP	IKRSAMNT	IKRSAPL

		IKRSLZAP	IKRTAIL	IKRUNDO	IKRUPGR
		IKRWOR			
IKRPRD	IS INVOKED BY	IKRAPRD	IKRBKPR	IKRDLIN	IKREMU
		IKRINSTL	IKRLCMPR	IKRLKUP	IKRMRGIN
		IKRPERS	IKRQUEUE	IKRRDCE	IKRRL4
		IKRSHGRP	IKRSHPRD		
IKRPRE	IS INVOKED BY	IKRAAPAR	IKRACMP1	IKRAPPLY	IKRAPRD
		IKRCKPTF	IKRDLIN	IKRECOR	IKREMU
		IKRINSTL	IKRISVBK	IKRLKUP	IKRMRGIN
		IKRPAPAL	IKRPCREQ	IKRPPTFL	IKRQUEUE
		IKRUIRP	IKRUPGR		
IKRPROC	IS INVOKED BY	IKRAAPAR	IKRACMP1	IKRACOMP	IKRAGEN
		IKRAPPLY	IKRAPRD	IKRAPTF	IKRARCH
		IKRBARDT	IKRBGRDT	IKRBINS	IKRBIO
		IKRBKPR	IKRBKT	IKRBKUP	IKRBPRTD
		IKRCGT	IKRCHECK	IKRCKASS	IKRCKCMP
		IKRCKCTL	IKRCKD	IKRCKFIX	IKRCKLV2
		IKRCKPTF	IKRCLMSH	IKRCLOSE	IKRCNVRT
		IKRCOMR	IKRCOPY	IKRCPC	IKRCPIN
		IKRCREAT	IKRCUTHF	IKRCVTLU	IKRDATE
		IKRDEBUG	IKRDEL	IKRDEQ	IKRDIAGN
		IKRDLELE	IKRDLIN	IKRDQFR	IKREAPL
		IKREARC	IKREBCK	IKRECOR	IKRECPY
		IKRECRE	IKREDMP	IKREINC	IKRELKP
		IKRELST	IKREMBDY	IKREMRG	IKREMU
		IKREPER	IKRERDC	IKRERMV	IKRERST
		IKRERTC	IKRERVK	IKRESEL	IKRESTL
		IKRETLR	IKREUND	IKREXIT	IKRFBA
		IKRFIX	IKRFRBUF	IKRGOGO	IKRGTAPR
		IKRGTBUF	IKRGTCMP	IKRGTFR	IKRGTGEN
		IKRGTGRP	IKRGTIMT	IKRGTINF	IKRGTHD
		IKRGTMSH	IKRGTPTF	IKRGTREC	IKRGTTAP
		IKRGWA	IKRHDBKT	IKRHDLV2	IKRHELP
		IKRHSLA	IKRINMSH	IKRIMSMT	IKRINSTL
		IKRISLNK	IKRISVBK	IKRLAPA	IKRLCMPR
		IKRLGNRC	IKRLIBST	IKRLINCM	IKRLKMSH
		IKRLKUP	IKRLMLF	IKRLMOD	IKRLOAD
		IKRLOFI	IKRLSUP	IKRMAIN	IKRMERGE
		IKRMOD	IKRMRGIN	IKRMSG	IKRNOTE
		IKROPBUF	IKROPEN	IKROPHF	IKROPLNK
		IKROPMSH	IKRPAPAL	IKRPARSE	IKRPCREQ
		IKRPDZAP	IKRPERS	IKRPGENL	IKRPHMSH
		IKRPPIO	IKRPPTFL	IKRPRAPL	IKRPRCS
		IKRPSDMD	IKRPTLNK	IKRPTMSH	IKRQUEUE
		IKRRACSV	IKRRAESV	IKRRALNK	IKRRASRV
		IKRRCV	IKRRDCE	IKRRDIPT	IKRRDLV2
		IKRRDNXT	IKRREGCK	IKRREQFR	IKRREWR
		IKRRLZAP	IKRRL2	IKRRL3	IKRRL4
		IKRRL5	IKRRL6	IKRRL7	IKRRMGEN
		IKRRTC	IKRRTRVE	IKRRWBUF	IKRSAASM
		IKRSALIB	IKRSAMNT	IKRSAPL	IKRSEND
		IKRSERR	IKRSET	IKRSHAPR	IKRSHCMP
		IKRSHGEN	IKRSHGRP	IKRSHPRD	IKRSHPTF
		IKRSHREL	IKRSLA	IKRSLZAP	IKRSPCLS

		IKRSREPQ	IKRSRTMD	IKRSTAT	IKRSTMTS
		IKRSVCMD	IKRSWTHF	IKRTAIL	IKRTPMSH
		IKRTRLCP	IKRUIRP	IKRUNDO	IKRUOMT
		IKRUPGR	IKRWKF	IKRWOF	IKRWOR
		IKWRSQ	IKRWTO	IKRWTP	
IKRPTF	IS INVOKED BY	IKRAPPLY	IKRAPTF	IKRBARDT	IKRBPRDT
		IKRCHECK	IKRDLIN	IKRECOR	IKREMU
		IKRGTINF	IKRHDBKT	IKRINSTL	IKRISLNK
		IKRISVBK	IKRLAPA	IKRLKUP	IKRLMLF
		IKRLMOD	IKRLSUP	IKRMRGIN	IKRPAPAL
		IKRPERS	IKRPPTFL	IKRQUEUE	IKRRL2
		IKRRL7	IKRSHPTF	IKRSLZAP	IKRSRTMD
		IKRTAIL	IKRUNDO	IKRUPGR	
IKRPTX	IS INVOKED BY	IKRGTINF	IKRLKUP	IKRMRGIN	IKRQUEUE
		IKRRL2			
IKRQUIT	IS INVOKED BY	IKRAAPAR	IKRAGEN	IKRAPRD	IKRAPTF
		IKRKBPR	IKRKBUP	IKRCGT	IKRCHECK
		IKRCKASS	IKRCKD	IKRCLMSH	IKRCPIN
		IKRDEQ	IKRDIAGN	IKRDLIN	IKREAPL
		IKRECOR	IKRELST	IKREMBDY	IKRERDC
		IKRESTL	IKRFBA	IKRGTBUF	IKRGTINF
		IKRHDLV2	IKRINSTL	IKRISLNK	IKRLGNRC
		IKRLIBST	IKRLINCM	IKRLOFI	IKRMAIN
		IKROPEN	IKROPHF	IKROPLNK	IKROPMSH
		IKRPARSE	IKRPERS	IKRQUEUE	IKRRAESV
		IKRRALNK	IKRRDLV2	IKRREGCK	IKRREQFR
		IKRRMGEN	IKRRTRVE	IKRSAASM	IKRSERR
		IKRSET	IKRSLA	IKRSLZAP	IKRSREPQ
		IKRSVCMD	IKRTAIL	IKRTPMSH	IKRUIRP
		IKRUNDO	IKRUPGR	IKWRSQ	
IKRSUP	IS INVOKED BY	IKRAPPLY	IKRAPTF	IKRBPRDT	IKRCKPTF
		IKRINSTL	IKRLKUP	IKRLSUP	IKRMRGIN
		IKRPPTFL	IKRQUEUE		
IKRTYPE	IS INVOKED BY	IKRAAPAR	IKRACMP1	IKRACOMP	IKRAGEN
		IKRAPPLY	IKRAPRD	IKRAPTF	IKRARCH
		IKRBARDT	IKRBGRDT	IKRKBPR	IKRKBKT
		IKRBPRDT	IKRCGT	IKRCHECK	IKRCKCTL
		IKRCKFIX	IKRCKLV2	IKRCKPTF	IKRDLIN
		IKREAPL	IKREARC	IKREBCK	IKRECOR
		IKREINC	IKRELKP	IKRELST	IKREMBDY
		IKREMU	IKRERDC	IKRERMV	IKRERST
		IKRERTC	IKRErvK	IKRESEL	IKRESTL
		IKRETLR	IKREUND	IKRGOGO	IKRHDLV2
		IKRINMSH	IKRIMSMT	IKRINSTL	IKRISVBK
		IKRLKUP	IKRLOFI	IKRMRGIN	IKRPAPAL
		IKRPARSE	IKRPCREQ	IKRPDZAP	IKRPGENL
		IKRPPTFL	IKRPRAPL	IKRRAESV	IKRRDCE
		IKRREGCK	IKRRLZAP	IKRRL2	IKRRL3
		IKRRL4	IKRRL7	IKRRTC	IKRRTRVE
		IKRSAASM	IKRSAMNT	IKRSAPL	IKRSLZAP
		IKRSRTMD	IKRTAIL	IKRUIRP	IKRUNDO
		IKRUPGR			
IKRXCpy	IS INVOKED BY	IKRKBPR	IKRKBUP	IKRCKD	IKRCOPY
		IKREBCK	IKRECPY	IKREMU	IKRERST

IKRXCRE	IS INVOKED BY	IKRFBA	IKRINSTL	IKROPHF	
		IKRAPPLY	IKRBKPR	IKRCREAT	IKRECRE
		IKRISVBK			
IKRXCUM	IS INVOKED BY	IKRAPPLY	IKRCGT	IKRCKLV2	IKRELST
		IKRESEL	IKRESTL	IKRGOGO	IKRHDLV2
		IKRISLNK	IKRISVBK	IKRSAPL	IKRSTMTS
		IKRUPGR			
IKRXDMP	IS INVOKED BY	IKRDEBUG	IKREDMP		
IKRXENT	IS INVOKED BY	IKRAAPAR	IKRACMP1	IKRACOMP	IKRAGEN
		IKRAPPLY	IKRAPRD	IKRAPTF	IKRARCH
		IKRCHECK	IKRCKCMP	IKRCKCTL	IKRCKFIX
		IKRCKLV2	IKRCKPTF	IKREAPL	IKREARC
		IKRECOR	IKREINC	IKREMBDY	IKRERVK
		IKRETLR	IKREUND	IKRHDLV2	IKRINMSH
		IKRIMSMT	IKRINSTL	IKRMRGIN	IKRPCREQ
		IKRPDZAP	IKRPGENL	IKRPRAPL	IKRRAESV
		IKRRALNK	IKRRLZAP	IKRRTRVE	IKRSAASM
		IKRSAMNT	IKRSLZAP	IKRTAIL	IKRUIRP
		IKRUNDO	IKRUOMT		
IKRXLKP	IS INVOKED BY	IKRELKP	IKRLKUP		
IKRXMRG	IS INVOKED BY	IKRAPPLY	IKRBKPR	IKREMRG	IKRINSTL
		IKRISVBK	IKRMERGE	IKRMRGIN	IKRUPGR
IKRXPER	IS INVOKED BY	IKREPER	IKRPERS		
IKRXRDC	IS INVOKED BY	IKREMBDY	IKRERDC	IKRINSTL	IKRLIBST
		IKRRDCE	IKRUNDO		
IKRXRMV	IS INVOKED BY	IKRDLIN	IKRECOR	IKREMBDY	IKREMU
		IKRERMV	IKRINSTL	IKRISVBK	IKRMRGIN
		IKRRMGEN	IKRSLZAP	IKRTAIL	IKRUNDO
		IKRUPGR			
IKRXRTC	IS INVOKED BY	IKRERTC	IKRISVBK	IKRPAPAL	IKRPPTFL
		IKRRL4	IKRRTC	IKRUPGR	
IKRXSTL	IS INVOKED BY	IKRCHECK	IKRCKLV2	IKRERST	IKRESTL
		IKRHDLV2	IKRINSTL	IKRTPMSH	

Module-IKRCOMRG Interrelations

Figures 12-6: Sorted by Using Modules

IKRAAPAR	USES	CRGCMND			
IKRAPPLY	USES	CRGMATRP	CRGCMND	CRGSLCT	CRGSLMTP
IKRBIO	USES	CRGCMND	CRGFLDTF		
IKRBKPR	USES	CRGCMND	CRGNTAPE	CRGSENDP	
IKRBKT	USES	CRGMATRP	CRGSENDP		
IKRBKUP	USES	CRGBFFLG	CRGBKUP	CRGBFBEG	CRGSCPAD
		CRGPRTL	CRGMATRP	CRGINRDR	CRGCMND
		CRGFLDTF	CRGSAPL	CRGINLP	CRGFLST
IKRCGT	USES	CRGMATRP	CRGCMND	CRGFLOPN	CRGFLDPN
		CRGFLDTF			CRGFLISV
IKRCHECK	USES	CRGCMND			
IKRCKD	USES	CRGCMND			
IKRCKFIX	USES	CRGSITP			
IKRCKLV2	USES	CRGCMND			
IKRCLOSE	USES	CRGFLST	CRGFLOPN	CRGFLSYS	CRGFLFID
IKRCOPY	USES	CRGFLDTF			
IKRCPIN	USES	CRGHFLRC	CRGINLP		
IKRCREAT	USES	CRGHFLRC			
IKRCUTHF	USES	CRGHFLRC			
IKRDEBUG	USES	CRGPRTCR	CRGPRTL	CRGHFLRC	
IKRDIAGN	USES	CRGINRDR			
IKRDLIN	USES	CRGCMND			
IKREAPL	USES	CRGPRTAB	CRGMATRP	CRGSITP	CRGNOPRT
IKREARC	USES	CRGMATRP	CRGSITP		
IKREBCK	USES	CRGSITP	CRGINLP		
IKRECOR	USES	CRGMATRP	CRGINEOF	CRGINRDR	CRGSITP
IKRECPY	USES	CRGSITP			
IKRECRE	USES	CRGSITP			
IKREDMP	USES	CRGSITP			
IKREINC	USES	CRGMATRP	CRGSITP		
IKRELKP	USES	CRGSITP			
IKRELST	USES	CRGSITP			
IKREMBDY	USES	CRGIPF	CRGPRTCR	CRGPRTL	CRGSITP
					CRGSENDP
IKREMRG	USES	CRGSITP			
IKREMU	USES	CRGCMND			
IKREPER	USES	CRGSITP			
IKRERDC	USES	CRGINRDR	CRGSITP		
IKRERMV	USES	CRGSITP			
IKRERST	USES	CRGSITP	CRGINLP		
IKRERTC	USES	CRGSITP			
IKRERVK	USES	CRGSITP	CRGNOPRT		
IKRESEL	USES	CRGSITP			
IKRESTL	USES	CRGSITP	CRGINLP	CRGIPF	CRGLNKF
IKRETLR	USES	CRGMATRP	CRGSITP		
IKREUND	USES	CRGSITP			
IKREXIT	USES	CRGCMND	CRGFLOPN	CRGSLCT	
IKRFBA	USES	CRGCMND			
IKRFX	USES	CRGBFBEG			
IKRFRBUF	USES	CRGBFBEG			

IKRGTBUF	USES	CRGBFOPN	CRGBKUP	CRGBFBEG				
IKRGTHD	USES	CRGHFDEV	CRGHFFBL	CRGFLDTF	CRGHFCKD	CRGHFFBA		
IKRGTREC	USES	CRGBFOPN	CRGBFBEG	CRGFLDTF				
IKRGWA	USES	CRGBFOPN	CRGBFBEG					
IKRHDLV2	USES	CRGINEOF	CRGINRDR	CRGCMND				
IKRINSTL	USES	CRGCMND	CRGSENDP	CRGFLDTF				
IKRISLNK	USES	CRGLNKF	CRGSENDP					
IKRISVBK	USES	CRGPRTAB	CRGMATRP	CRGPINFO	CRGCDLVL	CRGINEOF		
		CRGINRDR	CRGSYNER	CRGCMND	CRGSITP	CRGFLDTF		
		CRGSAPL	CRGSLCT	CRGNOPRT	CRGSLMTP	CRGPTFLI		
IKRLOAD	USES	CRGSCPAD	CRGPRTCR	CRGFLOPN	CRGFLOSV	CRGFLVSM		
		CRGFLF1D						
IKRMAIN	USES	CRGBFFLG	CRGBFBEG	CRGSCPAD	CRGSCPAG	CRGPRTCR		
		CRGPRTL	CRGMATRP	CRGCDLVL	CRGINEOF	CRGINRDR		
		CRGSYNER	CRGCMND	CRGSITP	CRGFLST	CRGFLFLG		
		CRGFLOPN	CRGFLDTF	CRGSAPL	CRGLMSHP	CRGNTAPE		
		CRGPTFLI						
IKRMOD	USES	CRGBFBEG						
IKRNOTE	USES	CRGHFDEV	CRGHFFBL	CRGHFLBL	CRGHFNTR	CRGHFNBL		
		CRGHFCKD	CRGHFCC	CRGHFHD				
IKROPBUF	USES	CRGBFOPN	CRGBFBEG	CRGSCPAD				
IKROPEN	USES	CRGCMND	CRGFLST	CRGFLOPN	CRGFLISV	CRGFLDTF		
IKROPHF	USES	CRGBKUP	CRGHFDEV	CRGHFFBL	CRGHFLBL	CRGHFLRC		
		CRGHFNTR	CRGHFNBL	CRGHFLU	CRGCMND	CRGFLISV		
		CRGFLDTF	CRGHFCKD	CRGHFCC	CRGHFHD	CRGHFRC		
		CRGHFFBA	CRGHFCC	CRGHFK1				
IKROPLNK	USES	CRGLNK#	CRGFLOPN	CRGFLISV	CRGFLOSV	CRGFLDTF		
IKRPARSE	USES	CRGBFFLG	CRGMATRP	CRGCDLVL	CRGINEOF	CRGINRDR		
		CRGSYNER	CRGCMND	CRGSITP	CRGNOPRT			
IKRPDZAP	USES	CRGINRDR	CRGSITP	CRGSENDP				
IKRPIO	USES	CRGINRDR						
IKRPRCS	USES	CRGNOPRT						
IKRPTLNK	USES	CRGLNK#	CRGFLDTF					
IKRQUEUE	USES	CRGBFBEG	CRGFLOPN					
IKRRAESV	USES	CRGAPOK	CRGCMND					
IKRRALNK	USES	CRGCMND	CRGLMSHP					
IKRRASRV	USES	CRGCMND						
IKRRDIPT	USES	CRGINEOF	CRGFLDTF	CRGSLCT				
IKRRDLV2	USES	CRGCDLVL	CRGINEOF	CRGINRDR	CRGSYNER	CRGCMND		
		CRGSITP						
IKRRDNXT	USES	CRGINEOF	CRGINRDR					
IKRREQFR	USES	CRGHFLRC	CRGCMND	CRGFLDTF	CRGSLCT			
IKRRLZAP	USES	CRGINRDR	CRGSITP	CRGSENDP				
IKRRTC	USES	CRGSCPAD	CRGHFLRC	CRGCMND				
IKRRWBUF	USES	CRGBFBEG	CRGFLOPN	CRGFLDTF				
IKRSAASM	USES	CRGCMND						
IKRSAMNT	USES	CRGCMND	CRGSENDP					
IKRSAPL	USES	CRGMATRP	CRGPINFO	CRGCDLVL	CRGINRDR	CRGSYNER		
		CRGCMND	CRGSITP	CRGSAPL	CRGSLCT			
IKRSERR	USES	CRGPRTAB	CRGINRDR	CRGCMND	CRGSITP			
IKRSLA	USES	CRGSCPAD						
IKRSLZAP	USES	CRGAPOK	CRGINEOF	CRGINRDR	CRGSITP	CRGSENDP		
		CRGRECV	CRGFLOPN					
IKRSPCLS	USES	CRGFLOPN	CRGFLDTF					
IKRSTMTS	USES	CRGPRTCR	CRGPRTL	CRGFLDTF				

IKRSWTHF	USES	CRGHFTB	CRGFLST
IKRTAIL	USES	CRGSENDP	CRGFLOPN CRGFLOSV CRGSLCT
IKRUIRP	USES	CRGMATRP	
IKRUNDO	USES	CRGINEOF	CRGINRDR CRGSITP
IKRUPGR	USES	CRGPRTCR	CRGCMND CRGFLDTF
IKRWKF	USES	CRGFLOPN	CRGFLDTF CRGFLVSM
IKRWOF	USES	CRGFLDTF	
IKRWOR	USES	CRGFLDTF	
IKRWSQ	USES	CRGFLDTF	
IKRWTO	USES	CRGFLDTF	
IKRWTP	USES	CRGPRTCR	CRGPRTL P CRGCMND

Figures 12-7: Sorted by Used Variables

CRGAPOK IS USED BY	IKRRAESV IKRSLZAP
CRGBFBEG IS USED BY	IKRKBKUP IKRFIX IKRFRBUF IKRGTBUF IKRGTTREC IKRGWA IKRMAIN IKRMOD IKROPBUF IKRQUEUE IKRRWBUF
CRGBFFLG IS USED BY	IKRKBKUP IKRMAIN IKRPARSE
CRGBFOPN IS USED BY	IKRGTBUF IKRGTTREC IKRGWA IKROPBUF
CRGBKUP IS USED BY	IKRKBKUP IKRGTBUF IKROPHF
CRGCDLVL IS USED BY	IKRISVBK IKRMAIN IKRPARSE IKRRDLV2 IKRSAPL
CRGCMND IS USED BY	IKRAAPAR IKRAPPLY IKRBIO IKRKBKPR IKRKBKUP IKRCGT IKRCHECK IKRCKD IKRCKLV2 IKRDLIN IKREMU IKREXIT IKRFBA IKRHDLV2 IKRINSTL IKRISVBK IKRMAIN IKROPEN IKROPHF IKRPARSE IKRRAESV IKRRALNK IKRRASRV IKRRDLV2 IKRREQFR IKRRTC IKRSAASM IKRSAMNT IKRSAPL IKRSERR IKRUPGR IKRWTP
CRGFLDPN IS USED BY	IKRCGT
CRGFLDTF IS USED BY	IKRBIO IKRKBKUP IKRCGT IKRCOPY IKRGTHD IKRGTTREC IKRINSTL IKRISVBK IKRMAIN IKROPEN IKROPHF IKROPLNK IKRPTLNK IKRDIPT IKRREQFR IKRRWBUF IKRSPCLS IKRSTMTS IKRUPGR IKRWKF IKRWOF IKRWOR IKRWSQ IKRWTO
CRGFLFLG IS USED BY	IKRMAIN
CRGFLF1D IS USED BY	IKRCLOSE IKRLOAD
CRGFLISV IS USED BY	IKRCGT IKROPEN IKROPHF IKROPLNK
CRGFLLSL IS USED BY	IKRKBKUP IKRCLOSE IKRMAIN IKROPEN IKRSWTHF
CRGFLOPN IS USED BY	IKRCGT IKRCLOSE IKREXIT IKRLOAD IKRMAIN IKROPEN IKROPLNK IKRQUEUE IKRRWBUF IKRSLZAP IKRSPCLS IKRTAIL IKRWKF
CRGFLOSIV IS USED BY	IKRLOAD IKROPLNK IKRTAIL
CRGFLSYS IS USED BY	IKRCLOSE
CRGFLVSM IS USED BY	IKRLOAD IKRWKF
CRGHFCC IS USED BY	IKRNOTE IKROPHF
CRGHFCCC IS USED BY	IKROPHF
CRGHFCKD IS USED BY	IKRGTHD IKRNOTE IKROPHF
CRGHFDEV IS USED BY	IKRGTHD IKRNOTE IKROPHF
CRGHFFBA IS USED BY	IKRGTHD IKROPHF
CRGHFFBL IS USED BY	IKRGTHD IKRNOTE IKROPHF
CRGHFHDI IS USED BY	IKRNOTE IKROPHF
CRGHFK1 IS USED BY	IKROPHF
CRGHFLBL IS USED BY	IKRNOTE IKROPHF
CRGHFLRC IS USED BY	IKRCPIN IKRCREAT IKRCUTHF IKRDEBUG IKROPHF IKRREQFR IKRRTC
CRGHFLU IS USED BY	IKROPHF
CRGHFNBL IS USED BY	IKRNOTE IKROPHF
CRGHFNTR IS USED BY	IKRNOTE IKROPHF

CRGHFRC	IS USED BY	IKROPHF			
CRGHFTB	IS USED BY	IKRSWTHF			
CRGINEOF	IS USED BY	IKRECOR	IKRHDLV2	IKRISVBK	IKRMAIN
		IKRPARSE	IKRRDIPT	IKRRDLV2	IKRRDNXT
		IKRSLZAP	IKRUNDO		
CRGINLP	IS USED BY	IKRBKUP	IKRCPIN	IKREBCK	IKRERST
		IKRESTL			
CRGIPF	IS USED BY	IKREMBDY	IKRESTL		
CRGINRDR	IS USED BY	IKRBKUP	IKRDIAGN	IKRECOR	IKRERDC
		IKRHDLV2	IKRISVBK	IKRMAIN	IKRPARSE
		IKRPDZAP	IKRPIO	IKRRDLV2	IKRRDNXT
		IKRRLZAP	IKRSAPL	IKRSERR	IKRSLZAP
		IKRUNDO			
CRGLNKF	IS USED BY	IKRESTL	IKRISLNK		
CRGLMSHP	IS USED BY	IKRMAIN	IKRRALNK		
CRGLNK#	IS USED BY	IKROPLNK	IKRPTLNK		
CRGMATRP	IS USED BY	IKRAPPLY	IKRBKT	IKRBKUP	IKRCGT
		IKREAPL	IKREARC	IKRECOR	IKREINC
		IKRETLR	IKRISVBK	IKRMAIN	IKRPARSE
		IKRSAPL	IKRUIRP		
CRGNOPRT	IS USED BY	IKREAPL	IKRERVK	IKRISVBK	IKRPARSE
		IKRPRCS			
CRGNTAPE	IS USED BY	IKRBKPR	IKRMAIN		
CRGPINFO	IS USED BY	IKRISVBK	IKRSAPL		
CRGPRTAB	IS USED BY	IKREAPL	IKRISVBK	IKRSERR	
CRGPRTCR	IS USED BY	IKRBKUP	IKRDEBUG	IKREMBDY	IKRLOAD
		IKRMAIN	IKRSTMTS	IKRUPGR	IKRWTP
CRGPRTLPL	IS USED BY	IKRBKUP	IKRDEBUG	IKREMBDY	IKRMAIN
		IKRSTMTS	IKRWTP		
CRGPTFLI	IS USED BY	IKRISVBK	IKRMAIN		
CRGRECVPL	IS USED BY	IKRSLZAP			
CRGSAPL	IS USED BY	IKRBKUP	IKRISVBK	IKRMAIN	IKRSAPL
CRGSCPAD	IS USED BY	IKRBKUP	IKRLOAD	IKRMAIN	IKROPBUF
		IKRRTC	IKRSLA		
CRGSCPAG	IS USED BY	IKRMAIN			
CRGSENDP	IS USED BY	IKRBKPR	IKRBKT	IKREMBDY	IKRINSTL
		IKRISLNK	IKRPDZAP	IKRRLZAP	IKRSAMNT
		IKRSLZAP	IKRTAIL		
CRGSITP	IS USED BY	IKRCKFIX	IKREAPL	IKREARC	IKREBCK
		IKRECOR	IKRECPY	IKRECRE	IKREDMP
		IKREINC	IKRELKP	IKRELST	IKREMBDY
		IKREMRG	IKREPER	IKRERDC	IKRERMV
		IKRERST	IKRERTC	IKRERVK	IKRESEL
		IKRESTL	IKRETLR	IKREUND	IKRISVBK
		IKRMAIN	IKRPARSE	IKRPDZAP	IKRRDLV2
		IKRRLZAP	IKRSAPL	IKRSERR	IKRSLZAP
		IKRUNDO			
CRGSLCT	IS USED BY	IKRAPPLY	IKREXIT	IKRISVBK	IKRRDIPT
		IKRREQFR	IKRSAPL	IKRTAIL	
CRGSLMTP	IS USED BY	IKRAPPLY	IKRISVBK		
CRGSYNER	IS USED BY	IKRISVBK	IKRMAIN	IKRPARSE	IKRRDLV2
		IKRSAPL			

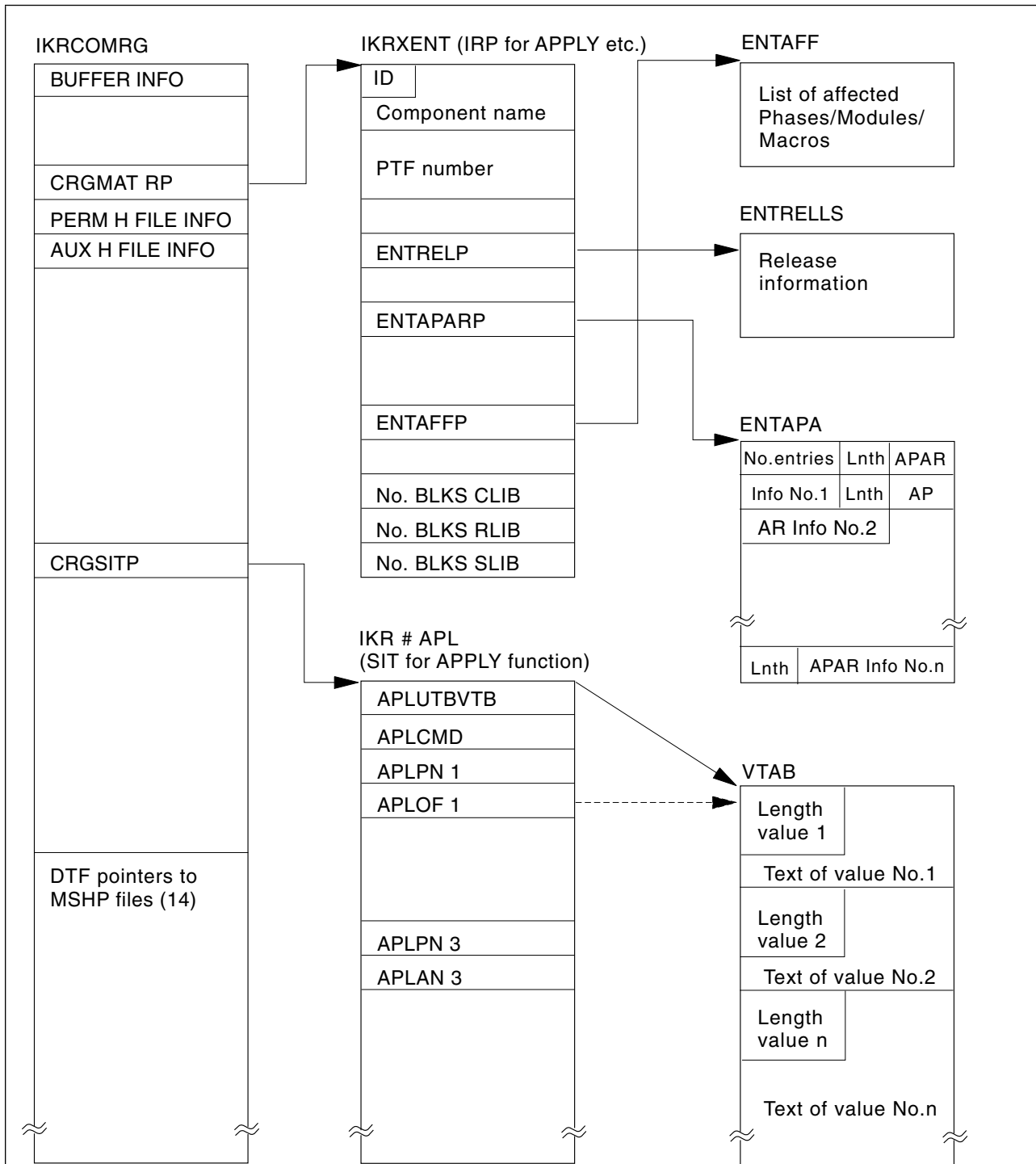
Chapter 4: Data Area Information

This chapter describes the data areas used in MSHP.

Figure 13 on page 274 shows, as an example, the major control blocks used by MSHP for processing the APPLY function. The control block structure for other MSHP functions is similar.

For quick information on data area names and functions, refer to Figure 14 on page 275; this also lists the page number where you find information on the storage layout of a given area.

The data areas are described under "Layout of Data Areas".



Note:

This illustrates how MSHP accesses its primary control blocks; displacements of data areas are not shown. Refer to individual data areas in this chapter for complete description

Figure 13. Example of MSHP Control Block Relationship for the APPLY Function

Name	Function	See Page
Common MSHP Tables and Layouts		
CRGHFCKD	"NOTE FORMAT OF AN CKD H.FILE BLOCK ADDRESS"	283
CRGHFFBA	"NOTE FORMAT OF AN FBA H.FILE BLOCK ADDRESS"	284
IKRBFRO	"BUFFER 0 LAYOUT"	285
IKRRCMRG	"MSHP COMMUNICATION REGION STRUCTURE"	279
IKRIFCTX	"INTERFACE FILE CONTROL TABLE"	286
Internal Representation (IRP)		
IKRXCOR	"CORRECT"	349
IKRXCPY	"BACKUP, COPY, RESTORE IRP"	350
IKRXCRE	"CREATE IRP"	351
IKRXCUM	"ACCEPT, LIST, SELECT PTF IRP"	352
IKRXDMP	"DUMP IRP"	355
IKRXENT	"ARCHIVE, APPLY, REVOKE AND INCORPORATE IRP"	356
ENTRELLS	"LIST OF APPLICABLE RELEASES"	359
ENTCMN	"COMMENT ON PTF"	360
ENTAPA	"LIST OF APARS RESOLVED BY THIS PTF"	361
ENTREQ	"REQUIREMENTS MATRIX"	362
ENTSUP	"LIST OF SUPERSEDED PTFS"	363
ENTAFF	"LIST OF AFFECTED MODULES, AND MACROS"	364
ENTFIX	"FIX DESCRIPTION FOR PHASE, MODULES AND MACROS"	365
ENTIML	"LIST OF INCLUDED MACROS"	366
ENTRLM	"REFERENCE LIST OF INCLUDED MACROS"	367
ENTINF	"LIST OF INFLUENCED MEMBERS"	368
IKRXLKP	"LOOKUP IRP"	369
IKRXMRG	"MERGE IRP"	370
IKRXPER	"PERSONALIZE IRP"	371
IKRXRDC	"RESIDENCE IRP"	372
IKRXRMV	"REMOVE IRP"	373
IKRXRTC	"RETRACE IRP"	374
IKRXSTL	"INSTALL IRP"	375

Figure 14 (Part 1 of 3). Index to Data Areas

Name	Function	See Page
Standardized Internal Table (SIT)		
IKR#AFF	"SIT STRUCTURE FOR THE AFFECTS COMMAND"	289
IKR#ALT	"SIT STRUCTURE FOR THE ALTER COMMAND"	292
IKR#APL	"SIT STRUCTURE FOR THE APPLY COMMAND"	293
IKR#ARC	"SIT STRUCTURE FOR THE ARCHIVE COMMAND"	295
IKR#BCK	"SIT STRUCTURE FOR THE BACKUP COMMAND"	296
IKR#CIS	"SIT STRUCTURE FOR THE COMPRISES COMMAND"	298
IKR#COR	"SIT STRUCTURE FOR THE CORRECT COMMAND"	300
IKR#CPT	"SIT STRUCTURE FOR THE COMPATIBLE COMMAND"	301
IKR#CPY	"SIT STRUCTURE FOR THE COPY COMMAND"	302
IKR#CRE	"SIT STRUCTURE FOR THE CREATE COMMAND"	303
IKR#DAT	"SIT STRUCTURE FOR THE DATA COMMAND"	304
IKR#DEL	"SIT STRUCTURE FOR THE DELETE COMMAND"	305
IKR#DFN	"SIT STRUCTURE FOR THE DEFINE COMMAND"	306
IKR#DMP	"SIT STRUCTURE FOR THE DUMP COMMAND"	308
IKR#EXE	"SIT STRUCTURE FOR THE EXECUTE COMMAND"	309
IKR#GEN	"SIT STRUCTURE FOR THE GENERATE COMMAND"	310
IKR#ICL	"SIT STRUCTURE FOR THE INCLUDE/EXCLUDE COMMAND"	311
IKR#INC	"SIT STRUCTURE FOR THE INCORPORATE COMMAND"	312
IKR#INF	"SIT STRUCTURE FOR THE INFLUENCE COMMAND"	313
IKR#IST	"SIT STRUCTURE FOR THE INSERT COMMAND"	315
IKR#IVL	"SIT STRUCTURE FOR THE INVOLVES COMMAND"	316
IKR#LIST	"SIT STRUCTURE FOR THE LIST COMMAND"	317
IKR#LKP	"SIT STRUCTURE FOR THE LOOKUP COMMAND"	319
IKR#MRG	"SIT STRUCTURE FOR THE MERGE COMMAND"	321
IKR#OR	"SIT STRUCTURE FOR THE OR COMMAND"	322
IKR#PER	"SIT STRUCTURE FOR THE PERSONALIZE COMMAND"	323
IKR#PTF	"SIT STRUCTURE FOR THE PTF COMMAND"	325
IKR#RDC	"SIT STRUCTURE FOR THE RESIDENCE COMMAND"	326
IKR#RMV	"SIT STRUCTURE FOR THE REMOVE COMMAND"	327
IKR#RPL	"SIT STRUCTURE FOR THE REPLACE COMMAND"	329
IKR#RQR	"SIT STRUCTURE FOR THE REQUIRES COMMAND"	330
IKR#RST	"SIT STRUCTURE FOR THE RESTORE COMMAND"	332
IKR#RSTA	"SIT STRUCTURE FOR THE RESTART (ESERV) COMMAND"	334
IKR#RSV	"SIT STRUCTURE FOR THE RESOLVES COMMAND"	335
IKR#RTC	"SIT STRUCTURE FOR THE RETRACE COMMAND"	336
IKR#RVK	"SIT STRUCTURE FOR THE REVOKE COMMAND"	337
IKR#SCN	"SIT STRUCTURE FOR THE SCAN COMMAND"	338
IKR#SEL	"SIT STRUCTURE FOR THE SELECT COMMAND"	339
IKR#STL	"SIT STRUCTURE FOR THE INSTALL COMMAND"	340
IKR#SUP	"SIT STRUCTURE FOR THE SUPERSEDES COMMAND"	343
IKR#TLR	"SIT STRUCTURE FOR THE TAILOR COMMAND"	344
IKR#UND	"SIT STRUCTURE FOR THE UNDO COMMAND"	346
IKR#VER	"SIT STRUCTURE FOR THE VERIFY COMMAND"	347

Figure 14 (Part 2 of 3). Index to Data Areas

Name	Function	See Page
History File Records		
IKRAIP	"STRUCTURE OF APARS-IN-PTF RECORD"	379
IKRAPA	"STRUCTURE OF (LOCAL) APAR FIX RECORD"	380
IKRCOF	"STRUCTURE OF COMPONENT-OF-FEATURE RECORD"	381
IKRCOM	"STRUCTURE OF COMMENT RECORD"	382
IKRFRE	"STRUCTURE OF FREE RECORD"	383
IKRGEN	"STRUCTURE OF GENERATED MEMBER RECORD"	384
IKRHD1	"STRUCTURE OF HEADER 1 RECORD"	386
IKRHD2	"STRUCTURE OF HEADER 2 RECORD"	387
IKRHD3	"STRUCTURE OF HEADER 3 RECORD"	388
IKRHDX	"STRUCTURE OF HEADER EXTENSION RECORD"	389
IKRIMR	"STRUCTURE OF INCLUDED MACRO REFERENCE RECORD"	390
IKRIMT	"STRUCTURE OF INCLUDED MACRO TABLE RECORD"	391
IKRINS	"STRUCTURE OF INSTALLATION RECORD"	392
IKRINX	"STRUCTURE OF INSTALLATION EXTENSION RECORD"	394
IKRLIB	"STRUCTURE OF LIBRARY RECORD"	395
IKRLSI	"STRUCTURE OF LIBRARIAN SEND APPENDAGE INTERFACE"	396
IKRMIA	"STRUCTURE OF MODULE-IN-APAR RECORD"	397
IKRMIP	"STRUCTURE OF MODULES-IN-PTF RECORD"	398
IKRPRD	"STRUCTURE OF PRODUCT RECORD"	399
IKRPRE	"STRUCTURE OF PRE/CO-REQUISITE-INFORMATION RECORD"	401
IKRPTF	"STRUCTURE OF PTF RECORD"	402
IKRPTX	"STRUCTURE OF PTF EXTENSION RECORD"	404
IKRSUP	"STRUCTURE OF SUPERSEDED-PTF'S RECORD"	405

Figure 14 (Part 3 of 3). Index to Data Areas

Layout of Data Areas

The following describes the layout of all data areas.

The data areas are listed in alphabetical sequence by function; e.g. SIT tables, IRP tables, and history file records are listed alphabetically within their respective groups (see data area list).

For each data area, an alphabetic index to field names within the area precedes the storage layout. The offset or reference for locating an equated label within the storage layout is given in decimal for each name that appears in the storage layout. The purpose of this alphabetic index is to enable you to find a label quickly without having to search the storage layout.

For each data area, the storage layout is summarized in tabular form, noting:

- The offset of each field in the data area (expressed in decimal and hexadecimal).
- The length of the field, in number of bytes.
- The symbolic label of the field.
- A brief description of its function, including bit settings (expressed in hexadecimal) when the field serves as an indicator.

Where bit settings are indicated, the symbolic labels that have been equated to the bit settings are given. These labels are used to refer to the numeric values in programs that use the data area, and are included in this manual as aids to help you understand the program listings. The offset given for a field documented in this manner applies only to the symbolic label assigned to the field as a unit. It does not apply to the labels equated to the bit settings (hex values).

Common MSHP Tables and Layouts

IKRCOMRG

OFFSETS	TYPE	LENGTH	NAME	DESCRIPTION
0	(0) STRUCTURE	164	IKRCOMRG	
0	(0) BITSTRING	1	CRGBFFLG	+00 CONTROL FLAGS
	1... ..		CRGBFOPN	80 BUFFER POOL IS OPENED
	.1.. ..		CRGBKUP	40 BACKUP HAS CONTROL
	..1.		CRGLMSHP	20 MSHP RE-LINKED
	...1		CRGNOPRT	10 SUPPRESS PRINTING
 1...		CRGPAGE	08 PAGE NUMBERING
1..		CRGRESET	04 RESET PAGE COUNT
1.		CRGIPF	02 SD PARAMETER (IPF)
1			01 RESERVED
1	(1) A-ADDRESS	3	CRGBFBEG	+01 BUFFERS ORIGIN ADDRESS
4	(4) A-ADDRESS	4	CRGSCPAD	+04 SYSTEM CONTROL PROG'S ADDR.

PRINT TABLE

8	(8) CHARACTER	4	CRGPRTAB	+08 PRINT TABLE
8	(8) SIGNED	2	CRGPRTCR	+08 CURRENT LINE NUMBER
10	(A) SIGNED	2	CRGPRTL P	+0A NUMBER OF LINES PER PAGE

SEMANTIC CHECKER INFORMATION

12	(C) A-ADDRESS	4	CRGMATRP	+0C I.R.P. POINTER
----	---------------	---	----------	--------------------

HISTORY FILE TABLES (1=PERM,2=TEMP)

16	(10) CHARACTER	32	CRGHFTB	+10 HISTORY FILE TABLE
16	(10) BITSTRING	1	CRGHFFLG	+10 H.FILE FLAG
	1... ..		CRGHFDEV	80 ON=FBA, OFF=CKD
	.1.. ..		CRGVSAM	40 ON=VSAM SPACE MGMT
	..11 1111			NOT YET USED
17	(11) UNSIGNED	4	CRGHFFBL	+11 FIRST H.FILE BLOCK 'NOTE' ADDRESS
21	(15) UNSIGNED	4	CRGHFLBL	+15 LAST H.FILE BLOCK 'NOTE' ADDRESS
25	(19) UNSIGNED	3	CRGHFLRC	+19 LAST H.FILE RECORD RBA
28	(1C) UNSIGNED	1	CRGHFNTR	+1C NUMBER OF TRACK/CYL (CKD ONLY)
29	(1D) UNSIGNED	1	CRGHFNBL	+1D NUMBER OF BLOCKS/TRACK (CKD ONLY)
30	(1E) BITSTRING	2	CRGHFLU	+1E ASSIGNED LOGICAL UNIT

PARSER INFORMATION

48	(30) BITSTRING 1... ..	2	CRGPINFO CRGCDLVL	+30 PARSING INFO FLAGS 80 ON: ==> LEVEL 1 COMMAND OFF:==> LEVEL 2 COMMAND
	.1.. ..		CRGINEOF	40 ON: ==> E.O.F. ON RDR CNS
	..1.		CRGINRDR	20 ON: ==> INPUT = READER OFF:==> INPUT = CONSOLE
	...1		CRGSYNER	10 ON: ==> SYNTAX ERROR
 1...		CRGMNIO	08 ON: ==> I/O REQUEST FROM IKRMAIN
111			RESERVED FOR FUTURE USE
	1111 1111			+32 PROCESSED COMMAND CODE
50	(32) CHARACTER	2	CRGCMND	
52	(34) A-ADDRESS	4	CRGSITP	+34 SIT AREA ADDRESS IN CPCB

APPENDAGE INFORMATION

56	(38) A-ADDRESS	4	CRGAPBUF	+38 ADR OF THE CHAIN OF BUFFERS OBTAINED BY AN APPENDAGE ROUTINE THESE BUFFERS MUST FREED BY THE MODULES CALLING A SCP PROGRAM
60	(3C) A-ADDRESS	4	CRGSENDP	+3C SEND APPENDAGE PARAMETER
64	(40) A-ADDRESS	4	CRGRECVP	+40 RECEIVE APPENDAGE PARAMETER
68	(44) SIGNED	2	CRGLNK#	+44 CURRENT NUMBER OF PROCESSED LINK RECORDS
70	(46) BITSTRING 1... ..	1	CRGAPFL	+46 APPENDAGE FLAG
	.1.. ..		CRGAPOK	APPENDAGE OK
	..1.		CRGLIBR	LIBR SERVICED
	...1		CRGWRKFL	WORKAREA RESERVED
 1...		CRGHD1	HEADER AFTER OPEN
1..		CRGFAIL	CALLED PROG. FAILED
11		CRGLNKF	LNK.EDIT FAILED NOT YET USED

HELP-PROMPTING

71	(47) CHARACTER	1		+47 NOT YET USED
72	(48) SIGNED	2	CRGCMDS	+48 VALID COMMANDS(SET)@D14CDMO
74	(4A) SIGNED	2	CRGCMDSV	+4A SAVED FUNCTION
76	(4C) CHARACTER	4		+4C NOT YET USED

MSHP FILE LIST

80	(50) CHARACTER	56	CRGFLLST	+50 FILE LIST (UP TO 14 FILES)
----	----------------	----	----------	--------------------------------

80	(50) BITSTRING	1	CRGFLFLG CRGFLOPN CRGFDPN	+50 FILE FLAG 80 FILE IS OPENED 40 THIS FILE IS DEVICE DEPENDENT
	1... ..			20 THIS IS A SYSTEM DTF
	.1... ..			10 IN DTF SKELETON SAVED
	..1.		CRGFLSYS	08 OUT DTF SKELETON SAVED
	...1		CRGFLISV	04 FORMAT 1 LABEL DELETED
 1...		CRGFLOSV	02 VSAM MANAGED FILE
1..		CRGFLF1D	NOT YET USED
1.		CRGFLVSM	
1			
81	(51) A-ADDRESS	3	CRGFLDTF	+51 DTF ADDRESS

MISCELLANEOUS INFORMATION

136	(88) CHARACTER	4	CRGSELCT	+88 SELECT INFO
136	(88) BITSTRING	1	CRGSAPL CRGSLCT CRGSUPR CRGSCAN CRGPRT CRGINDIR CRGMSHP\$ CRGLINK	+88 CONTROL BITS FOR SELECT 80 SELECT FUNCTION HAS CONTROL 40 SUPPRESS CANCELANION 20 SCAN FOR SERVICE TAPE 10 PRINT PTFNO OF SERVICETAPE 08 INDIRECT PTF APPLICATION 04 \$\$MSHP SUBLIB CREATED 02 LINK INTO \$\$MSHP SUBLIB 01 RESERVED
	1... ..			
	.1... ..			
	..1.			
	...1			
 1...			
1..			
1.			
1			
137	(89) A-ADDRESS	3	CRGSLMTP	+89 OF SELECT I.R.P.
140	(8C) A-ADDRESS	4	CRGASMAD	ASSEMBLER PTR
144	(90) A-ADDRESS	4	CRGXMSG	SERVICETAPE MSGS
148	(94) A-ADDRESS	4	CRGWRKA	WORKAREA POINTER
152	(98) SIGNED	2	CRGRCODE	RETURN CODE
154	(9A) SIGNED	2		RESERVED
156	(9C) UNSIGNED	4	CRGPARM CRGCOPY CRGPRINT CRGCHECK CRGSTACK CRGNTAPE CRGPTFLI	PARAMETER INTERFACE PARM=COPYHIST PARM=NOPRINT PARM=NOINSTALL PARM=PIDSTACKED PARM=NOTAPE PARM=PTFLIST
	1... ..			
	.1... ..			
	..1.			
	...1			
 1...			
1..			
11			
				1111 1111
				1111 1111
	1111 1111			RESERVED
160	(A0) A-ADDRESS	4	CRGINLP	LIBR.GETREC ENTRYPT.@D21BZWG

CROSS-REFERENCE

CRGAPBUF	56 (38)	CRGIPF	0 X'02'
CRGAPFL	70 (46)	CRGLIBR	70 X'40'
CRGAPOK	70 X'80'	CRGLINK	136 X'02'
CRGASMA	140 (8C)	CRGLMSHP	0 X'20'
CRGBFBEG	1 (1)	CRGLNK#	68 (44)
CRGBFFLG	0 (0)	CRGLNKF	70 X'04'
CRGBFOPN	0 X'80'	CRGMATRP	12 (C)
CRGBKUP	0 X'40'	CRGMNIO	48 X'08'
CRGCDLVL	48 X'80'	CRGMSHP\$	136 X'04'
CRGCHECK	156 X'20'	CRGNOPRT	0 X'10'
CRGCMDS	72 (48)	CRGNTAPE	156 X'08'
CRGCMDSV	74 (4A)	CRGPAGE	0 X'08'
CRGCMND	50 (32)	CRGPARM	156 (9C)
CRGCOPY	156 X'80'	CRGPINFO	48 (30)
CRGFAIL	70 X'08'	CRGPRINT	156 X'40'
CRGFLDPN	80 X'40'	CRGPRT	136 X'10'
CRGFLDTF	81 (51)	CRGPRTAB	8 (8)
CRGFLFLG	80 (50)	CRGPRTCR	8 (8)
CRGFLF1D	80 X'04'	CRGPRTL	10 (A)
CRGFLISV	80 X'10'	CRGPTFLI	156 X'04'
CRGFLST	80 (50)	CRGRCODE	152 (98)
CRGFLOPN	80 X'80'	CRGRECV	64 (40)
CRGFLOS	80 X'08'	CRGRESET	0 X'04'
CRGFLSYS	80 X'20'	CRGSAPL	136 (88)
CRGFLVSM	80 X'02'	CRGSCAN	136 X'20'
CRGHD1	70 X'10'	CRGSCPAD	4 (4)
CRGHFDEV	16 X'80'	CRGSELCT	136 (88)
CRGHFFBL	17 (11)	CRGSENDP	60 (3C)
CRGHFFLG	16 (10)	CRGSITP	52 (34)
CRGHFLBL	21 (15)	CRGSLCT	136 X'80'
CRGHFLRC	25 (19)	CRGSLMTP	137 (89)
CRGHFLU	30 (1E)	CRGSTACK	156 X'10'
CRGHFNBL	29 (1D)	CRGSUPR	136 X'40'
CRGHFNTR	28 (1C)	CRGSYNER	48 X'10'
CRGHFTB	16 (10)	CRGVSAM	16 X'40'
CRGINDIR	136 X'08'	CRGWRKA	148 (94)
CRGINEOF	48 X'40'	CRGWRKFL	70 X'20'
CRGINLP	160 (A0)	CRGXMSG	144 (90)
CRGINRDR	48 X'20'	IKRCOMRG	0 (0)

CRGHFCKD

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	4	CRGHFCKD	CKD LAYOUT
0	(0) SIGNED	2	CRGHFCC	CYLINDER NUMBER
2	(2) UNSIGNED	1	CRGHFHD	HEAD NUMBER
3	(3) UNSIGNED	1	CRGHFRC	RECORD NUMBER

CROSS-REFERENCE

CRGHFCC	0 (0)	CRGHFHD	2 (2)
CRGHFCKD	0 (0)	CRGHFRC	3 (3)

CRGHFFBA

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	4	CRGHFFBA	FBA LAYOUT
0	(0) UNSIGNED	3	CRGHFCCC	FBA BLOCK NUMBER
3	(3) UNSIGNED	1	CRGHFK1	RECORD NUMBER IN THIS CONTROL INTERVAL, ALWAYS 1 IN MSHP

CROSS-REFERENCE

CRGHFCCC	0 (0)	CRGHFK1	3 (3)
CRGHFFBA	0 (0)		

IKRBFRO

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	319	BFR0	
0	(0) SIGNED	2	BFRNBR	NUMBER OF BUFFERS
2	(2) SIGNED	2		
4	(4) CHARACTER	20	BFRLSTGT	5 LAST OBTAINED BUFFERS
4	(4) BITSTRING	1	BFRLSTGN	LAST OBTAINED BUFFER NUMBER
5	(5) A-ADDRESS	3	BFRLSTGA	LAST OBTAINED BUFFER ADDRESS
24	(18) A-ADDRESS	4		
28	(1C) A-ADDRESS	4		
32	(20) A-ADDRESS	4	BFRPHFTA	PERM. H.FILE BUFFER TABLE ADD.
36	(24) SIGNED	2	BFRPHFTL	PERM. H.FILE BUFFER TABLE LEN.
38	(26) SIGNED	2		
40	(28) A-ADDRESS	4	BFRTHFTA	TEMP. H.FILE BUFFER TABLE ADD.
44	(2C) SIGNED	2	BFRTHFTL	TEMP. H.FILE BUFFER TABLE LEN.
46	(2E) SIGNED	2		
48	(30) A-ADDRESS	16	BFRFUTUR	
64	(40) BITSTRING	1	BFR0TAB	BUFFER 0 TABLE ENTRY
	1... ..		BFR0WRKA	BUFFER 0 IS ALWAYS A WORKAREA
	.11.			RESERVED
	...1		BFR0ETER	BUFFER 0 IS ALWAYS ETERNAL
 1111			
65	(41) BITSTRING	254	BFR0TAB	BUFFER TABLE
	1... ..		BFRWRKA	BUFFER USED AS WORK AREA
	.1.		BFRLNTRM	LONG TERM H.FILE BUFFER
	..1.		BFRMODIF	MODIFIED H.FILE BUFFER
	...1		BFR0ETER	ETERNAL BUFFER
 1111			RESERVED

CROSS-REFERENCE

BFR0ETER	65 X'10'	BFRPHFTL	36 (24)
BFRFUTUR	48 (30)	BFR0TAB	65 (41)
BFRLNTRM	65 X'40'	BFRTHFTA	40 (28)
BFRLSTGA	5 (5)	BFRTHFTL	44 (2C)
BFRLSTGN	4 (4)	BFRWRKA	65 X'80'
BFRLSTGT	4 (4)	BFR0	0 (0)
BFRMODIF	65 X'20'	BFR0ETER	64 X'10'
BFRNBR	0 (0)	BFR0TAB	64 (40)
BFRPHFTA	32 (20)	BFR0WRKA	64 X'80'

IKRIFCTX

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>	
0	(0)	STRUCTURE	52	IKRIFCTX	
0	(0)	A-ADDRESS	4	IFCEOJ	RETURN ADDRESS AT CALLED PROGRAM EOJ
4	(4)	A-ADDRESS	4	IFCSEND	CURRENT SEND ROUTINE ADDRESS
8	(8)	CHARACTER	4	IFCRCV	
8	(8)	BITSTRING 1... ..	1	IFCRCVFL IFCRCVEN	RECEIVE ROUTINE FLAG RECEIVE ROUTINE ALREADY ENTERED
9	(9)	A-ADDRESS	3	IFCRCVAD	CURRENT RECEIVE ROUTINE ADDRESS
12	(C)	A-ADDRESS	4	IFCOPEN	MSHOPEN ROUTINE ADDRESS
16	(10)	A-ADDRESS	4	IFCCLOSE	MSHCLOSE ROUTINE ADDRESS
20	(14)	A-ADDRESS	4	IFCSAVE	SAVEAREA ADDRESS FOR THE CALLED SYSTEM CONTROL PROGRAM
24	(18)	A-ADDRESS	4	IFCREG13	LIOS GET OR PUT REG13 SAVE AREA
28	(1C)	A-ADDRESS	4	IFCPTPHS	IKRPTMSH OR IKRPHMSH RETURN ADDRESS
32	(20)	A-ADDRESS	4	IFCPHASE	MSHPHASE ROUTINE ADDRESS
36	(24)	A-ADDRESS	4	IFCINREC	ADDRESS OF SENT RECORD
40	(28)	A-ADDRESS	4	IFCOUTRC	ADDRESS OF RECEIVED RECORD
44	(2C)	A-ADDRESS	4	IFCTAPE	MSHTAPE ROUTINE ADDRESS
48	(30)	A-ADDRESS	4	IFCINCL	MSHINCL ROUTINE ADDRESS

CROSS-REFERENCE

IFCCLOSE	16 (10)	IFCRCVAD	9 (9)
IFCEOJ	0 (0)	IFCRCVEN	8 X'80'
IFCINCL	48 (30)	IFCRCVFL	8 (8)
IFCINREC	36 (24)	IFCREG13	24 (18)
IFCOPEN	12 (C)	IFCSAVE	20 (14)
IFCOUTRC	40 (28)	IFCSEND	4 (4)
IFCPHASE	32 (20)	IFCTAPE	44 (2C)
IFCPTPHS	28 (1C)	IKRIFCTX	0 (0)
IFCRCV	8 (8)		

Standardized Internal Table (SIT)

The SIF (Standardized Internal Format) is the control block structure returned from the parser to the MSHP routines. The SIF is constructed of two data areas: the SIT (Standardized Internal Table), and the VTAB (Value Table). For detailed information on the SIF, refer to the parser section of the VSE/Advanced Functions Diagnosis Reference: Service Aids manual.

An SIT is built for each MSHP control statement read and is passed to the semantic checker routine. There the information from the function statement and all its detail statements is combined into an IRP (Internal Representation) of the parameters. The IRP is then passed to the processor routine for execution (see Internal Representation (IRP) later in this chapter).

All SIT tables can be located by the label CRGSITP at displacement x'34' into IKRCOMRG. Within each SIT, at a displacement of x'04', is an identifier composed of the constant MH and a two digit command number which may be used to identify the last SIT built. The following chart correlates the SIT tables to the appropriate identifiers.

SIF Tables

MH01	CREATE
MH02	PERSONALIZE
MH03	COPY
MH04	MERGE
MH05	BACKUP
MH06	RESTORE
MH07	ARCHIVE
MH08	REMOVE
MH09	APPLY
MH10	CORRECT
MH11	DUMP
MH12	INCORPORATE
MH13	RETRACE
MH14	REVOKE
MH15	UNDO
MH16	UPGRADE
MH17	INSTALL
MH18	LIST
MH19	LOOKUP
MH20	SELECT
MH22	RESIDENCE
MH23	TAILOR
MH32	DEFINE
MH33	AFFECTS
MH34	REQUIRES
MH35	SUPERCEDES
MH37	RESOLVES
MH38	ALTER
MH40	INSERT
MH41	DELETE
MH42	REPLACE
MH43	PTF
MH44	OR
MH45	DATA
MH46	INVOLVES
MH47	VERIFY
MH48	COMPRISES
MH49	GENERATE
MH50	EXECUTE
MH51	INFLUENCES
MH52	SCAN
MH53	INCLUDE
MH54	EXCLUDE
MH55	COMPATIBLE
MH56	RESTART

IKR#AFF

OFFSETS	TYPE	LENGTH	NAME	DESCRIPTION
0	(0) STRUCTURE	58	AFFSIT	
0	(0) A-ADDRESS	4	AFFVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	AFFCMD	COMMAND NUMBER ==> MH33

ENTRY FOR PARAMETER PHASES

8	(8) A-ADDRESS	2	AFFPN1	PARAMETER NUMBER ==> PHASES
	1... ..		AFFPNM1	ON IF PARAMETER
	.1... ..		AFFDEF1	ON IF DEFAULT VALUE USED
	..1... ..		AFFPAIR1	ON IF PARM HAS PAIR ATTRIBUTE
10	(A) SIGNED	2	AFFAN1	ALTERNATIVE ALWAYS ==> 1

ENTRY FOR PARAMETER PHASE NAME LIST

12	(C) A-ADDRESS	2	AFFPN2	PARAMETER NUMBER ==> PHASE NAME LIST
14	(E) SIGNED	2	AFFRE2	REPETITION NUMBER
16	(10) SIGNED	2	AFFOF2	OFFSET INTO VALTAB TO LIST

ENTRY FOR PARAMETER EXPAND SIZE INCREMENT

18	(12) A-ADDRESS	2	AFFPN27	PARAMETER NUMBER ==> EXPAND
	1... ..		AFFPNM27	ON IF PARAMETER
	.1... ..		AFFDEF27	ON IF DEFAULT VALUE USED
	..1... ..		AFFPR27	ON IF PARM HAS PAIR ATTRIBUTE
20	(14) SIGNED	2	AFFOF27	OFFSET TO SIZE INCREMENT IN VT

ENTRY FOR PARAMETER MODULES

22	(16) A-ADDRESS	2	AFFPN3	PARAMETER NUMBER ==> MODULES
	1... ..		AFFPNM3	ON IF PARAMETER
	.1... ..		AFFDEF3	ON IF DEFAULT VALUE USED
	..1... ..		AFFPAIR3	ON IF PARM HAS PAIR ATTRIBUTE
24	(18) SIGNED	2	AFFAN3	ALTERNATIVE NUMBER IF NOT DEFAULTED IS 1. IF AFFAN3 = 2 THEN PARAMETERS 4 TO 7 ARE D DEFAULTED TO DUMMY VALUES. THE NEXT PARAMETER IS PARAMETER 8 I.E MACROS.

ENTRY FOR PARAMETER MODULE NAME LIST

26	(1A) A-ADDRESS	2	AFFPN4	PARAMETER NUMBER ==> MODULE NAME LIST

28	(1C) SIGNED	2	AFFRE4	REPETITION NUMBER
30	(1E) SIGNED	2	AFFOF4	OFFSET INTO VTAB ==> MNAME LST

ENTRY FOR PARAMETER LIOCS

32	(20) A-ADDRESS	2	AFFPN5	PARAMETER NUMBER ==> LIOCS
	1... ..		AFFPNM5	ON IF PARAMETER
	.1... ..		AFFDEF5	ON IF DEFAULT VALUE USED
	..1... ..		AFFPAIR5	ON IF PARM HAS PAIR ATTRIBUTE
34	(22) SIGNED	2	AFFAN5	ALTERNATIVE NUMBER ALWAYS = 1

ENTRY FOR PARAMETER CSECT NUMBER

36	(24) A-ADDRESS	2	AFFPN6	PARAMETER NUMBER ==> CSECT NUM
	1... ..		AFFPNM6	ON IF PARAMETER
	.1... ..		AFFDEF6	ON IF DEFAULT VALUE USED
	..1... ..		AFFPAIR6	ON IF PARM HAS PAIR ATTRIBUTE
38	(26) SIGNED	2	AFFOF6	OFFSET INTO VALUE TABLE

ENTRY FOR PARAMETER EXPAND SIZE INCREMENT

40	(28) A-ADDRESS	2	AFFPN7	PARAMETER NUMBER ==> EXPAND
	1... ..		AFFPNM7	ON IF PARAMETER
	.1... ..		AFFDEF7	ON IF DEFAULT VALUE USED
	..1... ..		AFFPAIR7	ON IF PARM HAS PAIR ATTRIBUTE
42	(2A) SIGNED	2	AFFOF7	OFFSET TO SIZE INCREMENT IN VT

ENTRY FOR PARAMETER MACROS

44	(2C) A-ADDRESS	2	AFFPN8	PARAMETER NUMBER ==> MACROS
	1... ..		AFFPNM8	ON IF PARAMETER
	.1... ..		AFFDEF8	ON IF DEFAULT VALUE USED
	..1... ..		AFFPAIR8	ON IF PARM HAS PAIR ATTRIBUTE
46	(2E) SIGNED	2	AFFAN8	ALTERNATIVE NUMBER ALWAYS = 1

ENTRY FOR PARAMETER MACRO NAME LIST

48	(30) A-ADDRESS	2	AFFPN9	PARAMETER NUMBER ==> MACRO NAME LIST
50	(32) SIGNED	2	AFFRE9	REPETITION NUMBER

52	(34) SIGNED	2	AFFOF9	OFFSET INTO VTAB ==> MNAME LST

ENTRY FOR PARAMETER SUBLIBRARY NAME

54	(36) A-ADDRESS	2	AFFPN10	PARAMETER NUMBER ==> SUBLIB
	1... ..		AFFPNM10	ON IF PARAMETER

	.1..	AFFDEF10	ON IF DEFAULT VALUE USED
	..1.	AFFPPR10	ON IF PARM HAS PAIR ATTRIBUTE

56	(38) SIGNED	2 AFFOF10	OFFSET TO SUBLIBRARY NAME IN VT

CROSS-REFERENCE

AFFAN1	10 (A)	AFFPNM1	8 X'80'
AFFAN3	24 (18)	AFFPNM10	54 X'80'
AFFAN5	34 (22)	AFFPNM27	18 X'80'
AFFAN8	46 (2E)	AFFPNM3	22 X'80'
AFFCMD	4 (4)	AFFPNM5	32 X'80'
AFFDEF1	8 X'40'	AFFPNM6	36 X'80'
AFFDEF10	54 X'40'	AFFPNM7	40 X'80'
AFFDEF27	18 X'40'	AFFPNM8	44 X'80'
AFFDEF3	22 X'40'	AFFPN1	8 (8)
AFFDEF5	32 X'40'	AFFPN10	54 (36)
AFFDEF6	36 X'40'	AFFPN2	12 (C)
AFFDEF7	40 X'40'	AFFPN27	18 (12)
AFFDEF8	44 X'40'	AFFPN3	22 (16)
AFFOF10	56 (38)	AFFPN4	26 (1A)
AFFOF2	16 (10)	AFFPN5	32 (20)
AFFOF27	20 (14)	AFFPN6	36 (24)
AFFOF4	30 (1E)	AFFPN7	40 (28)
AFFOF6	38 (26)	AFFPN8	44 (2C)
AFFOF7	42 (2A)	AFFPN9	48 (30)
AFFOF9	52 (34)	AFFPPR10	54 X'20'
AFFPAIR1	8 X'20'	AFFPR27	18 X'20'
AFFPAIR3	22 X'20'	AFFRE2	14 (E)
AFFPAIR5	32 X'20'	AFFRE4	28 (1C)
AFFPAIR6	36 X'20'	AFFRE9	50 (32)
AFFPAIR7	40 X'20'	AFFSIT	0 (0)
AFFPAIR8	44 X'20'	AFFVTB	0 (0)

IKR#ALT

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	18	ALTSIT	
0	(0) A-ADDRESS	4	ALTVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	ALTCMD	COMMAND NUMBER ==> MH38
ENTRY FOR PARAMETER ADDRESS INFO				
8	(8) UNSIGNED	2	ALTPN1	PARAMETER NUMBER ==> ADDR INFO
10	(A) SIGNED	2	ALTOF1	OFFSET TO ADDR INFO IN VTAB
ENTRY FOR PARAMETER OLD TEXT : NEW TEXT INFO				
12	(C) UNSIGNED	2	ALTPN2	PARAMETER NUMBER ==> PAIR
14	(E) SIGNED	2	ALTAN2	ALTERNATE NUMBER, INDICATES THE DATA TYPES OF THE VALUE ENTERED ALTAN2 = 1 ==> ITEM1 QSTR ITEM2 => QSTR ALTAN2 = 2 ==> ITEM1 => QSTR ITEM2 => HEXSTR ALTAN2 = 3 ==> ITEM1 => HEXSTR ITEM2 => QSTR ALTAN2 = 4 ==> ITEM1 => HEXSTR ITEM2 => HEXSTR
16	(10) SIGNED	2	ALTOF2	OFFSET TO PAIR ELEMENTS IN VTAB

CROSS-REFERENCE

ALTAN2	14 (E)	ALTPN1	8 (8)
ALTCMD	4 (4)	ALTPN2	12 (C)
ALTOF1	10 (A)	ALTSIT	0 (0)
ALTOF2	16 (10)	ALTVTB	0 (0)

IKR#APL

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	30	APLSIT	
0	(0) A-ADDRESS	4	APLVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	APLCMD	COMMAND NUMBER ==> MH09

ENTRY FOR PARAMETER COMPONENT : PTF

8	(8) UNSIGNED	2	APLPN1	PARAMETER NUMBER
	1... ..		APLPNM1	ON IF PARAMETER
	.1.. ..		APLDEF1	ON IF DEFAULT VALUE USED
	..1.		APLPAIR1	ON IF PARM HAS PAIR ATTRIBUTE
10	(A) UNSIGNED	2	APLOF1	OFFSET INTO VALUE TABLE

ENTRY FOR PARAMETER RELEASE LIST

12	(C) UNSIGNED	2	APLPN2	PARAMETER NUMBER
	1... ..		APLPNM2	ON IF PARAMETER
	.1.. ..		APLDEF2	ON IF DEFAULT VALUE USED
	..1.		APLPAIR2	ON IF PARM HAS PAIR ATTRIBUTE
14	(E) UNSIGNED	2	APLRE2	REPETITION COUNT
16	(10) SIGNED	2	APLOF2	OFFSET TO RELEASE INFO IN VTAB

ENTRY FOR PARAMETER REVOKABLE IRREV.

18	(12) UNSIGNED	2	APLPN3	PARAMETER NUMBER
	1... ..		APLPNM3	ON IF PARAMETER
	.1.. ..		APLDEF3	ON IF DEFAULT VALUE USED
	..1.		APLPAIR3	ON IF PARM HAS PAIR ATTRIBUTE

20	(14) UNSIGNED	2	APLAN3	ALTERNATIVE NUMBER AN = 1 ==> IRREVOKABLE AN = 2 ==> REVOKABLE
----	---------------	---	--------	--

ENTRY FOR PARAMETER DLIBBUILD NODLIBBUILD

22	(16) UNSIGNED	2	APLPN4	PARAMETER NUMBER
	1... ..		APLPNM4	ON IF PARAMETER
	.1.. ..		APLDEF4	ON IF DEFAULT VALUE USED
	..1.		APLPAIR4	ON IF PARM HAS PAIR ATTRIBUTE

24	(18) UNSIGNED	2	APLAN4	ALTERNATIVE NUMBER AN = 1 ==> NODLIBBUILD AN = 2 ==> DLIBBUILD
----	---------------	---	--------	--

ENTRY FOR PARAMETER NOLINK

26	(1A) UNSIGNED	2	APLPN5	PARAMETER NUMBER
	1... ..		APLPNM5	ON IF PARAMETER
	.1... ..		APLDEF5	ON IF DEFAULT VALUE USED
	..1.		APLPAIR5	ON IF PARM HAS PAIR ATTRIBUTE

28	(1C) UNSIGNED	2	APLAN5	ALTERNATIVE NUMBER ALWAYS 1 AN
				= 1 ==> NODLINK

CROSS-REFERENCE

APLAN3	20 (14)	APLPAIR5	26 X'20'
APLAN4	24 (18)	APLPNM1	8 X'80'
APLAN5	28 (1C)	APLPNM2	12 X'80'
APLCMD	4 (4)	APLPNM3	18 X'80'
APLDEF1	8 X'40'	APLPNM4	22 X'80'
APLDEF2	12 X'40'	APLPNM5	26 X'80'
APLDEF3	18 X'40'	APLPN1	8 (8)
APLDEF4	22 X'40'	APLPN2	12 (C)
APLDEF5	26 X'40'	APLPN3	18 (12)
APLOF1	10 (A)	APLPN4	22 (16)
APLOF2	16 (10)	APLPN5	26 (1A)
APLPAIR1	8 X'20'	APLRE2	14 (E)
APLPAIR2	12 X'20'	APLSIT	0 (0)
APLPAIR3	18 X'20'	APLVTB	0 (0)
APLPAIR4	22 X'20'		

IKR#ARC

OFFSETS	TYPE	LENGTH	NAME	DESCRIPTION
0	(0) STRUCTURE	26	ARCSIT	
0	(0) A-ADDRESS	4	ARCVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	ARCCMD	COMMAND NUMBER ==> MH07
ENTRY FOR PARAMETER PRODUCT NUMBER COMPONENT ID				
8	(8) A-ADDRESS	2	ARCPN1	PARAMETER NUMBER 01
10	(A) SIGNED	2	ARCAN1	AN = 1 ==> FEATURE NUMBER AN = 2 ==> COMPONENT ID
12	(C) SIGNED	2	ARCOF1	OFFSET INTO VTAB ==> COMPONENT NOTE: THE FOLLOWING PARAMETERS MAY ONLY BE PRESENT IF = 2
ENTRY FOR PARAMETER PTF,APAR,PHASE,MODULE,MACRO IF ARCPN23=2 ENTRY FOR PARAMETER SOFTREJECT IF ARCPN23=3				
14	(E) A-ADDRESS 1... .. .1.. ..	2	ARCDEF3	NOT USED HERE NOT USED HERE 1:=SOFTREJECT NOT SPECIFIED (ARCPN23=3)
15	(F) UNSIGNED	1	ARCPN23	PARAMETER NUMBER 2/3
16	(10) SIGNED	2	ARCAN2	ALTERNATIVE PARM NUMBER (IF ARCPN23=2) VAL = 2 ==> PARM = PTF VAL = 3 ==> PARM = APA VAL=4 ==> PARM=PHASE VAL=5 ==> PARM=MODULE VAL=6 ==> PARM=MACRO
ENTRY FOR PARAMETER PTF,APAR				
18	(12) A-ADDRESS	2	ARCPN3	PARAMETER NUMBER 03
20	(14) SIGNED	2	ARCOF3	VALTAB PTR FOR PTF NUM APAR
ENTRY FOR PARAMETER SUBLIB (IF ARCAN2=6) NOT PRESENT OTHERWISE				
22	(16) A-ADDRESS 1... .. .1.. ..	2	ARCPN4 ARCDEF4	PARAMETER NUMBER 04 NOT USED 0:= SUBLIB DEFINED
24	(18) SIGNED	2	ARCOF4	VALTAB PTR FOR SUBLIB

<u>CROSS-REFERENCE</u>			
ARCAN1	10 (A)	ARCOF4	24 (18)
ARCAN2	16 (10)	ARCPN1	8 (8)
ARCCMD	4 (4)	ARCPN23	15 (F)
ARCDEF3	14 X'40'	ARCPN3	18 (12)
ARCDEF4	22 X'40'	ARCPN4	22 (16)
ARCOF1	12 (C)	ARCSIT	0 (0)
ARCOF3	20 (14)	ARCVTB	0 (0)
IKR#BCK			

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0)	STRUCTURE	30 BCKSIT	
0	(0)	A-ADDRESS	4 BCKVTB	POINTER TO VALUE TABLE
4	(4)	CHARACTER	4 BCKCMD	COMMAND NUMBER ==> MH05
ENTRY FOR PARAMETER HISTORY/FEATURE				
8	(8)	A-ADDRESS	2 BCKPN1	ALWAYS 1
10	(A)	SIGNED	2 BCKAN1	ALTERNATE PARM NUMBER VAL = 1 ==> HISTORY VAL = 2 ==> PRODUCT
ENTRY FOR PARAMETER SYSTEM/AUXILIARY (IF BCKAN1=1) ENTRY FOR PRODUCT IDS (IF BCKAN1=2)				
12	(C)	A-ADDRESS	2 BCKPN2	ALWAYS 2
14	(E)	SIGNED	2 BCKAN2	IF BCKAN1=1: ALTERNATE NUMBER VAL = 1 ==> SYSTEM VAL = 2 ==> AUXILIARY
14	(E)	SIGNED	2 BCKRE2	IF BCKAN1=2: REPETITION FACTOR
16	(10)	SIGNED	2 BCKOF2	IF BCKAN1=2: OFFSET INTO VALUE TABLE FOR PRODUCT IDS
ENTRY FOR PARAMETER ID=... (IF BCKAN1=2)				
18	(12)	A-ADDRESS	2 BCKPN3	ALWAYS 3
		1... ..		NOT USED
		.1.. ..	BCKDEF3	1:=NOT SPECIFIED
20	(14)	SIGNED	2 BCKOF3	OFFSET INTO VALUE TABLE FOR ID
ENTRY FOR PARAMETER HEADER=... (IF BCKAN1=2)				
22	(16)	A-ADDRESS	2 BCKPN4	ALWAYS 4
		1... ..		NOT USED
		.1.. ..	BCKDEF4	1:=NOT SPECIFIED
24	(18)	SIGNED	2 BCKOF4	OFFSET INTO VALUE TABLE FOR HEADER INFO

ENTRY FOR PARAMETER PRODUCTION/SERVICE (IF BCKAN1=2)

26	(1A) A-ADDRESS	2	BCKPN5	ALWAYS 5
	1... ..			NOT USED
	.1... ..		BCKDEF5	1:=BOTH LIBS

28	(1C) SIGNED	2	BCKAN5	ALTERNATE PARM NUMBER VAL = 1
				==> PRODUCTION VAL = 2
				==> GENERATION

CROSS-REFERENCE

BCKAN1	10 (A)	BCKOF4	24 (18)
BCKAN2	14 (E)	BCKPN1	8 (8)
BCKAN5	28 (1C)	BCKPN2	12 (C)
BCKCMD	4 (4)	BCKPN3	18 (12)
BCKDEF3	18 X'40'	BCKPN4	22 (16)
BCKDEF4	22 X'40'	BCKPN5	26 (1A)
BCKDEF5	26 X'40'	BCKRE2	14 (E)
BCKOF2	16 (10)	BCKSIT	0 (0)
BCKOF3	20 (14)	BCKVTB	0 (0)

IKR#CIS

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	46	CISSIT	
0	(0) A-ADDRESS	4	CISVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	CISCMD	COMMAND NUMBER ==> MH48
ENTRY FOR PARAMETER COMPONENT				
8	(8) A-ADDRESS	2	CISPN1	VALUE 11 ==> COMPONENT
	1... ..		CISPNM1	ON (PARAMETER)
10	(A) SIGNED	2	CISOF1	OFFSET INTO VALUE TABLE
ENTRY FOR PARAMETER PHASES				
12	(C) A-ADDRESS	2	CISPN3	VALUE 1 ==> PHASES
	1... ..		CISPNM3	ON IF PARAMETER
	.1.. ..		CISDEF3	ON IF DEFAULT VALUE USED
14	(E) SIGNED	2		
ENTRY FOR PARAMETER PHASE NAME LIST				
16	(10) A-ADDRESS	2	CISPN4	VALUE 2 ==> PHASE NAME LIST
18	(12) SIGNED	2	CISRE4	REPETITION NUMBER
20	(14) SIGNED	2	CISOF4	OFFSET INTO VALTAB TO LIST
ENTRY FOR PARAMETER MODULES				
22	(16) A-ADDRESS	2	CISPN5	VALUE 3 ==> MODULES
	1... ..		CISPNM5	ON IF PARAMETER
	.1.. ..		CISDEF5	ON IF DEFAULT VALUE USED
24	(18) SIGNED	2		
ENTRY FOR PARAMETER MODULE NAME LIST				
26	(1A) A-ADDRESS	2	CISPN6	VALUE 4 ==> MODULE NAME LIST
28	(1C) SIGNED	2	CISRE6	REPETITION NUMBER
30	(1E) SIGNED	2	CISOF6	OFFSET INTO VTAB ==> MNAME LST
ENTRY FOR PARAMETER MACROS				
32	(20) A-ADDRESS	2	CISPN7	VALUE 5 ==> MACROS
	1... ..		CISPNM7	ON IF PARAMETER

	.1..		CISDEF7	ON IF DEFAULT VALUE USED
34	(22) SIGNED	2		

ENTRY FOR PARAMETER MACRO NAME LIST

36	(24) A-ADDRESS	2	CISPN8	VALUE 6 ==> MACRO NAME LIST
38	(26) SIGNED	2	CISRE8	REPETITION NUMBER
40	(28) SIGNED	2	CISOF8	OFFSET INTO VTAB ==> MNAME LST

ENTRY FOR PARAMETER SUBLIBRARY NAME

42	(2A) A-ADDRESS 1...1..	2	CISPN9 CISPNM9 CISDEF9	VALUE 7 ==> SUBLIB ON IF PARAMETER ON IF DEFAULT VALUE USED
44	(2C) SIGNED	2	CISOF9	OFFSET TO SUBLIBRARY NAME IN VT

CROSS-REFERENCE

CISCMD	4 (4)	CISPNM9	42 X'80'
CISDEF3	12 X'40'	CISPN1	8 (8)
CISDEF5	22 X'40'	CISPN3	12 (C)
CISDEF7	32 X'40'	CISPN4	16 (10)
CISDEF9	42 X'40'	CISPN5	22 (16)
CISOF1	10 (A)	CISPN6	26 (1A)
CISOF4	20 (14)	CISPN7	32 (20)
CISOF6	30 (1E)	CISPN8	36 (24)
CISOF8	40 (28)	CISPN9	42 (2A)
CISOF9	44 (2C)	CISRE4	18 (12)
CISPNM1	8 X'80'	CISRE6	28 (1C)
CISPNM3	12 X'80'	CISRE8	38 (26)
CISPNM5	22 X'80'	CISSIT	0 (0)
CISPNM7	32 X'80'	CISVTB	0 (0)

IKR#COR

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	16	CORSIT	
0	(0) A-ADDRESS	4	CORVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	CORCMD	COMMAND NUMBER ==> MH10

ENTRY FOR PARAMETER COMPONENT : APAR NUMBER

8	(8) UNSIGNED	2	CORPN1	PARAMETER NUMBER
	1... ..		CORPNM1	ON IF PARAMETER
	.1... ..		CORDEF1	ON IF DEFAULT VALUE USED
	..1... ..		CORPAIR1	ON IF PARM HAS PAIR ATTRIBUTE
10	(A) UNSIGNED	2	COROF1	OFFSET INTO VALUE TABLE

ENTRY FOR PARAMETER REVOKABLE IRREV.

12	(C) UNSIGNED	2	CORPN2	PARAMETER NUMBER
	1... ..		CORPNM2	ON IF PARAMETER
	.1... ..		CORDEF2	ON IF DEFAULT VALUE USED
	..1... ..		CORPAIR2	ON IF PARM HAS PAIR ATTRIBUTE
14	(E) UNSIGNED	2	CORAN2	ALTERNATIVE NUMBER AN = 1 ==> REVOKABLE AN = 2 ==> IRREVOKABLE

CROSS-REFERENCE

CORAN2	14 (E)	CORPNM1	8 X'80'
CORCMD	4 (4)	CORPNM2	12 X'80'
CORDEF1	8 X'40'	CORPN1	8 (8)
CORDEF2	12 X'40'	CORPN2	12 (C)
COROF1	10 (A)	CORSIT	0 (0)
CORPAIR1	8 X'20'	CORVTB	0 (0)
CORPAIR2	12 X'20'		

IKR#CPT

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	18	CPTSIT	
0	(0) A-ADDRESS	4	CPTVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	CPTCMD	COMMAND NUMBER ==> MH55

ENTRY FOR PARAMETER WITH

8	(8) UNSIGNED	2	CPTPN1	ALWAYS 1
10	(A) UNSIGNED	2		NOT USED HERE

ENTRY FOR PARAMETER PRODUCT-ID LIST

12	(C) UNSIGNED	2	CPTPN2	ALWAYS 2
14	(E) UNSIGNED	2	CPTRE2	REPETITION FACTOR
16	(10) UNSIGNED	2	CPTOF2	OFFSET INTO VALUE TABLE

CROSS-REFERENCE

CPTCMD	4 (4)	CPTRE2	14 (E)
CPTOF2	16 (10)	CPTSIT	0 (0)
CPTPN1	8 (8)	CPTVTB	0 (0)
CPTPN2	12 (C)		

IKR#CPY

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0)	STRUCTURE	20 CPYSIT	
0	(0)	A-ADDRESS	4 CPYVTB	POINTER TO VALUE TABLE
4	(4)	CHARACTER	4 CPYCMD	COMMAND NUMBER ==> MH03

ENTRY FOR PARAMETER HISTORY

8	(8)	A-ADDRESS	2 CPYPN1	PARAMETER NUMBER OF 1ST PARM
10	(A)	SIGNED	2 CPYOF1	OFFSET INTO VTAB ==> HISTORY

ENTRY FOR PARAMETER SYSTEM OR AUXILIARY

12	(C)	A-ADDRESS	2 CPYPN2	PARAMETER NUMBER OF 2ND PARM
14	(E)	SIGNED	2 CPYAN2	ALTERNATIVE PARM NUMBER VAL = 1 ==> PARM = SYSTEM VAL = 2 ==> PARM = AUXILIARY

ENTRY FOR PARAMETER SYSTEM OR AUXILIARY

16	(10)	A-ADDRESS	2 CPYPN3	PARAMETER NUMBER OF THIRD PARM
18	(12)	SIGNED	2 CPYAN3	ALTERNATIVE NUM ALWAYS = 1 THIS PARAMETER IS AUXILIARY IF CPYAN2 = 1 ELSE THE PARAME IS SYTEM CPYAN2 = 2

CROSS-REFERENCE

CPYAN2	14	(E)	CPYPN2	12	(C)
CPYAN3	18	(12)	CPYPN3	16	(10)
CPYCMD	4	(4)	CPYSIT	0	(0)
CPYOF1	10	(A)	CPYVTB	0	(0)
CPYPN1	8	(8)			

IKR#CRE

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	16	CRESIT	
0	(0) A-ADDRESS	4	CREVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	CRECMD	COMMAND NUMBER ==> MH01

ENTRY FOR PARAMETER HISTORY

8	(8) A-ADDRESS	2	CREPN1	PARAMETER NUMBER OF 1ST PARM
	1... ..		CREPNM1	ON IF PARAMETER
	.1... ..		CREDEF1	ON IF DEFAULT VALUE USED
	..1... ..		CREPAIR1	ON IF PARM HAS PAIR ATTRIBUTE
10	(A) SIGNED	2	CREOF1	OFFSET INTO VTAB ==> HISTORY

ENTRY FOR PARAMETER SYSTEM OR AUXILIARY

12	(C) A-ADDRESS	2	CREPN2	PARAMETER NUMBER OF 2ND PARM
	1... ..		CREPNM2	ON IF PARAMETER
	.1... ..		CREDEF2	ON IF DEFAULT VALUE USED
	..1... ..		CREPAIR2	ON IF PARM HAS PAIR ATTRIBUTE
14	(E) SIGNED	2	CREAN2	ALTERNATIVE PARM NUMBER VAL = 1 ==> PARM = SYSTEM VAL = 2 ==> PARM = AUXILIARY

CROSS-REFERENCE

CREAN2	14 (E)	CREPNM1	8 X'80'
CRECMD	4 (4)	CREPNM2	12 X'80'
CREDEF1	8 X'40'	CREPN1	8 (8)
CREDEF2	12 X'40'	CREPN2	12 (C)
CREOF1	10 (A)	CRESIT	0 (0)
CREPAIR1	8 X'20'	CREVTB	0 (0)
CREPAIR2	12 X'20'		

IKR#DAT

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	8	DATSIT	
0	(0) A-ADDRESS	4	DATVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	DATCMD	COMMAND NUMBER ==> MH45

<u>CROSS-REFERENCE</u>				
DATCMD	4	(4)	DATVTB	0 (0)
DATSIT	0	(0)		

IKR#DEL

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	12	DELSIT	
0	(0) A-ADDRESS	4	DELVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	DELCMD	COMMAND NUMBER ==> MH41
ENTRY FOR PARAMETER FROM LINE : NUMBER OF LINES				
8	(8) UNSIGNED	2	DELPN1	PARAMETER NUMBER
10	(A) SIGNED	2	DELOF1	OFFSET TO LINE INFO IN VTAB

<u>CROSS-REFERENCE</u>				
DELCMD	4	(4)	DELSIT	0 (0)
DELOF1	10	(A)	DELVTB	0 (0)
DELPN1	8	(8)		

IKR#DFN

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	32	DEFSIT	
0	(0) A-ADDRESS	4	DEFVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	DEF CMD	COMMAND NUMBER ==> MH32

ENTRY FOR PARAMETER LIBRARY INFO

8	(8) A-ADDRESS	2	DEF PN1	ALWAYS 1
10	(A) SIGNED	2	DEF AN1	ALTERNATIVE PARM NUMBER VAL = 5 ==> PARM = HISTORY VAL=7 => PARM=GENFILE

ENTRY FOR PARAMETER SYSTEM/AUXILIARY
ENTRY FOR PARAMETER LIBRARY (IF DEFAN1=7)

12	(C) A-ADDRESS	2	DEF PN2	ALWAYS 2
14	(E) SIGNED	2	DEF AN2	THE FOLLOWING VALUES ARE POSSIBLE: 1.) VALUES IF AND ONLY IF DEFAN1 = 5 ==> HISTORY = 1 ==> AUXILIARY VAL = 2 ==> SYSTEM 2.) VALUE FOR DEFAN1 = 7 VAL=1 ==> PROCEDURE VAL=2 ==> MACRO

ENTRY FOR PARAMETER EXTENT INFORMATION
ENTRY FOR PARAMETER MEMBER NAME (IF DEFAN1=7)

16	(10) A-ADDRESS	2	DEF PN3	ALWAYS 3
	1... ..			NOT USED HERE
	.1... ..		DEFDEF3	1:=NOT SPECIFIED
18	(12) SIGNED	2	DEF OF3	VALTAB OFFSET TO EXTENT INFO

ENTRY FOR PARAMETER SPLIT TRACK INFO
ENTRY FOR PARAMETER SUBLIB (IF DEFAN1=7)

20	(14) A-ADDRESS	2	DEF PN4	ALWAYS 4
	1... ..			NOT USED HERE
	.1... ..		DEFDEF4	1:=NOT SPECIFIED
22	(16) SIGNED	2	DEF OF4	OFFSET TO SPLIT TRACK IN CASE DEFAN1 = 5 ==> HISTORY OR TO SUBLIB NAME IF DEFAN1 = 7

ENTRY FOR PARAMETER UNIT = SYSNNN IN CASE DEFAN1 = 5.

```

-----
24  (18) A-ADDRESS    2  DEFPN5    ALWAYS 5
    1... ..
    .1... ..         DEFDEF5    NOT USED HERE
26  (1A) SIGNED      2  DEFDEF5    1:=NOT SPECIFIED
                                DEFOF5    OFFSET TO UNIT ID

```

ENTRY FOR PARAMETER ID = DATASET NAME IN CASE DEFAN1 = 5.

```

-----
28  (1C) A-ADDRESS    2  DEFPN6    ALWAYS 6
    1... ..
    .1... ..         DEFDEF6    NOT USED HERE
30  (1E) SIGNED      2  DEFOF6    ON IF DEFAULT VALUE USED
                                DEFOF6    OFFSET TO DIRECTORY SIZE IN VT

```

CROSS-REFERENCE

DEFAN1	10 (A)	DEFOF6	30 (1E)
DEFAN2	14 (E)	DEFPN1	8 (8)
DEFKMD	4 (4)	DEFPN2	12 (C)
DEFDEF3	16 X'40'	DEFPN3	16 (10)
DEFDEF4	20 X'40'	DEFPN4	20 (14)
DEFDEF5	24 X'40'	DEFPN5	24 (18)
DEFDEF6	28 X'40'	DEFPN6	28 (1C)
DEFOF3	18 (12)	DEFSIT	0 (0)
DEFOF4	22 (16)	DEFVTB	0 (0)
DEFOF5	26 (1A)		

IKR#DMP

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	16	DMPSIT	
0	(0) A-ADDRESS	4	DMPVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	DMPCMD	COMMAND NUMBER ==> MH11

ENTRY FOR PARAMETER HISTORY

8	(8) A-ADDRESS	2	DMPPN1	PARAMETER NUMBER OF 1ST PARM
	1... ..		DMPPNM1	ON IF PARAMETER
	.1... ..		DMPDEF1	ON IF DEFAULT VALUE USED
	..1... ..		DMPPAIR1	ON IF PARM HAS PAIR ATTRIBUTE
10	(A) SIGNED	2	DMPOF1	OFFSET INTO VTAB ==> HISTORY

ENTRY FOR PARAMETER SYSTEM OR AUXILIARY

12	(C) A-ADDRESS	2	DMPPN2	PARAMETER NUMBER OF 2ND PARM
	1... ..		DMPPNM2	ON IF PARAMETER
	.1... ..		DMPDEF2	ON IF DEFAULT VALUE USED
	..1... ..		DMPPAIR2	ON IF PARM HAS PAIR ATTRIBUTE
14	(E) SIGNED	2	DMPAN2	ALTERNATIVE PARM NUMBER VAL = 1 ==> PARM = SYSTEM VAL = 2 ==> PARM = AUXILIARY

CROSS-REFERENCE

DMPAN2	14 (E)	DMPPNM1	8 X'80'
DMPCMD	4 (4)	DMPPNM2	12 X'80'
DMPDEF1	8 X'40'	DMPPN1	8 (8)
DMPDEF2	12 X'40'	DMPPN2	12 (C)
DMPOF1	10 (A)	DMPSIT	0 (0)
DMPPAIR1	8 X'20'	DMPVTB	0 (0)
DMPPAIR2	12 X'20'		

IKR#EXE

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	18	EXESIT	
0	(0) A-ADDRESS	4	EXEVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	EXECMD	COMMAND NUMBER ==> MH50
ENTRY FOR PARAMETER SYSTEM-CONTROL-PROGRAM-LIST				
8	(8) UNSIGNED	2	EXEPN1	PARAMETER NUMBER ==> SCP LIST
10	(A) SIGNED	2	EXERE1	REPETITION FACTOR
12	(C) SIGNED	2	EXEOF1	OFFSET TO SCP LIST IN VTAB
ENTRY FOR PARAMETER XREF/NOXREF				
14	(E) UNSIGNED	2	EXEPN2	PARAMETER NUMBER ==> XREF/NOXR
16	(10) SIGNED	2	EXEAN2	ALTERNATE NUMBER AN = 1 ==> XREF AN = 2 ==> NOXREF

CROSS-REFERENCE

EXEAN2	16 (10)	EXEPN2	14 (E)
EXECMD	4 (4)	EXERE1	10 (A)
EXEOF1	12 (C)	EXESIT	0 (0)
EXEPN1	8 (8)	EXEVTB	0 (0)

IKR#GEN

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>	
0	(0)	STRUCTURE	20	GENSIT	
0	(0)	A-ADDRESS	4	GENVTB	POINTER TO VALUE TABLE
4	(4)	CHARACTER	4	GENCMD	COMMAND NUMBER ==> MH49
ENTRY FOR PARAMETER PHASE, MODULE, MACRO					
8	(8)	UNSIGNED	2	GENPN1	PARAMETER NUMBER
10	(A)	UNSIGNED	2	GENAN1	ALTERNATE NUMBER AN = 1 ==> PHASE AN = 2 ==> MODULE AN = 3 ==> MACRO
ENTRY FOR PARAMETER MEMBER NAME					
12	(C)	UNSIGNED	2	GENPN2	PARAMETER NUMBER
14	(E)	UNSIGNED	2	GENOF2	OFFSET INTO VALUE TABLE
ENTRY FOR PARAMETER SUBLIB					
16	(10)	UNSIGNED	2	GENPN3	PARAMETER NUMBER
		1... ..		GENDEF3	NOT USED
		.1.. ..		GENOF3	0 := SUBLIB SPECIFIED
18	(12)	UNSIGNED	2	GENOF3	OFFSET INTO VALUE TABLE NOTE: THIS PARAMETER IS PRESENT ONLY IF GENAN1 = 3

CROSS-REFERENCE

GENAN1	10 (A)	GENPN1	8 (8)
GENCMD	4 (4)	GENPN2	12 (C)
GENDEF3	16 X'40'	GENPN3	16 (10)
GENOF2	14 (E)	GENSIT	0 (0)
GENOF3	18 (12)	GENVTB	0 (0)

IKR#ICL

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>	
0	(0)	STRUCTURE	18	ICLSIT	
0	(0)	A-ADDRESS	4	ICLVTB	POINTER TO VALUE TABLE
4	(4)	CHARACTER	4	ICLCMD	COMMAND NUMBER ==> MH53

ENTRY FOR PARAMETER PRODUCT/COMPONENT/PTF/ALL

8	(8)	UNSIGNED	2	ICLPN1	ALWAYS 1
10	(A)	UNSIGNED	2	ICLAN1	INDICATES TYPE: 1 ==> COMPONENT 2 ==> PTF 3 ==> ALL (EXCLUDE ONLY) 4 ==> PRODUCT

ENTRY FOR PARAMETER ID

12	(C)	UNSIGNED	2	ICLPN2	ALWAYS 2
14	(E)	UNSIGNED	2	ICLRE2	REPETITION FACTOR
16	(10)	UNSIGNED	2	ICLOF2	OFFSET INTO VALUE TABLE

CROSS-REFERENCE

ICLAN1	10	(A)	ICLPN2	12	(C)
ICLCMD	4	(4)	ICLRE2	14	(E)
ICLOF2	16	(10)	ICLSIT	0	(0)
ICLPN1	8	(8)	ICLVTB	0	(0)

IKR#INC

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	16	INCSIT	
0	(0) A-ADDRESS	4	INCVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	INCCMD	COMMAND NUMBER ==> MH12
ENTRY FOR PARAMETER COMPONENT ID				
8	(8) UNSIGNED	2	INCPN1	PARAMETER NUMBER ==> COMPONENT
10	(A) SIGNED	2	INCOF1	OFFSET TO COMPONENT ID IN VTAB
ENTRY FOR PARAMETER RELEASE NUMBER				
12	(C) UNSIGNED 1... ..	2	INCPN2	PARAMETER NUMBER ==> RELEASE # NOT USED HERE
14	(E) SIGNED .1... ..	2	INCOF2	1:=NOT SPECIFIED OFFSET TO PTF LIST IN VTAB

CROSS-REFERENCE

INCCMD	4 (4)	INCPN1	8 (8)
INCDDEF2	12 X'40'	INCPN2	12 (C)
INCOF1	10 (A)	INCSIT	0 (0)
INCOF2	14 (E)	INCVTB	0 (0)

IKR#INF

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	42	INFSIT	
0	(0) A-ADDRESS	4	INFVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	INFCMD	COMMAND NUMBER ==> MH51
ENTRY FOR PARAMETER PHASES				
8	(8) A-ADDRESS 1... .. .1... ..	2	INFPN1 INFDEF1	PARAMETER NUMBER ==> PHASES NOT USED 0 := PHASES SPECIFIED
10	(A) SIGNED	2		NOT USED
ENTRY FOR PARAMETER PHASE NAME LIST				
12	(C) A-ADDRESS	2	INFPN2	PARAMETER NUMBER ==> PHASE NAME LIST
14	(E) SIGNED	2	INFRE2	REPETITION NUMBER
16	(10) SIGNED	2	INFOF2	OFFSET INTO VALTAB TO LIST
ENTRY FOR PARAMETER MODULES				
18	(12) A-ADDRESS 1... .. .1... ..	2	INFPN3 INFDEF3	PARAMETER NUMBER ==> MODULES NOT USED 0 := MODULES SPECIFIED
20	(14) SIGNED	2		NOT USED
ENTRY FOR PARAMETER MODULE NAME LIST				
22	(16) A-ADDRESS	2	INFPN4	PARAMETER NUMBER ==> MODULE NAME LIST
24	(18) SIGNED	2	INFRE4	REPETITION NUMBER
26	(1A) SIGNED	2	INFOF4	OFFSET INTO VTAB ==> MNAME LST
ENTRY FOR PARAMETER MACROS				
28	(1C) A-ADDRESS 1... .. .1... ..	2	INFPN5 INFDEF5	PARAMETER NUMBER ==> MACROS NOT USED 0 := MACROS SPECIFIED
30	(1E) SIGNED	2		NOT USED
ENTRY FOR PARAMETER MACRO NAME LIST				

32	(20)	A-ADDRESS	2	INFPN6	PARAMETER NUMBER ==> MACRO NAME LIST
34	(22)	SIGNED	2	INFRE6	REPETITION NUMBER
36	(24)	SIGNED	2	INFOF6	OFFSET INTO VTAB ==> MNAME LST
ENTRY FOR PARAMETER SUBLIBRARY NAME					
38	(26)	A-ADDRESS	2	INFPN7	PARAMETER NUMBER ==> SUBLIB NOT USED
		1... ..		INFDEF7	0 := SUBLIB SPECIFIED
		.1... ..			
40	(28)	SIGNED	2	INFOF7	OFFSET TO SUBLIBRARY NAME IN VT ONLY VALID IF INFDEF5=0

CROSS-REFERENCE

INFCMD	4	(4)	INFPN3	18	(12)
INFDEF1	8	X'40'	INFPN4	22	(16)
INFDEF3	18	X'40'	INFPN5	28	(1C)
INFDEF5	28	X'40'	INFPN6	32	(20)
INFDEF7	38	X'40'	INFPN7	38	(26)
INFOF2	16	(10)	INFRE2	14	(E)
INFOF4	26	(1A)	INFRE4	24	(18)
INFOF6	36	(24)	INFRE6	34	(22)
INFOF7	40	(28)	INFSIT	0	(0)
INFPN1	8	(8)	INFVTB	0	(0)
INFPN2	12	(C)			

IKR#IST

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	12	INSSIT	
0	(0) A-ADDRESS	4	INSVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	INSCMD	COMMAND NUMBER ==> MH40
ENTRY FOR PARAMETER LINE ID AFTER WHICH IS TO BE INSERTED				
8	(8) UNSIGNED	2	INSPN1	PARAMETER NUMBER
10	(A) SIGNED	2	INSOF1	OFFSET TO LINE ID IN VTAB

<u>CROSS-REFERENCE</u>				
INSCMD	4	(4)	INSSIT	0 (0)
INSOF1	10	(A)	INSVTB	0 (0)
INSPN1	8	(8)		

IKR#IVL

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	18	IVLSIT	
0	(0) A-ADDRESS	4	IVLVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	IVLCMD	COMMAND NUMBER ==> MH46
ENTRY FOR PARAMETER LINK				
8	(8) UNSIGNED	2	IVLPN1	PARAMETER NUMBER
10	(A) UNSIGNED	2	IVLAN1	ALTERNATE NUMBER==> ALWAYS 1
ENTRY FOR PARAMETER LINK BOOK LIST				
12	(C) UNSIGNED	2	IVLPN2	PARAMETER NUMBER
14	(E) UNSIGNED	2	IVLRE2	REPETITION COUNT
16	(10) SIGNED	2	IVLOF2	OFFSET TO LINE INFO IN VTAB

CROSS-REFERENCE

IVLAN1	10 (A)	IVLPN2	12 (C)
IVLCMD	4 (4)	IVLRE2	14 (E)
IVLOF2	16 (10)	IVLSIT	0 (0)
IVLPN1	8 (8)	IVLVTB	0 (0)

IKR#LIST

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	32	LISST	
0	(0) A-ADDRESS	4	LISVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	LISCMD	COMMAND NUMBER ==> MH18
ENTRY FOR PARAMETER SERVICETAPE/BACKOUTTAPE				
8	(8) UNSIGNED	2	LISPN1	ALWAYS 1
10	(A) UNSIGNED	2	LISAN1	ALTERNATIVE NUMBER AN = 5 ==> SERVICETAPE AN = 8 ==> BACKOUTTAPE
ENTRY FOR PARAMETER FROMTAPE/FROMDISK				
12	(C) UNSIGNED	2	LISPN11	ALWAYS 11
14	(E) UNSIGNED	2	LISAN11	ALTERNATIVE NUMBER AN = 1 ==> FROMTAPE AN = 2 ==> FROMDISK
ENTRY FOR PARAMETER DOCUMENT/NODOCUMENT				
16	(10) UNSIGNED 1... ..	2	LISPN2	ALWAYS 2 NOT USED HERE
	.1... ..		LISDEF2	1:= NOT SPECIFIED
18	(12) UNSIGNED	2	LISAN2	ALTERNATIVE NUMBER AN = 1 ==> DOCUMENT AN = 2 ==> NODOCUMENT
ENTRY FOR PARAMETER XREF/NOXREF				
20	(14) UNSIGNED	2	LISPN3	ALWAYS 3
22	(16) UNSIGNED	2	LISAN3	ALTERNATIVE NUMBER AN = 1 ==> NOXREF AN = 2 ==> XREF
ENTRY FOR PARAMETER COVER/NOCOVER				
24	(18) UNSIGNED	2	LISPN4	ALWAYS 4
26	(1A) UNSIGNED	2	LISAN4	ALTERNATIVE NUMBER AN = 1 ==> NOCOVER AN = 2 ==> COVER
ENTRY FOR PARAMETER CONTINUOUS/SEPERATE PRESENT ONLY IF LSIAN4=2				
28	(1C) UNSIGNED	2	LISPN5	ALWAYS 5

30 (1E) UNSIGNED 2 LISAN5

ALTERNATIVE NUMBER AN = 1 ==>
CONTINUOUS AN = 2 ==> SEPARATE

CROSS-REFERENCE

LISAN1	10 (A)	LISPN1	8 (8)
LISAN11	14 (E)	LISPN11	12 (C)
LISAN2	18 (12)	LISPN2	16 (10)
LISAN3	22 (16)	LISPN3	20 (14)
LISAN4	26 (1A)	LISPN4	24 (18)
LISAN5	30 (1E)	LISPN5	28 (1C)
LISCMD	4 (4)	LISSIT	0 (0)
LISDEF2	16 X'40'	LISVTB	0 (0)

IKR#LKP

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	28	LKPSIT	
0	(0) A-ADDRESS	4	LKPVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	LKPCMD	COMMAND NUMBER ==> MH19
ENTRY FOR PARAMETER COMPONENT NAME				
8	(8) UNSIGNED	2	LKPPN1	PARAMETER NUMBER
	1... ..		LKPPNM1	ON IF PARAMETER
	.1... ..		LKPDEF1	ON IF DEFAULT VALUE USED
	..1... ..		LKPPAIR1	ON IF PARM HAS PAIR ATTRIBUTE
10	(A) UNSIGNED	2	LKPOF1	OFFSET TO COMP NAME IN VTAB
ENTRY FOR PARAMETER PTF APAR PHASE MODULE MACRO				
12	(C) UNSIGNED	2	LKPPN2	PARAMETER NUMBER
	1... ..		LKPPNM2	ON IF PARAMETER
	.1... ..		LKPDEF2	ON IF DEFAULT VALUE USED
	..1... ..		LKPPAIR2	ON IF PARM HAS PAIR ATTRIBUTE
14	(E) UNSIGNED	2	LKPAN2	ALTERNATE NUMBER AN = 1 ==> PTF AN = 2 ==> APAR AN = 3 ==> PHASE AN = 4 ==> MODULE A 5 ==> FEATURE AN = 6 ==> MACRO
ENTRY FOR PARAMETER PTF-NUM APAR-NUM PHASE NAME ETC				
16	(10) UNSIGNED	2	LKPPN3	PARAMETER NUMBER
18	(12) UNSIGNED	2	LKPOF3	OFFSET INTO VALUE TABLE NOTE: THIS FIELD MUST BE INTERPRETED ACCORDING TO THE FOLLOW CODE LKPAN1 = 1 ==> PTF-NUMBER LKPAN1 = 2 ==> APAR-NUMBER LKPAN1 = 3 ==> PHASE-NAME LKPAN1 = 4 ==> MODULE-NAME LKPAN1 = 5 == FEATURE NUMBER LKPAN1 = 6 ==> MACRO-NAME
ENTRY FOR PARAMETER SUBLIB				
20	(14) UNSIGNED	2	LKPPN4	PARAMETER NUMBER
	1... ..		LKPPNM4	ON IF PARAMETER
	.1... ..		LKPDEF4	ON IF DEFAULT VALUE USED
	..1... ..		LKPPAIR4	ON IF PARM HAS PAIR ATTRIBUTE
22	(16) UNSIGNED	2	LKPOF4	OFFSET INTO VALUE TABLE NOTE: THIS PARAMETER IS PRESENT ONLY IF LKPAN1 = 5

ENTRY FOR PARAMETER DATA

```

-----
24  (18) UNSIGNED      2  LKPPN5      PARAMETER NUMBER
    1... ....
    .1.. ....          LKPDEF5      NOT USED
    ..1. ....          NOT USED
26  (1A) UNSIGNED     2
                                NOT USED

```

CROSS-REFERENCE

LKPAN2	14 (E)	LKPPAIR4	20 X'20'
LKPCMD	4 (4)	LKPPNM1	8 X'80'
LKPDEF1	8 X'40'	LKPPNM2	12 X'80'
LKPDEF2	12 X'40'	LKPPNM4	20 X'80'
LKPDEF4	20 X'40'	LKPPN1	8 (8)
LKPDEF5	24 X'40'	LKPPN2	12 (C)
LKPOF1	10 (A)	LKPPN3	16 (10)
LKPOF3	18 (12)	LKPPN4	20 (14)
LKPOF4	22 (16)	LKPPN5	24 (18)
LKPPAIR1	8 X'20'	LKPSIT	0 (0)
LKPPAIR2	12 X'20'	LKPVTB	0 (0)

IKR#MRG

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	20	MRGSIT	
0	(0) A-ADDRESS	4	MRGVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	MRGCMD	COMMAND NUMBER ==> MH04

ENTRY FOR PARAMETER HISTORY

8	(8) A-ADDRESS	2	MRGPN1	PARAMETER NUMBER OF 1ST PARM
10	(A) SIGNED	2	MRGOF1	OFFSET INTO VTAB ==> HISTORY

ENTRY FOR PARAMETER SYSTEM OR AUXILIARY

12	(C) A-ADDRESS	2	MRGPN2	PARAMETER NUMBER OF 2ND PARM
14	(E) SIGNED	2	MRGAN2	ALTERNATIVE PARM NUMBER VAL = 1 ==> PARM = SYSTEM VAL = 2 ==> PARM = AUXILIARY

ENTRY FOR PARAMETER SYSTEM OR AUXILIARY

16	(10) A-ADDRESS	2	MRGPN3	PARAMETER NUMBER OF THIRD PARM
18	(12) SIGNED	2	MRGAN3	ALTERNATIVE NUM ALWAYS = 1 THIS PARAMETER IS AUXILIARY IF MRGAN2 = 1 ELSE THE PARAME IS SYTEM MRGAN2 = 2

CROSS-REFERENCE

MRGAN2	14 (E)	MRGPN2	12 (C)
MRGAN3	18 (12)	MRGPN3	16 (10)
MRGCMD	4 (4)	MRGSIT	0 (0)
MRGOF1	10 (A)	MRGVTB	0 (0)
MRGPN1	8 (8)		

IKR#OR

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	8	ORSIT	
0	(0) A-ADDRESS	4	ORVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	ORCMD	COMMAND NUMBER ==> MH44

	<u>CROSS-REFERENCE</u>		
ORCMD	4 (4)	ORVTB	0 (0)
ORSIT	0 (0)		

IKR#PER

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	28	PERSIT	
0	(0) A-ADDRESS	4	PERVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	PERCMD	COMMAND NUMBER ==> MH02
ENTRY FOR PARAMETER CUSTOMER NAME				
8	(8) A-ADDRESS	2	PERPN1	PARAMETER NUMBER OF 1ST PARM
	1... ..		PERPNM1	ON IF PARAMETER
	.1... ..		PERDEF1	ON IF DEFAULT VALUE USED
	..1... ..		PERPAIR1	ON IF PARM HAS PAIR ATTRIBUTE
10	(A) SIGNED	2	PEROF1	OFFSET INTO VTAB ==> CUST NAME
ENTRY FOR PARAMETER CUSTOMER ADDRESS				
12	(C) A-ADDRESS	2	PERPN2	PARAMETER NUMBER ==> ADDRESS
	1... ..		PERPNM2	ON IF PARAMETER
	.1... ..		PERDEF2	ON IF DEFAULT VALUE USED
	..1... ..		PERPAIR2	ON IF PARM HAS PAIR ATTRIBUTE
14	(E) SIGNED	2	PEROF2	OFFSET INTO VTAB ==> CUST ADDR
ENTRY FOR PARAMETER CUSTOMER PHONE NUMBER				
16	(10) A-ADDRESS	2	PERPN3	PARAMETER NUMBER ==> PHONE
	1... ..		PERPNM3	ON IF PARAMETER
	.1... ..		PERDEF3	ON IF DEFAULT VALUE USED
	..1... ..		PERPAIR3	ON IF PARM HAS PAIR ATTRIBUTE
18	(12) SIGNED	2	PEROF3	OFFSET INTO VTAB ==> PHONE NUM
ENTRY FOR PARAMETER PROGRAMMER NAME				
20	(14) A-ADDRESS	2	PERPN4	PARAMETER NUMBER ==> PROGRAMMER
	1... ..		PERPNM4	ON IF PARAMETER
	.1... ..		PERDEF4	ON IF DEFAULT VALUE USED
	..1... ..		PERPAIR4	ON IF PARM HAS PAIR ATTRIBUTE
22	(16) SIGNED	2	PEROF4	OFFSET INTO VTAB ==> PGMER NAME
ENTRY FOR PARAMETER ENVIRONMENT INFORMATION				
24	(18) A-ADDRESS	2	PERPN5	PARAMETER NUMBER ==> ENVIRONMEN
	1... ..		PERPNM5	ON IF PARAMETER

.1..
 ..1.
 26 (1A) SIGNED

PERDEF5
 PERPAIR5
 2 PEROF5

ON IF DEFAULT VALUE USED
 ON IF PARM HAS PAIR ATTRIBUTE
 OFFSET INTO VTAB ==> ENVI INFO

CROSS-REFERENCE

PERCMD	4 (4)	PERPAIR4	20 X'20'
PERDEF1	8 X'40'	PERPAIR5	24 X'20'
PERDEF2	12 X'40'	PERPNM1	8 X'80'
PERDEF3	16 X'40'	PERPNM2	12 X'80'
PERDEF4	20 X'40'	PERPNM3	16 X'80'
PERDEF5	24 X'40'	PERPNM4	20 X'80'
PEROF1	10 (A)	PERPNM5	24 X'80'
PEROF2	14 (E)	PERPN1	8 (8)
PEROF3	18 (12)	PERPN2	12 (C)
PEROF4	22 (16)	PERPN3	16 (10)
PEROF5	26 (1A)	PERPN4	20 (14)
PERPAIR1	8 X'20'	PERPN5	24 (18)
PERPAIR2	12 X'20'	PERSIT	0 (0)
PERPAIR3	16 X'20'	PERVTB	0 (0)

IKR#PTF

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	14	PTFSIT	
0	(0) A-ADDRESS	4	PTFVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	PTFCMD	COMMAND NUMBER ==> MH43
ENTRY FOR PARAMETER PTF-NUMBER				
8	(8) A-ADDRESS	2	PTFPN1	ALWAYS 1
10	(A) SIGNED	2	PTFRE1	REPETITION FACTOR
12	(C) SIGNED	2	PTFOF1	OFFSET INTO VTAB ==> PTFNUM

CROSS-REFERENCE

PTFCMD	4 (4)	PTFRE1	10 (A)
PTFOF1	12 (C)	PTFSIT	0 (0)
PTFPN1	8 (8)	PTFVTB	0 (0)

IKR#RDC

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0)	STRUCTURE	32 RDCSIT	
0	(0)	A-ADDRESS	4 RDCVTB	POINTER TO VALUE TABLE
4	(4)	CHARACTER	4 RDCCMD	COMMAND NUMBER ==> MH22

ENTRY FOR PARAMETER PRODUCT

8	(8)	A-ADDRESS	2 RDCPN1	ALWAYS 1
10	(A)	SIGNED	2	NOT USED HERE

ENTRY FOR PARAMETER PRODUCT-ID

12	(C)	A-ADDRESS	2 RDCPN2	ALWAYS 2
14	(E)	UNSIGNED	2 RDCRE2	REPETITION FACTOR
16	(10)	SIGNED	2 RDCOF2	OFFSET INTO VALUE TAB

ENTRY FOR PARAMETER PRODUCTION LIBRARY

18	(12)	A-ADDRESS	2 RDCPN3	ALWAYS 3
		1... ..		NOT USED
		.1... ..	RDCDEF3	1:=NOT SPECIFIED
20	(14)	SIGNED	2 RDCOF3	OFFSET TO LIB.SUBLIB

ENTRY FOR PARAMETER SERVICE LIBRARY

22	(16)	A-ADDRESS	2 RDCPN4	ALWAYS 4
		1... ..		NOT USED
		.1... ..	RDCDEF4	1:=NOT SPECIFIED
24	(18)	SIGNED	2	NOT USED HERE

ENTRY FOR PARAMETER SERVICE LIBRARY NAMES

26	(1A)	A-ADDRESS	2 RDCPN5	ALWAYS 5
28	(1C)	UNSIGNED	2 RDCRE5	REPETITION FACTOR
30	(1E)	SIGNED	2 RDCOF5	OFFSET TO LIB.SUBLIB

<u>CROSS-REFERENCE</u>			
RDCCMD	4 (4)	RDCPN3	18 (12)
RDCDEF3	18 X'40'	RDCPN4	22 (16)
RDCDEF4	22 X'40'	RDCPN5	26 (1A)
RDCOF2	16 (10)	RDCRE2	14 (E)
RDCOF3	20 (14)	RDCRE5	28 (1C)
RDCOF5	30 (1E)	RDCSIT	0 (0)
RDCPN1	8 (8)	RDCVTB	0 (0)
RDCPN2	12 (C)		

IKR#RMV

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	26	REMSIT	
0	(0) A-ADDRESS	4	REMTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	REMCMD	COMMAND NUMBER ==> MH08
ENTRY FOR PARAMETER PRODUCT NUMBER COMPONENT ID				
8	(8) A-ADDRESS	2	REMPN1	PARAMETER NUMBER 01
10	(A) SIGNED	2	REMAN1	ALTERNATE NUMBER AN = 1 ==> OBSOLETE AN = 2 ==> FEATURE NUMBER AN = 3 ==> COMPONENT ID
12	(C) SIGNED	2	REMOF1	OFFSET INTO VTAB NOTE: THE FOLLOWING PARAMETERS ARE PRESENT ONLY IF AN = 3
ENTRY FOR PARAMETER RELEASE,PTF,APAR,PHASE,MODULE,MACRO				
14	(E) A-ADDRESS	2	REMPN2	PARAMETER NUMBER 02
	1... ..			NOT USED
	.1.. ..		REMDEF2	0:=PARAMETER SPECIFIED
16	(10) SIGNED	2	REMAN2	ALTERNATIVE PARM NUMBER VAL = 1 ==> PARM = RELEASE VAL = 2 ==> PARM = PTF VAL = 3 == PARM = APAR VAL=4 ==> PARM=PHASE VAL=5 ==> PARM=MODULE VAL=6 ==> PARM=MACRO
ENTRY FOR PARAMETER REL,PTF,APAR,ETC. INFO				
18	(12) A-ADDRESS	2	REMPN3	PARAMETER NUMBER 03
20	(14) SIGNED	2	REMOF3	VALTAB PTR FOR PTF NUM APAR
ENTRY FOR PARAMETER RELEASE (IF REMAN2=2 OR 3) ENTRY FOR PARAMETER SUBLIB (IF REMAN2=6) NOT PRESENT OTHERWISE				

22	(16) A-ADDRESS	2	REMPN4	PARAMETER NUMBER 04
	1... ..			NOT USED
	.1... ..		REMDEF4	0:=PARAMETER SPECIFIED

24	(18) SIGNED	2	REMOF4	VTAB PTR FOR RELEASE INFO

CROSS-REFERENCE

REMAN1	10 (A)	REMOF4	24 (18)
REMAN2	16 (10)	REMPN1	8 (8)
REMCMD	4 (4)	REMPN2	14 (E)
REMDEF2	14 X'40'	REMPN3	18 (12)
REMDEF4	22 X'40'	REMPN4	22 (16)
REMOF1	12 (C)	REMSIT	0 (0)
REMOF3	20 (14)	REMTB	0 (0)

IKR#RPL

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	12	REPSIT	
0	(0) A-ADDRESS	4	REPVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	REPCMD	COMMAND NUMBER ==> MH42
ENTRY FOR PARAMETER FROM LINE : NUMBER OF LINES				
8	(8) UNSIGNED	2	REPPN1	PARAMETER NUMBER
10	(A) SIGNED	2	REPOF1	OFFSET TO LINE INFO IN VTAB

<u>CROSS-REFERENCE</u>				
REPCMD	4	(4)	REPSIT	0 (0)
REPOF1	10	(A)	REPVTB	0 (0)
REPPN1	8	(8)		

IKR#RQR

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	42	REQSIT	
0	(0) A-ADDRESS	4	REQVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	REQCMD	COMMAND NUMBER ==> MH34

ENTRY FOR PARAMETER ==> COMPONENT NAME

8	(8) A-ADDRESS	2	REQPN1	PARAMETER NUMBER OF 1ST PARM
	1... ..		REQPNM1	ON IF PARAMETER
	.1... ..		REQDEF1	ON IF DEFAULTED
	..1... ..		REQPAIR1	ON IF PARM HAS PAIR ATTRIBUTE
10	(A) SIGNED	2	REQOF1	OFFSET INTO VTAB ==> COMP NAME

ENTRY FOR PARAMETER PREREQUISITE

12	(C) A-ADDRESS	2	REQPN2	PARAMETER NUMBER ==> PRE
	1... ..		REQPNM2	ON IF PARAMETER
	.1... ..		REQDEF2	ON IF DEFAULT VALUE USED
	..1... ..		REQPAIR2	ON IF PARM HAS PAIR ATTRIBUTE
14	(E) SIGNED	2	REQAN2	ALTERNATIVE NUMBER ALWAYS = 1

ENTRY FOR PARAMETER PREREQUISITE LIST

16	(10) UNSIGNED	2	REQPN3	PARAMETER NUMBER ==> PREREQ. PTF LIST
18	(12) SIGNED	2	REQRE3	REPETITION FACTOR
20	(14) SIGNED	2	REQOF3	OFFSET TO PRE PTF LIST IN VTAB

ENTRY FOR PARAMETER COREQUISITE

22	(16) UNSIGNED	2	REQPN4	PARAMETER NUMBER ==> CO
	1... ..		REQPNM4	ON IF PARAMETER
	.1... ..		REQDEF4	ON IF DEFAULT VALUE USED
	..1... ..		REQPAIR4	ON IF PARM HAS PAIR ATTRIBUTE
24	(18) SIGNED	2	REQAN4	ALTERNATIVE NUMBER ALWAYS = 1

ENTRY FOR PARAMETER COREQUISITE LIST

26	(1A) UNSIGNED	2	REQPN5	PARAMETER NUMBER ==> COREQ. PTF LIST
28	(1C) SIGNED	2	REQRE5	REPETITION FACTOR
30	(1E) SIGNED	2	REQOF5	OFFSET TO CO PTF LIST IN VTAB

ENTRY FOR PARAMETER NEGATIVE PREREQUISITE

```

-----
32  (20) A-ADDRESS      2  REQPN6      PARAMETER NUMBER ==> NOT
    1... ....          REQPNM6      ON IF PARAMETER
    .1.. ....          REQDEF6      ON IF DEFAULT VALUE USED
    ..1. ....          REQPAIR6     ON IF PARM HAS PAIR ATTRIBUTE
34  (22) SIGNED         2  REQAN6      ALTERNATIVE NUMBER ALWAYS = 1

```

ENTRY FOR PARAMETER NEGATIVE PREREQUISITE LIST

```

-----
36  (24) UNSIGNED      2  REQPN7      PARAMETER NUMBER ==>
                                NEG.PREREQ PTF LIST
38  (26) SIGNED        2  REQRE7      REPETITION FACTOR
-----
40  (28) SIGNED        2  REQOF7      OFFSET TO -PRE PTF LIST IN
                                VTAB

```

CROSS-REFERENCE

REQAN2	14 (E)	REQPNM1	8 X'80'
REQAN4	24 (18)	REQPNM2	12 X'80'
REQAN6	34 (22)	REQPNM4	22 X'80'
REQCMD	4 (4)	REQPNM6	32 X'80'
REQDEF1	8 X'40'	REQPN1	8 (8)
REQDEF2	12 X'40'	REQPN2	12 (C)
REQDEF4	22 X'40'	REQPN3	16 (10)
REQDEF6	32 X'40'	REQPN4	22 (16)
REQOF1	10 (A)	REQPN5	26 (1A)
REQOF3	20 (14)	REQPN6	32 (20)
REQOF5	30 (1E)	REQPN7	36 (24)
REQOF7	40 (28)	REQRE3	18 (12)
REQPAIR1	8 X'20'	REQRE5	28 (1C)
REQPAIR2	12 X'20'	REQRE7	38 (26)
REQPAIR4	22 X'20'	REQSIT	0 (0)
REQPAIR6	32 X'20'	REQVTB	0 (0)

IKR#RST

OFFSETS	TYPE	LENGTH	NAME	DESCRIPTION
0	(0) STRUCTURE	38	RSTSIT	
0	(0) A-ADDRESS	4	RSTVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	RSTCMD	COMMAND NUMBER ==> MH06

ENTRY FOR PARAMETER HISTORY/PRODUCT/SYSRES

8	(8) A-ADDRESS	2	RSTPN1	ALWAYS 1
10	(A) SIGNED	2	RSTAN1	ALTERNATIVE PARM NUMBER VAL = 1 ==> HISTORY VAL = 2 ==> PRODUCT VAL = 3 ==> SYSRES

ENTRY FOR PARAMETER HISTORY TYPE (FOR RSTAN1=1)
ENTRY FOR PARAMETER TAPE-ID (FOR RSTAN1=2)

12	(C) A-ADDRESS	2	RSTPN2	ALWAYS 2 NOT USED HERE
	1... ..		RSTDEF2	1:=NOT SPECIFIED
14	(E) SIGNED	2	RSTAN2	FOR RSTAN1=1: VAL = 1 ==> SYSTEM VAL = 2 ==> AUXILIARY
14	(E) SIGNED	2	RSTOF2	FOR RSTAN1=2,3: VALUE TABLE OFFSET

ENTRY FOR PARAMETER TARGET LIBRARY TYPE (RSTAN1=2,3)

16	(10) A-ADDRESS	2	RSTPN3	ALWAYS 3
18	(12) SIGNED	2	RSTAN3	ALTERNATIVE NUMBER: VAL = 1 ==> COMMON LIBRARY VAL = 2 ==> PRODUCTION LIB

ENTRY FOR PARAMETER COMMON/PRODUCTION LIB
(RSTAN1=2,3)

20	(14) A-ADDRESS	2	RSTPN4	ALWAYS 4
22	(16) SIGNED	2	RSTOF4	OFFSET INTO VALUE TABLE: COMMON LIB (IF RSTAN3=1) PRODUCTION LIB (IF RSTAN3=2)

ENTRY FOR PARAMETER SERVICE (IF RSTAN3=2)

24	(18) A-ADDRESS	2	RSTPN5	ALWAYS 5
26	(1A) SIGNED	2		NOT USED HERE

ENTRY FOR PARAMETER INTO (IF RSTAN3=2)

28	(1C) A-ADDRESS	2	RSTPN6	ALWAYS 6
30	(1E) SIGNED	2		NOT USED HERE

ENTRY FOR PARAMETER SERVICE LIBRARY (IF RSTAN3=2)

32	(20) A-ADDRESS	2	RSTPN7	ALWAYS 7
34	(22) UNSIGNED	2	RSTRE7	REPETITION FACTOR

36	(24) SIGNED	2	RSTOF7	OFFSET INTO VALUE TAB
----	-------------	---	--------	-----------------------

CROSS-REFERENCE

RSTAN1	10 (A)	RSTPN2	12 (C)
RSTAN2	14 (E)	RSTPN3	16 (10)
RSTAN3	18 (12)	RSTPN4	20 (14)
RSTCMD	4 (4)	RSTPN5	24 (18)
RSTDEF2	12 X'40'	RSTPN6	28 (1C)
RSTOF2	14 (E)	RSTPN7	32 (20)
RSTOF4	22 (16)	RSTRE7	34 (22)
RSTOF7	36 (24)	RSTSIT	0 (0)
RSTPN1	8 (8)	RSTVTB	0 (0)

IKR#RSTA

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	12	RSTASIT	
0	(0) A-ADDRESS	4	RSTAVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	RSTACMD	COMMAND NUMBER ==> MH56
ENTRY FOR PARAMETER LINE ID AFTER WHICH IS TO BE INSERTED				
8	(8) UNSIGNED	2	RSTAPN1	PARAMETER NUMBER
10	(A) SIGNED	2	RSTAOF1	OFFSET TO LINE ID IN VTAB

<u>CROSS-REFERENCE</u>				
RSTACMD	4	(4)	RSTASIT	0 (0)
RSTAOF1	10	(A)	RSTAVTB	0 (0)
RSTAPN1	8	(8)		

IKR#RSV

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	22	RESSIT	
0	(0) A-ADDRESS	4	RESVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	RESCMD	COMMAND NUMBER ==> MH37

ENTRY FOR PARAMETER COMMENT

8	(8) UNSIGNED	2	RESPN1	PARAMETER NUMBER ==> COMMENT
	1... ..		RESPNM1	ON IF PARAMETER
	.1... ..		RESDEF1	ON IF DEFAULT VALUE USED
	..1... ..		RESPAIR1	ON IF PARM HAS PAIR ATTRIBUTE
10	(A) SIGNED	2	RESOF1	OFFSET TO COMMENT IN VALUE TAB

ENTRY FOR PARAMETER APAR

12	(C) UNSIGNED	2	RESPN2	PARAMETER NUMBER ==> APAR
	1... ..		RESPNM2	ON IF PARAMETER
	.1... ..		RESDEF2	ON IF DEFAULT VALUE USED
	..1... ..		RESPAIR2	ON IF PARM HAS PAIR ATTRIBUTE
14	(E) SIGNED	2	RESAN2	ALTERNATIVE NUMBER ALWAYS = 1

ENTRY FOR PARAMETER APAR LIST

16	(10) UNSIGNED	2	RESPN3	PARAMETER NUMBER ==> APAR LST
18	(12) SIGNED	2	RESRE3	REPETITION FACTOR
20	(14) SIGNED	2	RESOF3	OFFSET TO APAR LIST IN VTAB

CROSS-REFERENCE

RESAN2	14 (E)	RESPNM1	8 X'80'
RESCMD	4 (4)	RESPNM2	12 X'80'
RESDEF1	8 X'40'	RESPN1	8 (8)
RESDEF2	12 X'40'	RESPN2	12 (C)
RESOF1	10 (A)	RESPN3	16 (10)
RESOF3	20 (14)	RESRE3	18 (12)
RESPAIR1	8 X'20'	RESSIT	0 (0)
RESPAIR2	12 X'20'	RESVTB	0 (0)

IKR#RTC

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	16	RTCSIT	
0	(0) A-ADDRESS	4	RTCVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	RTCCMD	COMMAND NUMBER ==> MH13

ENTRY FOR PARAMETER LIST

8	(8) A-ADDRESS	2	RTCPN1	PARAMETER NUMBER
	1... ..		RTCPNM1	ON IF PARAMETER
	.1.. ..		RTCDEF1	ON IF DEFAULT VALUE USED
	..1.		RTCPAIR1	ON IF PARM HAS PAIR ATTRIBUTE
10	(A) SIGNED	2	RTCAN1	ALTERNATIVE PARM NUMBER AN = 1 ==> PARM = COMPONENT AN = 2 ==> PARM = FEATURES AN = ==> PARM = PTFS AN = 4 ==> PARM = APARS AN = 5 ==> PARM = MODULES

ENTRY FOR PARAMETER ID=COMPONENT-ID

12	(C) A-ADDRESS	2	RTCPN2	PARAMETER NUMBER
	1... ..		RTCPNM2	ON IF PARAMETER
	.1.. ..		RTCDEF2	ON IF DEFAULT VALUE USED
	..1.		RTCPAIR2	ON IF PARM HAS PAIR ATTRIBUTE
14	(E) SIGNED	2	RTCOF2	OFFSET INTO VTAB ==> COMP-ID

CROSS-REFERENCE

RTCAN1	10 (A)	RTCPNM1	8 X'80'
RTCCMD	4 (4)	RTCPNM2	12 X'80'
RTCDEF1	8 X'40'	RTCPN1	8 (8)
RTCDEF2	12 X'40'	RTCPN2	12 (C)
RTCOF2	14 (E)	RTCSIT	0 (0)
RTCPAIR1	8 X'20'	RTCVTB	0 (0)
RTCPAIR2	12 X'20'		

IKR#RVK

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	12	RVKSIT	
0	(0) A-ADDRESS	4	RVKVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	RVKCMD	COMMAND NUMBER ==> MH14

ENTRY FOR PARAMETER COMPONENT : PTF-NUMBER

8	(8) UNSIGNED	2	RVKPN1	PARAMETER NUMBER
10	(A) UNSIGNED	2	RVKOF1	OFFSET INTO VALUE TABLE

	<u>CROSS-REFERENCE</u>			
RVKCMD	4	(4)	RVKSIT	0 (0)
RVKOF1	10	(A)	RVKVTB	0 (0)
RVKPN1	8	(8)		

IKR#SCN

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0)	22	SCNSIT	
0	(0)	4	SCNVTB	POINTER TO VALUE TABLE
4	(4)	4	SCNCMD	COMMAND NUMBER ==> MH52

ENTRY FOR PARAMETER OFFSET ADDRESS

8	(8)	2	SCNPN2	PARAMETER NUMBER ==> OFFSET
	1... ..		SCNPNM2	ON IF PARAMETER
	.1... ..		SCNDEF2	ON IF DEFAULT USED
10	(A)	2	SCNOF2	OFFSET TO ADDR INFO IN VTAB

ENTRY FOR PARAMETER SEARCH ARGUMENT KEY

12	(C)	2	SCNPN3	PARAMETER NUMBER ==> KEY-ARG
	1... ..		SCNPNM3	ON IF PARAMETER
	.1... ..		SCNDEF3	ON IF DEFAULT USED
14	(E)	2	SCNAN3	OFFSET TO KEY INFO IN VTAB

ENTRY FOR PARAMETER SEARCH ARGUMENT

16	(10)	2	SCNPN1	PARAMETER NUMBER ==> ARGUMENT
	1... ..		SCNPNM1	ON IF PARAMETER
	.1... ..		SCNDEF1	ON IF DEFAULT USED
18	(12)	2	SCNAN1	ALTERNATE NUMBER, INDICATES THE DATA TYPES OF THE VALUE ENTERED SCNAN1 = 1 ==> ITEM HEXSTR SCNAN1 = 2 ==> ITEM = QSTR

20	(14)	2	SCNOF1	OFFSET TO ARGUMENT INF IN VTAB
----	------	---	--------	--------------------------------

CROSS-REFERENCE

SCNAN1	18 (12)	SCNPNM1	16 X'80'
SCNAN3	14 (E)	SCNPNM2	8 X'80'
SCNCMD	4 (4)	SCNPNM3	12 X'80'
SCNDEF1	16 X'40'	SCNPN1	16 (10)
SCNDEF2	8 X'40'	SCNPN2	8 (8)
SCNDEF3	12 X'40'	SCNPN3	12 (C)
SCNOF1	20 (14)	SCNSIT	0 (0)
SCNOF2	10 (A)	SCNVTB	0 (0)

IKR#SEL

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0)	STRUCTURE	16 SELSIT	
0	(0)	A-ADDRESS	4 SELVTB	POINTER TO VALUE TABLE
4	(4)	CHARACTER	4 SELCMD	COMMAND NUMBER ==> MH20
ENTRY FOR PARAMETER GENFILE				
8	(8)	UNSIGNED	2 SELPN1	PARAMETER NUMBER
10	(A)	UNSIGNED	2 SELAN1	ALTERNATE NUMBER GENFILE ==> 1
ENTRY FOR PARAMETER COMPONENT				
12	(C)	UNSIGNED	2 SELPN2	PARAMETER NUMBER
				NOT USED
				1... ..
				.1... ..
				ON IF DEFAULT VALUE USED
				SELDEF2
				NOT USED
				..1... ..
14	(E)	UNSIGNED	2 SELAN2	NOT USED IF SELAN1=1
14	(E)	UNSIGNED	2 SELOF2	OFFSET TO COMPONENT IN VALUE TABLE. PRESENT ONLY FOR SELAN1=2

CROSS-REFERENCE

SELAN1	10 (A)	SELPN1	8 (8)
SELAN2	14 (E)	SELPN2	12 (C)
SELCMD	4 (4)	SELSIT	0 (0)
SELDEF2	12 X'40'	SELVTB	0 (0)
SELOF2	14 (E)		

IKR#STL

OFFSETS	TYPE	LENGTH	NAME	DESCRIPTION
0	(0) STRUCTURE	58	ISTSIT	
0	(0) A-ADDRESS	4	ISTVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	ISTCMD	COMMAND NUMBER ==> MH17

ENTRY FOR PARAMETER SYSRES/PRODUCT/SERVICE/BACKOUT

8	(8) UNSIGNED	2	ISTPN1	ALWAYS 1
10	(A) UNSIGNED	2	ISTAN1	ALTERNATE NUMBER AN = 1 ==> SYSRES AN = 2 ==> PRODUCT AN = 3 ==> SERVICE AN = 4 ==> BACKOUT

ENTRY FOR PARAMETER FROMTAPE/FROMDISK (IF Istan1=1/2)
ENTRY FOR PARAMETER REVOKABLE/RESTART (IF Istan1=3/4)

12	(C) UNSIGNED	2	ISTPN2	ALWAYS 2 NOT USED HERE
	1... ..			
	.1.. ..		ISTDEF2	1:=IRREVOKABLE (IF Istan1=3)
14	(E) UNSIGNED	2	ISTAN2	ALTERNATIVE NUMBER AN=1 ==> FROMTAPE (ISTAN1=1/2) AN=2 ==> FROMDISK (ISTAN1=1/2) AN= ==> REVOKABLE(ISTAN1=3/4) AN=2 ==> RESTART (ISTAN1=3/4)

ENTRY FOR PARAMETER TAPELABEL IF AN1=1,2 PROD,SYSTEM
ENTRY FOR PARAMETER FROMT/FROMD IF AN1=3 (SERVICE)

16	(10) A-ADDRESS	2	ISTPN21	PARAMETER NUMBER NOT USED HERE
	1... ..			
	.1.. ..		ISTDEF21	1:=NOT SPECIFIED
18	(12) SIGNED	2	ISTAN21	ALTERNATIVE PARM NUMB 1:=TL 2:=TAPELABEL AN1=3 1=FROMT 2=FROMD

ENTRY FOR TAPE FILENAME VALUE IF AN1=1/2
ENTRY FOR PARAMETER IPF IF AN1=3 SERVICE

20	(14) A-ADDRESS	2	ISTPN22	PARAMETER NUMBER NOT USED HERE
	1... ..			
	.1.. ..		ISTDEF22	IF AN1=3 NOT SPECIF.
22	(16) SIGNED	2	ISTOF22	OFFSET IN VALUE TABLE

ENTRY FOR PARAMETER ID IF AN1=1,2 PROD,SYSTEM
ENTRY FOR PARAMETER TAPES=... IF AN1=3 SERVICE

```

-----
24  (18) UNSIGNED      2  ISTPN3      PARAMETER NUMBER
    1... ....
    .1.. ....          ISTDEF3      NOT USED HERE
26  (1A) UNSIGNED      2  ISTOF3      1:=NOT SPECIFIED
                                VALUE TABLE OFFSET

```

ENTRY FOR COMMON TARGET LIBRARY

```

-----
28  (1C) UNSIGNED      2  ISTPN4      ALWAYS 4
    1... ....
    .1.. ....          ISTDEF4      NOT USED HERE
30  (1E) UNSIGNED      2  ISTOF4      1:=NOT SPECIFIED
                                VALUE TABLE OFFSET

```

ENTRY FOR COMMON TARGET LIBRARY NAME

```

-----
32  (20) UNSIGNED      2  ISTPN5      ALWAYS 5
    1... ....
    .1.. ....          ISTDEF5      NOT USED HERE
34  (22) UNSIGNED      2  ISTOF5      1:=NOT SPECIFIED
                                VALUE TABLE OFFSET

```

ENTRY FOR PARAMETER PRODUCTION

```

-----
36  (24) UNSIGNED      2  ISTPN6      ALWAYS 6
    1... ....
    .1.. ....          ISTDEF6      NOT USED HERE
38  (26) UNSIGNED      2                                1:=NOT SPECIFIED
                                NOT USED HERE

```

ENTRY FOR PRODUCTION TARGET SUBLIB

```

-----
40  (28) UNSIGNED      2  ISTPN7      ALWAYS 7
    1... ....
    .1.. ....          ISTDEF7      NOT USED HERE
42  (2A) UNSIGNED      2  ISTOF7      1:=NOT SPECIFIED
                                VALUE TABLE OFFSET

```

ENTRY FOR PARAMETER SERVICE

```

-----
44  (2C) UNSIGNED      2  ISTPN8      ALWAYS 8
    1... ....
    .1.. ....          ISTDEF8      NOT USED HERE
46  (2E) UNSIGNED      2                                1:=NOT SPECIFIED
                                NOT USED HERE

```

ENTRY FOR PARAMETER INTO

```

-----
48  (30) UNSIGNED      2  ISTPN9      ALWAYS 9
    1... ....
    .1.. ....          ISTDEF9      NOT USED HERE
50  (32) UNSIGNED      2                                1:=NOT SPECIFIED
                                NOT USED HERE

```

ENTRY FOR SERVICE TARGET SUBLIB(S)

52	(34) UNSIGNED	2	ISTPN10	ALWAYS 10
	1... ..			NOT USED HERE
	.1.. ..		ISTDEF10	1:=NOT SPECIFIED
54	(36) UNSIGNED	2	ISTRE10	REPETITION FACTOR
56	(38) UNSIGNED	2	ISTOF10	VALUE TABLE OFFSET

CROSS-REFERENCE

ISTAN1	10 (A)	ISTOF4	30 (1E)
ISTAN2	14 (E)	ISTOF5	34 (22)
ISTAN21	18 (12)	ISTOF7	42 (2A)
ISTCMD	4 (4)	ISTPN1	8 (8)
ISTDEF10	52 X'40'	ISTPN10	52 (34)
ISTDEF2	12 X'40'	ISTPN2	12 (C)
ISTDEF21	16 X'40'	ISTPN21	16 (10)
ISTDEF22	20 X'40'	ISTPN22	20 (14)
ISTDEF3	24 X'40'	ISTPN3	24 (18)
ISTDEF4	28 X'40'	ISTPN4	28 (1C)
ISTDEF5	32 X'40'	ISTPN5	32 (20)
ISTDEF6	36 X'40'	ISTPN6	36 (24)
ISTDEF7	40 X'40'	ISTPN7	40 (28)
ISTDEF8	44 X'40'	ISTPN8	44 (2C)
ISTDEF9	48 X'40'	ISTPN9	48 (30)
ISTOF10	56 (38)	ISTRE10	54 (36)
ISTOF22	22 (16)	ISTSIT	0 (0)
ISTOF3	26 (1A)	ISTVTB	0 (0)

IKR#SUP

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	14	SUPSIT	
0	(0) A-ADDRESS	4	SUPVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	SUPCMD	COMMAND NUMBER ==> MH35
ENTRY FOR PARAMETER PTF LIST				
8	(8) UNSIGNED	2	SUPPN1	PARAMETER NUMBER ==> PTF LIST
10	(A) SIGNED	2	SUPRE1	REPETITION FACTOR
12	(C) SIGNED	2	SUPOF1	OFFSET TO PTF LIST IN VTAB

CROSS-REFERENCE

SUPCMD	4 (4)	SUPRE1	10 (A)
SUPOF1	12 (C)	SUPSIT	0 (0)
SUPPN1	8 (8)	SUPVTB	0 (0)

IKR#TLR

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>	
0	(0)	STRUCTURE	32	TLRSIT	
0	(0)	A-ADDRESS	4	TLRVTB	POINTER TO VALUE TABLE
4	(4)	CHARACTER	4	TLRCMD	COMMAND NUMBER ==> MH23

ENTRY FOR PARAMETER COMPONENT NAME

8	(8)	UNSIGNED	2	TLRPN1	PARAMETER NUMBER 01
10	(A)	UNSIGNED	2	TLROF1	OFFSET INTO VALUE TABLE

ENTRY FOR PARAMETER PHASE MODULE MACRO

12	(C)	UNSIGNED	2	TLRPN2	PARAMETER NUMBER 02
14	(E)	UNSIGNED	2	TLRAN2	ALTERNATE NUMBER AN = 1 ==> PHASE AN = 2 ==> MODULE AN = 3 ==> MACRO

ENTRY FOR PARAMETER PHASE-NAME MODULE-NAME MACRO-NAME

16	(10)	UNSIGNED	2	TLRPN3	PARAMETER NUMBER 03
18	(12)	UNSIGNED	2	TLROF3	OFFSET INTO VALUE TABLE NOTE: THIS FIELD MUST BE INTERPRETED ACCORDING TO THE FOLLOW CODE TLRAN1 = 1 ==> PHASE-NAME TLRAN1 = 2 ==> MODULE-NAME TLRAN1 = 3 ==> MACRO-NAME

ENTRY FOR PARAMETER SUBLIB

20	(14)	UNSIGNED	2	TLRPN4	PARAMETER NUMBER 04
		1... ..			NOT USED
		.1... ..		TLRDEF4	0 := SUBLIB SPECIFIED
22	(16)	UNSIGNED	2	TLROF4	OFFSET INTO VALUE TABLE NOTE: THIS PARAMETER IS VALID ONLY IF TLRAN1 = 3

ENTRY FOR PARAMETER KEEPDATA

24	(18)	UNSIGNED	2	TLRPN5	PARAMETER NUMBER 05
		1... ..			NOT USED
		.1... ..		TLRDEF5	0 := KEEPDATA SPECIFIED
26	(1A)	UNSIGNED	2		NOT USED

ENTRY FOR PARAMETER DLIBBUILD

```

-----
28 (1C) UNSIGNED      2 TLRPN6      PARAMETER NUMBER 06
   1... ....
   .1.. ....          TLRDEF6      NOT USED
30 (1E) UNSIGNED      2              0 := DLIBBUILD SPECIFIED
                                NOT USED

```

CROSS-REFERENCE

TLRAN2	14 (E)	TLRPN1	8 (8)
TLRCMD	4 (4)	TLRPN2	12 (C)
TLRDEF4	20 X'40'	TLRPN3	16 (10)
TLRDEF5	24 X'40'	TLRPN4	20 (14)
TLRDEF6	28 X'40'	TLRPN5	24 (18)
TLROF1	10 (A)	TLRPN6	28 (1C)
TLROF3	18 (12)	TLRSIT	0 (0)
TLROF4	22 (16)	TLRVTB	0 (0)

IKR#UND

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	12	UNDSIT	
0	(0) A-ADDRESS	4	UNDTVBTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	UNDCMD	COMMAND NUMBER ==> MH15

ENTRY FOR PARAMETER COMPONENT : APAR NUMBER

8	(8) UNSIGNED	2	UNDPN1	PARAMETER NUMBER
10	(A) UNSIGNED	2	UNDOF1	OFFSET INTO VALUE TABLE

<u>CROSS-REFERENCE</u>				
UNDCMD	4	(4)	UNDSIT	0 (0)
UNDOF1	10	(A)	UNDTVBTB	0 (0)
UNDPN1	8	(8)		

IKR#VER

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	12	VERSIT	
0	(0) A-ADDRESS	4	VERVTB	POINTER TO VALUE TABLE
4	(4) CHARACTER	4	VERCMD	COMMAND NUMBER ==> MH47
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> ENTRY FOR PARAMETER LINE ID AFTER WHICH IS TO BE INSERTED </div>				
8	(8) UNSIGNED	2	VERPN1	PARAMETER NUMBER
10	(A) SIGNED	2	VEROF1	OFFSET TO LINE ID IN VTAB

<u>CROSS-REFERENCE</u>				
VERCMD	4	(4)	VERSIT	0 (0)
VEROF1	10	(A)	VERVTB	0 (0)
VERPN1	8	(8)		

Internal Representation

The Internal Representation (IRP) data area serves as input to the processor routines. It is a collection of the information from the standardized internal table (SIT) and value table (VTAB) structures for the function statement and all its detail statements. As such, it contains all the information needed by the respective processor to perform the function requested by the user.

The IRP consists of indicators (bit switches) describing the operation to be performed, and the variable information needed to process the command, such as module name, PTF/APAR number, etc. Refer to the following pages for descriptions of the individual IRPs.

For the format of the SIT and the VTAB, see Figure 13 on page 274 and **SIF tables** earlier in this chapter and *VSE/Advanced Functions Diagnosis Reference: Service Aids*.

IKRXCOR

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>	
0	(0)	STRUCTURE	20	CORRECT	
0	(0)	CHARACTER	12	CORCMP	COMPONENT ID WITH OPTIONAL FEATURE CODE
0	(0)	CHARACTER	9	CORCMPID	COMPONENT IN THE 'RETAIN' FORMAT, FOR INSTANCE 5745SCJCL (WITH DASHES REMOVED)
0	(0)	CHARACTER	6	CORCMPGR	GROUP ID THAT IS COMMON FOR RELATED COMPONENTS, FOR INSTANCE 5745SC
6	(6)	CHARACTER	3	CORCMPSF	SUFFIX FOR SPECIFIC COMPONENT, LIKE FOR INSTANCE JCL
9	(9)	CHARACTER	3	CORCMPFT	FEATURE CODE. BIN ZEROS, IF NOT SPECIFIED
12	(C)	CHARACTER	8	CORAPAR	APAR NUMBER OR LOCAL FIX NUMBER
20	(14)	BITSTRING	0		1 = OPTION EFFECTIVE, 0 = NOT EFFECTIVE
	1... ..			COREVOK	THE CORRECTIONS ARE TO BE RECORDED IN THE HISTORY FILE, SO THAT THEY CAN BE REVOKED (PROVIDE A BACKOUT CAPABILITY)

CROSS-REFERENCE

CORAPAR	12	(C)	CORCMPID	0	(0)
CORCMP	0	(0)	CORCMPSF	6	(6)
CORCMPFT	9	(9)	COREVOK	20	X'80'
CORCMPGR	0	(0)	CORRECT	0	(0)

IKRXCPY

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>	
0	(0)	STRUCTURE	44	COPY	
0	(0)	SIGNED	2	COPSRC	SOURCE FOR COPY SYSTEM (ALIAS PERM): 1, AUXILIARY (ALIAS TEMP): 2, TAPE: 3
2	(2)	SIGNED	2	COPTGT	TARGET FOR COPY SAME TYPES AS ABOVE

TARGET AND SOURCE TO BE DIFFERENT WILL BE ENSURED
PARAMETERS FOR PRODUCT BACKUP

4	(4)	UNSIGNED	1	COPLTYP	REQUESTED TYPE: 1=PRODUCTION ONLY, 2=SERVICE ONLY, 3=BOTH SUBLIBS
5	(5)	CHARACTER	36	COPCMDP	BACKUP COMMAND PARAM.
5	(5)	CHARACTER	18	COPTPFN	TAPE FILE ID
5	(5)	CHARACTER	1		QUOTE
6	(6)	CHARACTER	16	COPIDN	FILE ID
22	(16)	CHARACTER	1		RESERVED FOR QUOTE
23	(17)	CHARACTER	18	COPHEADI	HEADER INFO
	1... ..			COPHEAD\$	MEMBER SPECIFIED
24	(18)	CHARACTER	17	COPHEADN	MEMBER NAME
42	(2A)	CHARACTER	2	COPFLST	REQUESTED PRODUCTS
42	(2A)	SIGNED	2	COPFT#	# OF ELEMENTS IN LIST
44	(2C)	CHARACTER	0	COPFEATS	PRODUCT-IDS

CROSS-REFERENCE

COPCMDP	5 (5)	COPIDN	6 (6)
COPFEATS	44 (2C)	COPLTYP	4 (4)
COPFLST	42 (2A)	COPSRC	0 (0)
COPFT#	42 (2A)	COPTGT	2 (2)
COPHEAD\$	23 X'80'	COPTPFN	5 (5)
COPHEADI	23 (17)	COPY	0 (0)
COPHEADN	24 (18)		

IKRXCRE

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	2	CREATE	
0	(0) SIGNED	2	CREHFILE	TYPE OF HISTORY FILE TO BE CREATED SYSTEM (ALIAS PERM): 1, AUXILIARY (ALIAS TEMP): 2

	<u>CROSS-REFERENCE</u>			
CREATE	0 (0)	CREHFILE	0	(0)

IKRXCUM

OFFSETS	TYPE	LENGTH	NAME	DESCRIPTION
0	(0) STRUCTURE	104	CUMPTF	
0	(0) UNSIGNED	1	CUMTYPE	TYPE OF OPERATION INSTALL BACKOUT: 3, INSTALL SERVICE: 4, LIST SERVICETAPE: 5, INSTA RESTART=LINK: 6, SELECT GENFILE: 7,
1	(1) BITSTRING	1	CUMFLG	CONTROL FLAG
	1... ..		CUMREVOK	1:=REVOBABLE SPECIFIED
	.1.. ..		CUMCOVER	1:=LIST COVERLETTER
	..1.		CUMXREF	1:=LIST CROSS-REF.
	...1		CUMDOCU	1:=LIST DOCUMENT
 1...		CUMSEPAR	1:=SEPERATE PTFS
1..		CUMDATA	1:=PICK UP DATA
1.		CUMIPF	1:=SERVICE DIALOG
1		CUMDISK	1:=SERVICEFILE DISK

THE FOLLOWING PART IS MEANINGFUL ONLY FOR: 'SELECT GENFILE'

2	(2) CHARACTER	26	CUMSELG	
2	(2) CHARACTER	12	CUMCMP	COMPONENT WITH OPTIONAL FEATURE CODE
2	(2) CHARACTER	9	CUMCMPID	COMPONENT ID
11	(B) CHARACTER	3	CUMCMPLV	LEVEL
14	(E) CHARACTER	11	CUMGFILE	GENERATION FILE INFO
14	(E) BITSTRING	1		FLAG BYTE
	1... ..		CUMGF\$	1:=CUMGFID PRESENT
15	(F) CHARACTER	10	CUMGFID	GENERATION FILE ID
15	(F) CHARACTER	1	CUMGFLB	LIBRARY TYPE PROCEDURE:'P'
16	(10) CHARACTER	1	CUMGFCL	LIBRARY CLASS MACRO:'.', PROCEDURE:'-'
17	(11) CHARACTER	8	CUMGFNM	PROCEDURE/MACRO NAME
26	(1A) SIGNED	2	CUMGENS	SET-ID OF MEMBERS TO BE RE-GENERATED

THE FOLLOWING PART IS MEANINGFUL ONLY FOR: 'INSTALL
SERVICE/BACKOUT'

28	(1C) CHARACTER	69	CUMISV	
28	(1C) BITSTRING	1	ISVCNTL	CONTROL FLAG
	1... ..		ISVEXCI	1:=INTERNAL EXCLUDE SPECIFIED
	.1.. ..		ISVEXCA	1:=EXCLUDE ALL SPECIFIED
29	(1D) CHARACTER	12	ISVINCS	SET-IDS OF INCLUDES
29	(1D) CHARACTER	4	ISVINCPR	INCLUDED PRODUCTS
33	(21) CHARACTER	4	ISVINCCO	INCLUDED COMPONENTS
37	(25) CHARACTER	4	ISVINCPT	INCLUDED PTFS
41	(29) CHARACTER	12	ISVEXCS	SET-IDS OF EXCLUDES
41	(29) CHARACTER	4	ISVEXCPR	EXCLUDED PRODUCTS
45	(2D) CHARACTER	4	ISVEXCCO	EXCLUDED COMPONENTS
49	(31) CHARACTER	4	ISVEXCPT	EXCLUDED PTFS

53	(35)	CHARACTER	4	ISVEXCIN	INTERNALLY EXCLUDED PTF'S
58	(3A)	SIGNED	2	ISVINCL	SET-ID OF USER-INCLUDED PTF'S

60	(3C)	SIGNED	2	ISVMEMB	SET-ID OF MEMBERS TO CATALOG
62	(3E)	SIGNED	2	ISVARCH	SET-ID OF ARCHIVED PTF'S

64	(40)	SIGNED	2	ISVNARCH	SET-ID OF PTF'S WHICH COULD NOT BE ARCHIVED
66	(42)	SIGNED	2	ISVPTFMB	SET-ID OF PTF'S AND ASSOCIATED MEMBERS

68	(44)	CHARACTER	12	ISVCURRP	CURRENT PTF, ITS MEMBERS AND SUPERSEDED PTF'S

80	(50)	SIGNED	2	ISVCURRL	SET-ID OF CURRENT SUBLIBS
82	(52)	CHARACTER	15	ISVSYRS	IPL-ED SYSTEM SUBLIB NAME

THE FOLLOWING PART IS MEANINGFUL ONLY FOR: 'LIST
SERVICETAPE'

98	(62)	CHARACTER	2	CUMLSTS	
98	(62)	SIGNED	2	CUMPTFL	SET-ID OF PTF'S TO BE LISTED

100	(64)	CHARACTER	4	CUMTAPES	

100	(64)	BITSTRING	1	TAPEFL	FLAG BYTE FOR TAPES
		1... ..		CUMTAP	TAPES PARAMETER SPECIFIED
		.1.. ..		TAPEHIST	HISTORY AT BEGIN OF TAPE
		..1.		TESTHIST	ALLOW OPEN OF HISTORY
		...1 1111			RESERVED
102	(66)	UNSIGNED	2	TAPENU	NUMBER OF SERVICE TAPES

CROSS-REFERENCE

CUMCMP	2 (2)	CUMXREF	1 X'20'
CUMCMPID	2 (2)	ISVARCH	62 (3E)
CUMCMPLV	11 (B)	ISVCNTL	28 (1C)
CUMCOVER	1 X'40'	ISVCURRL	80 (50)
CUMDATA	1 X'04'	ISVCURRP	68 (44)
CUMDISK	1 X'01'	ISVEXCA	28 X'40'
CUMDOCU	1 X'10'	ISVEXCCO	45 (2D)
CUMFLG	1 (1)	ISVEXCI	28 X'80'
CUMGENS	26 (1A)	ISVEXCIN	53 (35)
CUMGF\$	14 X'80'	ISVEXCPR	41 (29)
CUMGFCL	16 (10)	ISVEXCPT	49 (31)
CUMGFID	15 (F)	ISVEXCS	41 (29)
CUMGFIE	14 (E)	ISVINCCO	33 (21)
CUMGFLB	15 (F)	ISVINCL	58 (3A)
CUMGFNM	17 (11)	ISVINCPR	29 (1D)
CUMIPF	1 X'02'	ISVINCPPT	37 (25)
CUMISV	28 (1C)	ISVINCS	29 (1D)
CUMLSTS	98 (62)	ISVMEMB	60 (3C)
CUMPTF	0 (0)	ISVNARCH	64 (40)
CUMPTFL	98 (62)	ISVPTFMB	66 (42)
CUMREVOK	1 X'80'	ISVSYSRS	82 (52)
CUMSELG	2 (2)	TAPEFL	100 (64)
CUMSEPAR	1 X'08'	TAPEHIST	100 X'40'
CUMTAP	100 X'80'	TAPENU	102 (66)
CUMTAPES	100 (64)	TESTHIST	100 X'20'
CUMTYPE	0 (0)		

IKRXDMP

	<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
	0	(0) STRUCTURE	2	DUMP	
	0	(0) SIGNED	2	DUMTYPE	TYPE OF HISTORY FILE TO BE DUMPED SYSTEM (ALIAS PERM): 1, AUXILIARY (ALIAS TEMP): 2
	<u>CROSS-REFERENCE</u>				
DUMP	0 (0)			DUMTYPE	0 (0)

IKRXENT

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	131	ENTHIVE	
0	(0) UNSIGNED	1	ENTFUNC	FUNCTION CONTROL STATEMENT ARCHIVE: 1, APPLY: 2, REVOKE:3 INCORPORATE:4, CORRECT:5 U UNDO:6 TAILOR:7
1	(1) CHARACTER	12	ENTCMP	COMPONENT ID WITH OPTIONAL FEATURE CODE
1	(1) CHARACTER	9	ENTCMPID	COMPONENT IN THE 'RETAIN' FORMAT, FOR INSTANCE 5745SCJCL (WITH DASHES REMOVED
1	(1) CHARACTER	6	ENTCMPGR	GROUP ID THAT IS COMMON FOR RELATED COMPONENTS, FOR INSTANCE 5745SC
7	(7) CHARACTER	3	ENTCMPSF	SUFFIX FOR SPECIFIC COMPONENT, LIKE FOR INSTANCE JCL
10	(A) CHARACTER	3	ENTCMLV	LEVEL. BIN ZEROS, IF NOT SPECIFIED
13	(D) CHARACTER	6	ENTPRDPL	PRODUCT ID
13	(D) CHARACTER	3	ENTPRPPP	PRODUCT CODE
16	(10) CHARACTER	3	ENTPRRFF	PRODUCT LEVEL
19	(13) CHARACTER	8	ENTPTAP	PTF OR APAR OR LOCAL FIX NUMBER
27	(1B) UNSIGNED	1	ENTTYPE	TYPE OF ENTRY TO BE MADE IN HISTORY FILE PTF: 2, FEATURE: 4, COMP: 1, APAR: 3, GENER GENERATED MEMBER: 5,
28	(1C) BITSTRING	1		1 = OPTION EFFECTIVE, 0 = NOT EFFECTIVE
	1... ..		ENTDLIBB	THIS PTF IS APPLIED (IBM INTERNALLY) TO BUILD A DLIB TAILOR DLIBBUILD
	.1..		ENTBKREQ	A BACKOUT PTF IS REQUESTED
	..1.		ENTNOLNK	NO LINK REQUESTED(DLIBBUILD)
	...1		ENTSOFT	SOFT PTF REJECTION
 1...		ENTKEEPD	TAILOR KEEPDATA
1..		ENTXREF	TAILOR XREF
1.		ENTPROD	PRODUCTION SUBLIB ONLY
32	(20) A-ADDRESS	4	ENTRELP	POINTER TO LIST OF APPLICABLE RELEASES (PTF), OR THE RELEASE OF APAR (NULL, IF OMITT OMITTED). OR THE RELEASE OF A COMPONENT (ENSURED TO BE PRESENT)
36	(24) A-ADDRESS	4	ENTCMNP	POINTER TO COMMENT ON THIS PTF/APAR/GENERATED MEMBER (TO BE INSERTED IN THE HISTORY FILE
40	(28) A-ADDRESS	4	ENTAPARP	POINTER TO LIST OF APAR-NUMBERS

44	(2C) A-ADDRESS	4	ENTREQP	POINTER TO REQUIREMENTS MATRIX
48	(30) A-ADDRESS	4	ENTSUPP	POINTER TO LIST OF SUPERSEDED COMPONENTS/PTFS
52	(34) A-ADDRESS	4	ENTAFFP	POINTER TO LIST OF AFFECTED PHASES, MODULES, AND MACROS
56	(38) A-ADDRESS	4	ENTCISP	POINTER TO COMPRISES DATA FOR ARCHIVE FEATURE
60	(3C) CHARACTER	52	ENTGEN	GENERATED MEMBER FOR USE BY TAILOR
60	(3C) CHARACTER	10	ENTGENID	GEN.MEMBER IDENTIFIER
60	(3C) CHARACTER	1	ENTGENLB	CONCERNED LIBRARY TYPE IS : C- FOR PHASES, R- FOR MODULES, FOR MACROS SUBLIB QUALIFI (DEFAULT E)
61	(3D) CHARACTER	1	ENTGENCL	CONCERNED CLASS IS : MACRO:'-' OTHER:'-'
62	(3E) CHARACTER	8	ENTGENNM	NAME OF GEN.MEMBER PADDED WITH TRAILING BLANKS
70	(46) BITSTRING 1... .. .1.. ..	1	ENTGFLAG ENTGCHK ENTGNRC	FLAG BYTE CONTROL STATEMENTS FOR SCP'S ARE CHECKED GENERIC NAME SPECIFIED
72	(48) A-ADDRESS	4	ENTGOMTP	PTR. TO OVERALL INCLUDED MACRO TABLE (SEE ENTIML)
76	(4C) A-ADDRESS	4	ENTGSMTP	PTR. TO SPECIFIC INCL. MACRO TABLE (SEE ENTIML)
80	(50) A-ADDRESS	4	ENTGSMRP	PTR TO SPECIFIC INCLUDED MACRO REFERENCE(SEE ENTRLM)
84	(54) A-ADDRESS	4	ENTGPRDP	PTR. TO DATA FOR TAILORING THE MEMBER (SEE ENTFIX)
88	(58) CHARACTER 1... ..	18	ENTGSCP ENTGSCP\$	PROGRAMS TO BE CALLED SCP NAME PRESENT
89	(59) CHARACTER	8	ENTGSCP#	NAME OF SYSTEM CONTROL PROGRAM
108	(6C) SIGNED	4	ENTGCRD#	# OF OUTPUT CARDS PRODUCED BY THE FIRST CALLED SCP
112	(70) A-ADDRESS	4	ENTINFP	PTR. TO LIST OF INFLUENCED LIBRARY MEMBERS
116	(74) CHARACTER	15	ENTLIB	TARGET LIB/SUBLIB INFO
116	(74) CHARACTER	7	ENTLBNM	LIBRARY NAME
123	(7B) CHARACTER	8	ENTSLNM	SUBLIBRARY NAME

CROSS-REFERENCE

ENTAFFP	52 (34)	ENTGSCP	88 (58)
ENTAPARP	40 (28)	ENTGSCP\$	88 X'80'
ENTBKREQ	28 X'40'	ENTGSCP#	89 (59)
ENTCISP	56 (38)	ENTGSMRP	80 (50)
ENTCMNP	36 (24)	ENTGSMT#	76 (4C)
ENTCMP	1 (1)	ENTHIVE	0 (0)
ENTCMPGR	1 (1)	ENTINFP	112 (70)
ENTCMPID	1 (1)	ENTKEEPD	28 X'08'
ENTCMPLV	10 (A)	ENTLBNM	116 (74)
ENTCMPSF	7 (7)	ENTLIB	116 (74)
ENTDLIBB	28 X'80'	ENTNOLNK	28 X'20'
ENTFUNC	0 (0)	ENTPRDPL	13 (D)
ENTGCHK	70 X'80'	ENTPROD	28 X'02'
ENTGCRD#	108 (6C)	ENTPRPPP	13 (D)
ENTGEN	60 (3C)	ENTPRRFF	16 (10)
ENTGENCL	61 (3D)	ENTPTAP	19 (13)
ENTGENID	60 (3C)	ENTRELP	32 (20)
ENTGENLB	60 (3C)	ENTREQP	44 (2C)
ENTGENNM	62 (3E)	ENTSLNM	123 (7B)
ENTGFLAG	70 (46)	ENTSOFT	28 X'10'
ENTGNRC	70 X'40'	ENTSUPP	48 (30)
ENTGOMTP	72 (48)	ENTTYPE	27 (1B)
ENTGPRDP	84 (54)	ENTXREF	28 X'04'

ENTRELLS

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	2	ENTRELLS	
0	(0) SIGNED	2	ENTREL#	NUMBER OF ENTRIES IN APPLICABLE RELEASES LIST
2	(2) CHARACTER	0	ENTRELS	COMPLETE RELEASE IDENTIFIER
2	(2) CHARACTER	2	ENTRELVS	VERSION
4	(4) CHARACTER	2	ENTRELEA	PROPER RELEASE NUMBER
6	(6) CHARACTER	2	ENTRELMN	MAINTENANCE LEVEL

	<u>CROSS-REFERENCE</u>			
ENTREL#	0	(0)	ENTRELMN	6 (6)
ENTRELEA	4	(4)	ENTRELS	2 (2)
ENTRELLS	0	(0)	ENTRELVS	2 (2)

ENTCMN

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	5	ENTCMN	
0	(0) A-ADDRESS	4	ENTCMNPN	POINTER TO NEXT COMMENT
4	(4) UNSIGNED	1	ENTCMN#	LENGTH OF COMMENT IN BYTES
5	(5) CHARACTER	0	ENTCMNT	TEXT OF COMMENT THE TEXT IS STRIPPED OF THE DELIMITING QUOTES

CROSS-REFERENCE

ENTCMN	0 (0)	ENTCMNPN	0 (0)
ENTCMN#	4 (4)	ENTCMNT	5 (5)

ENTAPA

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	2	ENTAPA	
0	(0) SIGNED	2	ENTAPA#	NUMBER OF APARS IN APAR LIST (FOR THE EVENTUALITY OF A LONGER APAR NUMBER IN THE FUTURE)
2	(2) CHARACTER	0	ENTAPAR	
2	(2) CHARACTER	7		COMPLETE APAR NUMBER OF TODAY ALPHABETIC PREFIX
2	(2) CHARACTER	2	ENTAPAPF	
4	(4) CHARACTER	5	ENTAPANO	NUMERICAL PART OF APAR 'NUMBER'

CROSS-REFERENCE

ENTAPA	0 (0)	ENTAPAPF	2 (2)
ENTAPA#	0 (0)	ENTAPAR	2 (2)
ENTAPANO	4 (4)		

ENTREQ

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	2	ENTREQ	
0	(0) SIGNED	2	ENTREQ#	NUMBER OF ENTRIES IN REQUIREMENTS MATRIX
2	(2) CHARACTER	0	ENTREQS	
2	(2) SIGNED	2	ENTREQOR	INDEX OF FIRST OR ONLY ALTERNATE REQUIREMENT, OR NONE: 'FFFF'X
4	(4) SIGNED	2	ENTREQND	INDEX OF NEXT 'AND' REQUIREMENT, IF THERE ISN'T ANY NONE: 'FFFF'X
6	(6) UNSIGNED	1	ENTREQTY	TYPE OF REQUIREMENT PRE: 1, CO: 2, NOT: 3
7	(7) UNSIGNED	1	ENTREQCL	CLASS OF REQUIREMENT COMP:1, FEATURE:4, PTF OR APAR :2
8	(8) CHARACTER	9	ENTREQCP	COMPONENT OR
8	(8) CHARACTER	12	ENTREQPR	
8	(8) CHARACTER	6	ENTREQFT	PRODUCT ID
8	(8) CHARACTER	3	ENTRQPPP	PRODUCT CODE
11	(B) CHARACTER	3	ENTRQRFF	PRODUCT LEVEL
17	(11) CHARACTER	8	ENTREQPT	PTF NUMBER OR APAR

CROSS-REFERENCE

ENTREQ	0 (0)	ENTREQPR	8 (8)
ENTREQ#	0 (0)	ENTREQPT	17 (11)
ENTREQCL	7 (7)	ENTREQS	2 (2)
ENTREQCP	8 (8)	ENTREQTY	6 (6)
ENTREQFT	8 (8)	ENTRQPPP	8 (8)
ENTREQND	4 (4)	ENTRQRFF	11 (B)
ENTREQOR	2 (2)		

ENTSUP

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	2	ENTSUP	
0	(0) SIGNED	2	ENTSUP#	NUMBER OF SUPERSEDED PTFS (FOR THE EVENTUALITY OF A LONGER PTF NUMBER IN THE FUTURE
2	(2) CHARACTER	0	ENTSUPPT	
2	(2) CHARACTER	7		PTF NUMBER OF TODAY
2	(2) CHARACTER	2	ENTSUPPF	ALPHABETIC PREFIX
4	(4) CHARACTER	5	ENTSUPNO	NUMERICAL PART OF PTF NUMBER

CROSS-REFERENCE

ENTSUP	0 (0)	ENTSUPPF	2 (2)
ENTSUP#	0 (0)	ENTSUPPT	2 (2)
ENTSUPNO	4 (4)		

ENTAFF

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	6	ENTAFF	
0	(0) A-ADDRESS	4	ENTAFFNP	POINTER TO CONTINUATION OF 'AFFECTED LIST'
4	(4) SIGNED	2	ENTAFF#	NUMBER OF ENTRIES IN 'AFFECTED-LIST'
6	(6) CHARACTER	0	ENTAFFS	
6	(6) CHARACTER	10	ENTAFFID	IDENTIFIER AS IT WILL BE ENTERED IN HISTORY FILE
6	(6) CHARACTER	1	ENTAFFLB	LIBRARY AFFECTED PHASE: 'C', MODULE: 'R'. IN CASE OF SOURCE STATEMENT LIBRARY AFFECTE THIS FIELD CONTAINS THE SUBLIBRARY QUALIFIER (A DEFAULT OF 'E' IS PROVIDED)
7	(7) CHARACTER	1	ENTAFFCL	CLASS AFFECTED MACRO: '.', LIOCS: '+', OTHER: '-'
8	(8) CHARACTER	8	ENTAFFNM	PHASE, MODULE, OR MACRO NAME. LEFT ALIGNED, PADDED WITH TRAILING BLANKS
16	(10) A-ADDRESS	4	ENTAFFDP	POINTER TO FIX DESCRIPTION LIST

CROSS-REFERENCE

ENTAFF	0 (0)	ENTAFFLB	6 (6)
ENTAFF#	4 (4)	ENTAFFNM	8 (8)
ENTAFFCL	7 (7)	ENTAFFNP	0 (0)
ENTAFFDP	16 (10)	ENTAFFS	6 (6)
ENTAFFID	6 (6)		

ENTFIX

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	7	ENTFIX	
0	(0) A-ADDRESS	4	ENTFIXNP	POINTER TO CONTINUATION OF 'FIX DESCRIPTION' LIST
4	(4) UNSIGNED	1	ENTFIX#	LENGTH OF FIX DESCRIPTION
5	(5) CHARACTER	2	ENTFIXS	FIX DESCRIPTION
5	(5) UNSIGNED	1	ENTFIXST	TYPE OF DESCRIPTION
6	(6) UNSIGNED	1	ENTFIXSL	LENGTH OF DESCRIPTION
7	(7) CHARACTER	0	ENTFIXSD	FIX DESCRIPTION DATA

CROSS-REFERENCE

ENTFIX	0 (0)	ENTFIXSD	7 (7)
ENTFIX#	4 (4)	ENTFIXSL	6 (6)
ENTFIXNP	0 (0)	ENTFIXST	5 (5)
ENTFIXS	5 (5)		

ENTIML

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	6	ENTIML	
0	(0) A-ADDRESS	4	ENTIMLNP	POINTER TO CONTINUATION OF 'INCLUDED MACRO LIST'
4	(4) SIGNED	2	ENTIML#	NUMBER OF ENTRIES IN 'INCLUDED MACRO LIST'
6	(6) CHARACTER	0	ENTIMLEN	INCLUDED MACRO ENTRY WITH IDENTIFIER AS IT WILL BE USED IN HISTORY FILE
6	(6) UNSIGNED	1	ENTIMLCO	CONTROL BYTE FOR THIS MACRO ENTRY
7	(7) CHARACTER	10	ENTIMLID	MEMBER ID
7	(7) CHARACTER	1	ENTIMLLB	LIBRARY AFFECTED PHASE: 'C', MODULE: 'R'. IN CASE OF SOURCE STATEMENT LIBRARY AFFECTE THIS FIELD CONTAINS THE SUBLIBRARY QUALIFIER (DEFAULT OF 'E' IS PROVIDED)
8	(8) CHARACTER	1	ENTIMLCL	CLASS AFFECTED MACRO: '.', LIOCS: '+', OTHER: '-'
9	(9) CHARACTER	8	ENTIMLNM	PHASE, MODULE, OR MACRO NAME. LEFT ALIGNED, PADDED WITH TRAILING BLANKS

CROSS-REFERENCE

ENTIML	0 (0)	ENTIMLID	7 (7)
ENTIML#	4 (4)	ENTIMLLB	7 (7)
ENTIMLCL	8 (8)	ENTIMLNM	9 (9)
ENTIMLCO	6 (6)	ENTIMLNP	0 (0)
ENTIMLEN	6 (6)		

ENTRLM

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	6	ENTRLM	
0	(0) A-ADDRESS	4	ENTRLMNP	POINTER TO CONTINUATION OF 'REFERENCE LIST OF MACROS'
4	(4) SIGNED	2	ENTRLM#	NUMBER OF ENTRIES IN 'REFERENCE LIST OF MACROS'
6	(6) SIGNED	0	ENTRLMRN	REFERENCE TO MACRO TABLE IS RELATIVE NUMBER WITHIN IT

CROSS-REFERENCE

ENTRLM	0 (0)	ENTRLMNP	0 (0)
ENTRLM#	4 (4)	ENTRLMRN	6 (6)

ENTINF

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	6	ENTINF	
0	(0) A-ADDRESS	4	ENTINFNP	PTR TO NEXT LIST
4	(4) SIGNED	2	ENTINF#	NUMBER OF ENTRIES
6	(6) CHARACTER	0	ENTINFID	INFLUENCED MEMBER ID
6	(6) CHARACTER	1	ENTINFLB	LIBRARY AFFECTED
7	(7) CHARACTER	1	ENTINFCL	CLASS AFFECTED
8	(8) CHARACTER	8	ENTINFNM	MEMBER NAME

CROSS-REFERENCE

ENTINF	0 (0)	ENTINFLB	6 (6)
ENTINF#	4 (4)	ENTINFNM	8 (8)
ENTINFCL	7 (7)	ENTINFNP	0 (0)
ENTINFID	6 (6)		

IKRXLKP

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	38	LOOKUP	
0	(0) UNSIGNED	1	LKPTYPE	WHAT TO LOOK UP LATEST SREVICE: 1, PTFS: 2, APARS: 3, FEATURES: 4, MODULES: 5
1	(1) BITSTRING 1... ..	1	LKPCMPFL	FLAG: 0 = COMPONENT NOT SPECIFIED 1 = COMPONENT WAS SPECIFIED
	.1..		LKPDATA	1:=DATA DISPLAY REQUESTED
	..11 1111		LKPFLAG	RESERVED
2	(2) CHARACTER	12	LKPCOMP	COMPONENT
2	(2) CHARACTER	9	LKPCMPID	COMPONENT-ID
11	(B) CHARACTER	3	LKPCMPLV	COMPONENT-LEVEL
14	(E) CHARACTER	6	LKPFEAT	FEATURE-ID
14	(E) CHARACTER	3	LKPPPP	PRODUCT CODE
17	(11) CHARACTER	3	LKPRFF	PRODUCT LEVEL
20	(14) CHARACTER	8	LKPPTAP	NUMBER OF PTF, OR APAR, TO BE LOOKED UP
28	(1C) CHARACTER	10	LKPMODID	IDENTIFIER AS IT WILL BE ENTERED IN HISTORY FILE
28	(1C) CHARACTER	1	LKPMODLB	LIBRARY MEMBER PHASE: 'C', MODULE: 'R'. IN CASE OF SOURCE STATEMENT LIBRARY MEMBER, T FIELD CONTAINS THE SUBLIBRARY QUALIFIER (A DEFAULT OF 'E' IS PROVIDED)
29	(1D) CHARACTER	1	LKPMODCL	LIBRARY MEMBER MACRO: '.', OTHER: '-'
30	(1E) CHARACTER	8	LKPMODNM	PHASE, MODULE, OR MACRO NAME. LEFT ALIGNED, PADDED WITH TRAILING BLANKS

CROSS-REFERENCE

LKPCMPFL	1 X'80'	LKPMODID	28 (1C)
LKPCMPID	2 (2)	LKPMODLB	28 (1C)
LKPCMPLV	11 (B)	LKPMODNM	30 (1E)
LKPCOMP	2 (2)	LKPPPP	14 (E)
LKPDATA	1 X'40'	LKPPTAP	20 (14)
LKPFEAT	14 (E)	LKPRFF	17 (11)
LKPFLAG	1 X'3F'	LKPTYPE	0 (0)
LKPMODCL	29 (1D)	LOOKUP	0 (0)

IKRXMRG

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	20	MERGE	
0	(0) SIGNED	2	MERSRC	SOURCE FOR MERGE SYSTEM (ALIAS PERM): 1, AUXILIARY (ALIAS TEMP): 2, TAPE: 3
2	(2) SIGNED	2	MERTGT	TARGET FOR MERGE SAME TYPES AS ABOVE

TARGET AND SOURCE TO BE DIFFERENT WILL BE ENSURED

4	(4) UNSIGNED	1	MERACT	ACTION TO BE PERFORMED: 1:=TEST FOR SPACE 2:=PERFORM MERGE
6	(6) SIGNED	2	MERFREE	NUMBER OF RECORDS WHICH MUST BE FREE ADDITIONALLY (APPLIES TO ACTION TEST ONLY)
8	(8) BITSTRING	1	MERFLAG	MERGE CONTROL FLAG
	1... ..		MERKTIME	1:=KEEP DATE AND TIME
	.1.. ..		MERFLPTF	1:=FLAG MERGED PTF AS INCORRECT
	..1.		MERSEL	1:=SELECTIVE MERGE
	...1		MEROCOMP	1:=COMPONENTS ONLY
 1...		MERNOPRT	1:=NO MESSAGE PRINTING
12	(C) A-ADDRESS	4	MERCOMPP	PTR TO SELECTIVE LIST
16	(10) A-ADDRESS	4	MERPRODP	PTR TO SELECTIVE LIST

CROSS-REFERENCE

MERACT	4 (4)	MERNOPRT	8 X'08'
MERCOMPP	12 (C)	MEROCOMP	8 X'10'
MERFLAG	8 (8)	MERPRODP	16 (10)
MERFLPTF	8 X'40'	MERSEL	8 X'20'
MERFREE	6 (6)	MERSRC	0 (0)
MERGE	0 (0)	MERTGT	2 (2)
MERKTIME	8 X'80'		

IKRXPER

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	169	PERSONAL	
0	(0) BITSTRING	1	PER\$	1=PRESENT, 0=OMITTED
	1... ..		PER\$CUST	PRESENCE OF CUSTOMER NAME
	.1.. ..		PER\$ADDR	PRESENCE OF ADDRESS
	..1.		PER\$PHON	PRESENCE OF PHONE NUMBER
	...1		PER\$PROG	PRESENCE OF PROGRAMMER NAME
 1...		PER\$ENV	PRESENCE OF ENVIRONMENT SPECIFICATION
1	(1) CHARACTER	168		FIELDS HOLDING THE SPECIFIED VALUES LEFT ALIGNED WITH TRAILING BLANKS
1	(1) CHARACTER	20	PERCUST	CUSTOMER NAME
21	(15) CHARACTER	45	PERADDR	ADDRESS
66	(42) CHARACTER	17	PERPHONE	PHONE NUMBER
83	(53) CHARACTER	24	PERPROGR	SYSTEM PROGRAMMER'S NAME
107	(6B) CHARACTER	62	PERENV	ENVIRONMENT DESCRIPTION

CROSS-REFERENCE

PER\$	0 (0)	PERADDR	21 (15)
PER\$ADDR	0 X'40'	PERCUST	1 (1)
PER\$CUST	0 X'80'	PERENV	107 (6B)
PER\$ENV	0 X'08'	PERPHONE	66 (42)
PER\$PHON	0 X'20'	PERPROGR	83 (53)
PER\$PROG	0 X'10'	PERSONAL	0 (0)

IKRXRDC

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	6	IKRXRDC	IRP FOR FUNCTION RESIDENCE
0	(0) SIGNED	2	RDCPRSET	SET CONTAINING THE SPECIFIED PRODUCT-IDS
2	(2) SIGNED	2	RDCTGTP	SET CONTAINING TARGET INFO FOR PRODUCTION PART (0 IF EMPTY)
4	(4) SIGNED	2	RDCTGTS	SET CONTAINING TARGET INFO FOR SERVICE PART (0 OF EMPTY)

<u>CROSS-REFERENCE</u>				
IKRXRDC	0	(0)	RDCTGTP	2 (2)
RDCPRSET	0	(0)	RDCTGTS	4 (4)

IKRXRMV

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>	
0	(0)	STRUCTURE	37	REMOVE	
0	(0)	UNSIGNED	1	REMTYPE	TYPE OF HISTORY ENTRIES TO BE REMOVED PTF: 2, FEATURE: 4, COMP: 1, APAR: 3, OBSOLETE 5, GENERATED MEMBER: 6
1	(1)	CHARACTER	12	REMCMP	COMPONENT ID WITH OPTIONAL FEATURE CODE
1	(1)	CHARACTER	9	REMCMPID	COMPONENT IN THE 'RETAIN' FORMAT, FOR INSTANCE 5745SCJCL (WITH DASHES REMOVED
1	(1)	CHARACTER	6	REMCMPGR	GROUP ID THAT IS COMMON FOR RELATED COMPONENTS, FOR INSTANCE 5745SC
7	(7)	CHARACTER	3	REMCMPSF	SUFFIX FOR SPECIFIC COMPONENT, LIKE FOR INSTANCE JCL
10	(A)	CHARACTER	3	REMCPLV	LEVEL ID
13	(D)	CHARACTER	6	REMPROD	PRODUCT WITH LEVEL-ID
13	(D)	CHARACTER	3	REMPRPP	PRODUCT ID
16	(10)	CHARACTER	3	REMPRFF	LEVEL ID
19	(13)	CHARACTER	8	REMP TAP	PTF OR APAR NUMBER TO BE REMOVED
27	(1B)	CHARACTER	10	REMGENM	GENERATED MEMBER
27	(1B)	CHARACTER	1	REMGMLB	MEMBER TYPE PHASE: 'C', MODULE: 'R'. IN CASE OF MACROS THIS FIELD CONTAINS THE SUBLIB QUALIFIER.
28	(1C)	CHARACTER	1	REMGMCL	MEMBER CLASS MACRO: '.', OTHER: '-'
29	(1D)	CHARACTER	8	REMGNM	MEMBER NAME

CROSS-REFERENCE

REMCMP	1 (1)	REMGNM	29 (1D)
REMCMPGR	1 (1)	REMOVE	0 (0)
REMCMPID	1 (1)	REMPRFF	16 (10)
REMCPLV	10 (A)	REMPROD	13 (D)
REMCMPSF	7 (7)	REMPRPP	13 (D)
REMGENM	27 (1B)	REMP TAP	19 (13)
REMGMCL	28 (1C)	REMTYPE	0 (0)
REMGMLB	27 (1B)		

IKRXRTC

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	14	RETRACE	
0	(0) UNSIGNED	1	RETTYE	TYPE OF HISTORY TO BE RETRACED COMPONENTS: 1, FEATURES: 4, PTFS: 2, APARS: 3, MODULE 5 OVERVIEW: 6
1	(1) CHARACTER 1... ..	13	RETCMPI RETCMP\$	COMPONENT INFORMATION 1:=COMPONENT-ID PRESENT
2	(2) CHARACTER	12	RETCMPID	COMPONENT-ID

CROSS-REFERENCE

RETCMP\$	1 X'80'	RETRACE	0 (0)
RETCMPI	1 (1)	RETTYE	0 (0)
RETCMPID	2 (2)		

IKRXSTL

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	34	IKRXSTL	
0	(0) UNSIGNED	1	ISTOBJ	OBJECT TO INSTALL/RESTORE: SYSRES=1, PRODUCT=2 SERVICE=3, BACKOUT=4
1	(1) UNSIGNED	1	ISTTYPE	TYPE OF ACTIVITY: INSTALL: 1, RESTORE: 2
2	(2) UNSIGNED	1	ISTSOURC	SOURCE FROM WHERE TO PICK UP: FROMTAPE: 1, FROMDISK: 2
3	(3) BITSTRING	1	ISTFLAG	CONTROL FLAG
	1... ..		ISTPR	1:=PRODUCTION PART REQUESTED
	.1.. ..		ISTSV	1:=SERVICE PART(S) REQUESTED
	..1.		ISTOLD	1:=OLD FORMAT SHIPMENT PACKAGE
4	(4) CHARACTER	19	ISTTAPF	TAPEFILE INFO
4	(4) BITSTRING	1		CONTROL FLAG (0:=PRESENT)
	1... ..		ISTTAPF\$	TAPEFILE IDENTIFIER SPECIFIED
	.111 1111			RESERVED
5	(5) CHARACTER	18	ISTTAPFI	TAPEFILE IDENTIFIER
5	(5) CHARACTER	1	QUOTE	QUOTE
6	(6) CHARACTER	16	FILEID	ENTRY WITHOUT QUOTES
22	(16) CHARACTER	1		ENTRY RESERVED FOR QUOTE
23	(17) BITSTRING	1	ISTOLDL	PRESENCE OF OLD FORMAT PRIVATE LIBRARIES ON TAPE (1:=PRESENT)
	1... ..		ISTCL\$	CORE IMAGE LIBRARY
	.1.. ..		ISTR\$	RELO LIBRARY
	..1.		ISTSL\$	SOURCE STATEMENT LIBRARY
24	(18) SIGNED	2	ISTPROD	ID OF THE SET CONTAINING THE TARGET LIBRARY/SUBLIBRARY INFO FOR THE PRODUCTION PART.
26	(1A) SIGNED	2	ISTSERV	ID OF THE SET CONTAINING THE TARGET LIBRARY/SUBLIBRARY INFO FOR THE SERVICE PART.
28	(1C) SIGNED	2	ISTCPT	ID OF THE SET CONTAINING THE LIST OF COMPATIBLE PRODUCTS
30	(1E) SIGNED	2	ISTSUPD	ID OF THE SET CONTAINING THE LIST OF SUPERSEDED PRODUCTS WHICH ARE TO BE DELETED
32	(20) SIGNED	2	ISTOBS	ID OF THE SET CONTAINING THE LIST OF OBSOLETE PRODUCTS

CROSS-REFERENCE

FILEID	6 (6)	ISTR\$	23 X'40'
IKRXSTL	0 (0)	ISTSERV	26 (1A)
ISTCL\$	23 X'80'	ISTSL\$	23 X'20'
ISTCPT	28 (1C)	ISTSOURC	2 (2)
ISTFLAG	3 (3)	ISTSUPD	30 (1E)
ISTOBJ	0 (0)	ISTSV	3 X'40'
ISTOBS	32 (20)	ISTTAPF	4 (4)
ISTOLD	3 X'20'	ISTTAPF\$	4 X'80'
ISTOLDL	23 (17)	ISTTAPFI	5 (5)
ISTPR	3 X'80'	ISTTYPE	1 (1)
ISTPROD	24 (18)	QUOTE	5 (5)

History File Records

The MSHP history file is composed of several different record types, each of which is 64 (X'40') bytes in length. The records are maintained in the IJSYSHF extent and describe the history of all installation and maintenance activity on the VSE system. Each history file record is identified by an identifier of one or two bytes at the beginning of the record. Refer to Figure 15 on page 378 for a list of all possible record ID's and their respective record types. Layout and contents of every history file record depends on its type.

The following pages show the contents of the individual disk record types.

ID	Record type	Chain headers to record type	Chain from Record Type	DSECT Name
A	AIP APARS-in-PTF	NONE	PTF	IKRAIP
B	LIB library name	NONE	INS	IKRLIB
C	COM comments	NONE	PRD,PTF,MIA,GEN	IKRCOM
D	PRD product	COF,COM,MIP,PRE	HDR1	IKRPRD
F	FREE free record	NONE	HDR1	IKRFRE
G	GEN gen.member rec.	IMR,COM(C),COM(K)	INX	IKRGEN
H1	HDR1 header 1	FREE,INS,PRD	NONE (RBA=000000)	IKRHD1
H2	HDR2 header 2	NONE	NONE (RBA=000040)	IKRHD2
H3	HDR3 header 3	HDX	NONE (RBA=000080)	IKRHD3
I	INS installation	APA,LIB,PRE,PTF,INX	HDR1	IKRINS
K	COM comments	NONE	GEN	IKRCOM
L	APA local/APAR fix	MIA	INS	IKRAPA
M	MIP mods-in-PTF	NONE	COF,PRD,PTF	IKRMIP
N	MIA mods-in-APAR	COM,APA	APA	IKRMIA
O	COF comp. of feature	MIP	PRD	IKRCOF
P	PTF PTF installation	AIP,COM,MIP,PRE,SUP,PTX	INS	IKRPTF
Q	PRE pre/co-req	NONE	INS,PTF,PRD	IKRPRE
S	SUP superseded	NONE	PTF	IKRSUP
U	MIP mods-in-PTF	NONE	PTX,MIA	IKRMIP
V	IMT incl.macro table	NONE	INX	IKRIMT
W	IMR incl.macro ref.	NONE	GEN	IKRIMR
X	HDX header extension	NONE	HDR 3	IKRHDX
Y	INX inst.ext.record	IMT,GEN	INS	IKRINX
Z	PTX PTF extension	COM,MIP	PTF	IKRPTX

Figure 15. History File Summary

IKRAIP

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRAIP	APARS-IN-PTF RECORD
0	(0) CHARACTER	1	AIPID	IDENTIFICATION := 'A'
1	(1) CHARACTER	6	AIPQUEUE	APARS-IN-PTF RECORD QUEUE
1	(1) A-ADDRESS	3	AIPFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	AIPBWD	BACKWARD CHAIN
7	(7) CHARACTER	56	AIPENTRY	UP TO 7 APARS-IN-PTF ENTRIES
63	(3F) CHARACTER	1		NOT USED

CROSS-REFERENCE

AIPBWD	4 (4)	AIPID	0 (0)
AIPENTRY	7 (7)	AIPQUEUE	1 (1)
AIPFWD	1 (1)	IKRAIP	0 (0)

IKRAPA

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRAPA	APAR (LOCAL) FIX RECORD
0	(0) CHARACTER	1	APAID	IDENTIFICATION := 'L'
1	(1) CHARACTER	6	APAQUEUE	APAR FIX RECORD QUEUE
1	(1) A-ADDRESS	3	APAFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	APABWD	BACKWARD CHAIN
7	(7) UNSIGNED	1	APAQCNT	NUMBER OF ASSOCIATED QUEUES
8	(8) CHARACTER	6	APAMODQ	MODULE-IN-APAR RECORD QUEUE (ORIGIN)
8	(8) A-ADDRESS	3	APAMODF	1ST MODULE-IN-APAR RECORD
11	(B) A-ADDRESS	3	APAMODL	X'FFFFFF' RBA
14	(E) CHARACTER	6	APACOMQ	COMMENT RECORD QUEUE
14	(E) A-ADDRESS	3	APACOMF	1ST COMMENT RECORD
17	(11) A-ADDRESS	3	APACOML	X'FFFFFF' RBA
20	(14) CHARACTER	6	APAINFQ	INFLUENCED MEMB QUEUE
20	(14) A-ADDRESS	3	APAINFF	1ST COMMENT RECORD
23	(17) A-ADDRESS	3	APAINFL	X'FFFFFF' RBA
26	(1A) CHARACTER	6	APAREQQ	REQUIREMENT QUEUE
26	(1A) A-ADDRESS	3	APAREQF	1ST REQ. RECORD
29	(1D) A-ADDRESS	3	APAREQL	X'FFFFFF' RBA
32	(20) CHARACTER	17		RESERVED
49	(31) CHARACTER	6	APADATE	APPLICATION DATE
55	(37) BITSTRING	1	APARFLAG	STATUS FLAG
	1... ..		APAINCOM	1 := INCOMPLETE FIX
	.1.. ..		APAINVAL	1 := INVALIDATED FIX
	..1.		APAINCOR	1 := INVALID COREQ
	...1		APAPROD	1 := FIX AFFECTS PRODUCTION SUBLIB ONLY
56	(38) CHARACTER	8	APARNO	APAR NUMBER

CROSS-REFERENCE

APABWD	4 (4)	APAMODF	8 (8)
APACOMF	14 (E)	APAMODL	11 (B)
APACOML	17 (11)	APAMODQ	8 (8)
APACOMQ	14 (E)	APAPROD	55 X'10'
APADATE	49 (31)	APAQCNT	7 (7)
APAFWD	1 (1)	APAQUEUE	1 (1)
APAID	0 (0)	APAREQF	26 (1A)
APAINCOM	55 X'80'	APAREQL	29 (1D)
APAINCOR	55 X'20'	APAREQQ	26 (1A)
APAINFF	20 (14)	APARFLAG	55 (37)
APAINFL	23 (17)	APARNO	56 (38)
APAINFQ	20 (14)	IKRAPA	0 (0)
APAINVAL	55 X'40'		

IKRCOF

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRCOF	"COF" RECORD
0	(0) CHARACTER	1	COFID	IDENTIFICATION := '0'
1	(1) CHARACTER	6	COFQUEUE	"COF " RECORD QUEUE
1	(1) A-ADDRESS	3	COFFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	COFBWD	BACKWARD CHAIN
7	(7) CHARACTER	6	COFMODQ	LIB MEMBER QUEUE (ORIGIN)
7	(7) A-ADDRESS	3	COFMODF	1ST MOD RECORD
10	(A) A-ADDRESS	3	COFMODL	X'FFFFFF' RBA
13	(D) CHARACTER	35		NOT USED
48	(30) CHARACTER	6	COFREL	RELEASE IDENTIFICATION
48	(30) CHARACTER	2	COFRELVV	VERSION
50	(32) CHARACTER	2	COFRELRR	RELEASE
52	(34) CHARACTER	2	COFRELMM	MAINTENANCE
54	(36) CHARACTER	9	COFCMPID	COMPONENT-ID
63	(3F) BITSTRING	1	COFFLAG	STATUS FLAG (UNUSED)

CROSS-REFERENCE

COFBWD	4 (4)	COFMODQ	7 (7)
COFCMPID	54 (36)	COFQUEUE	1 (1)
COFFLAG	63 (3F)	COFREL	48 (30)
COFFWD	1 (1)	COFRELMM	52 (34)
COFID	0 (0)	COFRELRR	50 (32)
COFMODF	7 (7)	COFRELVV	48 (30)
COFMODL	10 (A)	IKRCOF	0 (0)

IKRCOM

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRCOM	COMMENT RECORD
0	(0) CHARACTER	1	COMID	IDENTIFICATION := 'C'
1	(1) CHARACTER	6	COMQUEUE	COMMENT RECORD QUEUE
1	(1) A-ADDRESS	3	COMFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	COMBWD	BACKWARD CHAIN
7	(7) CHARACTER	57	COMMENT	ADDITIONAL INFORMATION

CROSS-REFERENCE

COMBWD	4 (4)	COMMENT	7 (7)
COMFWD	1 (1)	COMQUEUE	1 (1)
COMID	0 (0)	IKRCOM	0 (0)

IKRFRE

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRFRE	UNUSED RECORD
0	(0) CHARACTER	1	FREID	IDENTIFICATION := 'F'
1	(1) CHARACTER	6	FREQUEUE	FREE RECORD QUEUE
1	(1) A-ADDRESS	3	FREFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	FREBWD	BACKWARD CHAIN
7	(7) CHARACTER	57		FILLED WITH ZEROS

CROSS-REFERENCE

FREBWD	4 (4)	FREQUEUE	1 (1)
FREFWD	1 (1)	IKRFRE	0 (0)
FREID	0 (0)		

IKRGEN

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRGEN	GENERATED MEMBER RECORD
0	(0) CHARACTER	1	GENID	IDENTIFICATION := 'G'
1	(1) CHARACTER	6	GENQUEUE	GENERATED MEMBER RECORD QUEUE
1	(1) A-ADDRESS	3	GENFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	GENBWD	BACKWARD CHAIN
7	(7) UNSIGNED	1	GENQCNT	NO.OF ASSOCIATED CHAINS
8	(8) CHARACTER	6	GENIMRQ	INCL. MACRO REF. RECORD QUEUE
8	(8) A-ADDRESS	3	GENIMRF	1ST INCL. MACRO RECORD
11	(B) A-ADDRESS	3	GENIMRL	X'FFFFFF' RBA
14	(E) CHARACTER	6	GENPDQ	PROCESSED DATA RECORD QUEUE
14	(E) A-ADDRESS	3	GENPDF	1ST PROCESSED DATA RECORD
17	(11) A-ADDRESS	3	GENPDL	X'FFFFFF' RBA
20	(14) CHARACTER	6	GENCRQ	COMMENT RECORD QUEUE
20	(14) A-ADDRESS	3	GENCRF	1ST COMMENT MEMBER RECORD
23	(17) A-ADDRESS	3	GENCRL	X'FFFFFF' RBA
26	(1A) UNSIGNED	4	GENBLKS	OCCUPIED LIBRARY BLOCKS
30	(1E) CHARACTER	17		RESERVED FOR FUTURE USE
47	(2F) CHARACTER	10	GENMN	MEMBER NAME
47	(2F) CHARACTER	1	GENMNLB	LIBRARY TYPE IS : C- FOR PHASES R- FOR MODULES, FOR MACROS SUBLIB QUALIFIER (DEFAULT-E)
48	(30) CHARACTER	1	GENMNCL	CONCERNED CLASS IS: MACRO='.', OTHER = '-'
49	(31) CHARACTER	8	GENMNM	NAME OF GEN. MEMBER PADDED WITH TRAILING BLANKS
57	(39) CHARACTER	6	GENGDAT	GENERATION DATE
63	(3F) BITSTRING	1	GENFLAG	STATUS FLAG
	1... ..			RESERVED
	.1..		GENREG	MEMBER REGRESSED
	..1.		GENDLB	MEMBER GENERATED DLIBBUILD
	...1		GENKPD	MEMBER GENERATED KEEPDATA
 1...		GENXRF	MEMBER GENERATED XREF
111			RESERVED

CROSS-REFERENCE

GENBLKS	26 (1A)	GENKPD	63 X'10'
GENBWD	4 (4)	GENMN	47 (2F)
GENCRF	20 (14)	GENMNCL	48 (30)
GENCRL	23 (17)	GENMNLB	47 (2F)
GENCRQ	20 (14)	GENMNM	49 (31)
GENDLB	63 X'20'	GENPDF	14 (E)
GENFLAG	63 (3F)	GENPDL	17 (11)
GENFWD	1 (1)	GENPDQ	14 (E)
GENGDAT	57 (39)	GENQCNT	7 (7)
GENID	0 (0)	GENQUEUE	1 (1)
GENIMRF	8 (8)	GENREG	63 X'40'
GENIMRL	11 (B)	GENXRF	63 X'08'
GENIMRQ	8 (8)	IKRGEN	0 (0)

IKRHD1

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0)	STRUCTURE	64 IKRHD1	HEADER 1 RECORD
0	(0)	CHARACTER	2 HD1ID	IDENTIFICATION := 'H1'
0	(0)	CHARACTER	1 HDID	GENERAL HEADER ID := 'H'
1	(1)	CHARACTER	1	FILLER
2	(2)	CHARACTER	6 HD1PRDQ	PRODUCT RECORD QUEUE (ORIGIN)
2	(2)	A-ADDRESS	3 HD1PRDF	1ST PRODUCT RECORD
5	(5)	A-ADDRESS	3 HD1PRDL	X'FFFFFF' RBA
8	(8)	CHARACTER	6 HD1INSQ	INSTALLATION RECORD QUEUE (ORIGIN)
8	(8)	A-ADDRESS	3 HD1INSF	1ST INSTALLATION RECORD
11	(B)	A-ADDRESS	3 HD1INSL	X'FFFFFF' RBA
14	(E)	CHARACTER	6 HD1FREEQ	FREE RECORD QUEUE (ORIGIN)
14	(E)	A-ADDRESS	3 HD1FREEF	1ST FREE RECORD
17	(11)	A-ADDRESS	3 HD1FREEL	X'FFFFFF' RBA
20	(14)	CHARACTER	20 HD1CUST	CUSTOMER NAME
20	(14)	CHARACTER	4 HD1NOCST	PRESENT CHECK FIELD
24	(18)	CHARACTER	16	FILLER
40	(28)	CHARACTER	24 HD1SYSPG	SYSTEM PROGRAMMER NAME

CROSS-REFERENCE

HDID	0 (0)	HD1INSQ	8 (8)
HD1CUST	20 (14)	HD1NOCST	20 (14)
HD1FREEF	14 (E)	HD1PRDF	2 (2)
HD1FREEL	17 (11)	HD1PRDL	5 (5)
HD1FREEQ	14 (E)	HD1PRDQ	2 (2)
HD1ID	0 (0)	HD1SYSPG	40 (28)
HD1INSF	8 (8)	IKRHD1	0 (0)
HD1INSL	11 (B)		

IKRHD2

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRHD2	HEADER 2 RECORD
0	(0) CHARACTER	2	HD2ID	IDENTIFICATION := 'H2'
2	(2) CHARACTER	17	HD2PHONE	CUSTOMER TELEPHONE NUMBER
19	(13) CHARACTER	45	HD2ADDR	CUSTOMER ADDRESS
19	(13) CHARACTER	4	HD2NOADR	PRESENT CHECK FIELD
23	(17) CHARACTER	41		FILLER

CROSS-REFERENCE

HD2ADDR	19 (13)	HD2PHONE	2 (2)
HD2ID	0 (0)	IKRHD2	0 (0)
HD2NOADR	19 (13)		

IKRHD3

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRHD3	HEADER 3 RECORD
0	(0) CHARACTER	2	HD3ID	IDENTIFICATION := 'H3'
2	(2) CHARACTER	56	HD3ENV	ENVIRONMENT (1ST PART)
58	(3A) CHARACTER	6	HD3HDXQ	HEADER EXTENSION QUEUE
58	(3A) A-ADDRESS	3	HD3HDXF	1ST EXTENSION RECORD
61	(3D) A-ADDRESS	3	HD3HDXL	X'FFFFFFFF' RBA

CROSS-REFERENCE

HD3ENV	2 (2)	HD3HDXQ	58 (3A)
HD3HDXF	58 (3A)	HD3ID	0 (0)
HD3HDXL	61 (3D)	IKRHD3	0 (0)

IKRHDX

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRHDX	HEADER EXTENSION RECORD
0	(0) CHARACTER	1	HDXID	IDENTIFICATION := 'X'
1	(1) CHARACTER	6	HDXQUEUE	HEADER EXTENSION RECORD QUEUE
1	(1) A-ADDRESS	3	HDXFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	HDXBWD	BACKWARD CHAIN
7	(7) CHARACTER	6	HDXENV	ENVIRONMENT (CONTINUED)
13	(D) UNSIGNED	1	HDXREL	RELEASE IDENTIFIER
14	(E) A-ADDRESS	3	HDXHWM	HIGHEST RBA EVER USED
17	(11) SIGNED	4	HDXFRE#	NUMBER OF FREE RECORDS@D14CDMO
21	(15) CHARACTER	6	HDXVRL	RELEASE FOR RETRACE
21	(15) CHARACTER	2	HDXVV	VERSION OF BASE PROD.
23	(17) CHARACTER	2	HDXRR	RELEASE OF BASE PROD.
25	(19) CHARACTER	2	HDXLL	REFRESH LEVEL
27	(1B) CHARACTER	37		RESERVED

CROSS-REFERENCE

HDXBWD	4 (4)	HDXQUEUE	1 (1)
HDXENV	7 (7)	HDXREL	13 (D)
HDXFRE#	17 (11)	HDXRR	23 (17)
HDXFWD	1 (1)	HDXVRL	21 (15)
HDXHWM	14 (E)	HDXVV	21 (15)
HDXID	0 (0)	IKRHDX	0 (0)
HDXLL	25 (19)		

IKRIMR

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRIMR	INCLUDED MACRO REFER. RECORD
0	(0) CHARACTER	1	IMRID	IDENTIFICATION := 'W'
1	(1) CHARACTER	6	IMRQUEUE	INCLUDED MACRO RECORD QUEUE
1	(1) A-ADDRESS	3	IMRFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	IMRBWD	BACKWARD CHAIN
7	(7) UNSIGNED	1	IMRACT#	ACTUAL COUNT OF ENTRIES
8	(8) SIGNED	50	IMRIME	INCL. MACRO REF. ENTRIES
58	(3A) CHARACTER	6		RESERVED FOR FUTURE USE

CROSS-REFERENCE

IKRIMR	0 (0)	IMRID	0 (0)
IMRACT#	7 (7)	IMRIME	8 (8)
IMRBWD	4 (4)	IMRQUEUE	1 (1)
IMRFWD	1 (1)		

IKRIMT

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRIMT	INCLUDED MACRO TABLE RECORD
0	(0) CHARACTER	1	IMTID	IDENTIFICATION := 'V'
1	(1) CHARACTER	6	IMTQUEUE	INCLUDED MACRO RECORD QUEUE
1	(1) A-ADDRESS	3	IMTFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	IMTBWD	BACKWARD CHAIN
7	(7) UNSIGNED	1	IMTACT#	ACTUAL COUNT OF ENTRIES
8	(8) CHARACTER	55	IMTMET	MACRO ENTRIES AS TABLE
8	(8) UNSIGNED	1	IMTMCO	CONTROL BYTE FOR THIS MACRO
9	(9) CHARACTER	1	IMTMLB	LIBRARY AFFECTED
10	(A) CHARACTER	1	IMTMCL	CLASS AFFECTED
11	(B) CHARACTER	8	IMTMNM	NAME OF MACRO, LEFT ALIGNED PADDED WITH TRAILING BLANKS
63	(3F) CHARACTER	1		RESERVED FOR FUTURE USE

CROSS-REFERENCE

IKRIMT	0 (0)	IMTMCO	8 (8)
IMTACT#	7 (7)	IMTMET	8 (8)
IMTBWD	4 (4)	IMTMLB	9 (9)
IMTFWD	1 (1)	IMTMNM	11 (B)
IMTID	0 (0)	IMTQUEUE	1 (1)
IMTMCL	10 (A)		

IKRINS

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRINS	INSTALLATION RECORD
0	(0) CHARACTER	1	INSID	IDENTIFICATION := 'I'
1	(1) CHARACTER	6	INSQUEUE	INSTALLATION RECORD QUEUE
1	(1) A-ADDRESS	3	INSFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	INSBWD	BACKWARD CHAIN
7	(7) CHARACTER	6	INSPTFQ	PFT RECORD QUEUE (ORIGIN)
7	(7) A-ADDRESS	3	INSPTFF	1ST PTF RECORD
10	(A) A-ADDRESS	3	INSPTFL	X'FFFFFF' RBA
13	(D) CHARACTER	6	INSAPARQ	APAR FIX RECORD QUEUE (ORIGIN)
13	(D) A-ADDRESS	3	INSAPARF	1ST APAR FIX RECORD
16	(10) A-ADDRESS	3	INSAPARL	X'FFFFFF' RBA
19	(13) CHARACTER	6	INSPREQ	PRE/CO-REQUISITE-INFO RECORD QUEUE (ORIGIN)
19	(13) A-ADDRESS	3	INSPREF	1ST PRE/CO-REQUISITE-INFO RECORD
22	(16) A-ADDRESS	3	INSPREL	X'FFFFFF' RBA
25	(19) CHARACTER	6	INSDATE	INSTALLATION DATE
31	(1F) CHARACTER	6	INSREL	RELEASE IDENTIFICATION
31	(1F) CHARACTER	2	INSRELVV	VERSION
33	(21) CHARACTER	2	INSRELRR	RELEASE
35	(23) CHARACTER	2	INSRELMM	MAINTENANCE
37	(25) CHARACTER	12	INSCMP	COMPONENT-ID
37	(25) CHARACTER	9	INSCMPID	COMPONENT-NAME
46	(2E) CHARACTER	3	INSCMPLV	COMPONENT-LEVEL
49	(31) CHARACTER	2		NOT USED
51	(33) CHARACTER	6	INSEXTQ	INSTALL. EXT. QUEUE
51	(33) A-ADDRESS	3	INSEXTF	1ST INST.EXT.RECORD
54	(36) A-ADDRESS	3	INSEXTL	X'FFFFFF' RBA
57	(39) CHARACTER	6	INSLIBQ	LIBRARY NAMES QUEUE
57	(39) A-ADDRESS	3	INSLIBF	1ST LIBRARY NAMES RECORD
60	(3C) A-ADDRESS	3	INSLIBL	X'FFFFFF' RBA
63	(3F) BITSTRING	1	INSFLAG	STATUS FLAG
	11..			NOT USED
	..1.		INSINCOR	COMPONENT INCORRECTLY APPLIED
	...1		INSSOFT	SOFT PTF REJECTION ACCEPTED
 1...		INSPROD	PRODUCTION PART ONLY
1..		INSSERV	SERVICE PART ONLY
1.		INSDEF	DEFAULTING REQUESTED
1		INSOBSOL	COMPONENT IS OBSOLETE

CROSS-REFERENCE

IKRINS	0 (0)	INSLIBL	60 (3C)
INSAPARF	13 (D)	INSLIBQ	57 (39)
INSAPARL	16 (10)	INSOBSOL	63 X'01'
INSAPARQ	13 (D)	INSPREF	19 (13)
INSBWD	4 (4)	INSPREL	22 (16)
INSCMP	37 (25)	INSPREQ	19 (13)
INSCMPID	37 (25)	INSPROD	63 X'08'
INSCMPLV	46 (2E)	INSPTFF	7 (7)
INSDATE	25 (19)	INSPTFL	10 (A)
INSDEF	63 X'02'	INSPTFQ	7 (7)
INSEXTF	51 (33)	INSQUEUE	1 (1)
INSEXTL	54 (36)	INSREL	31 (1F)
INSEXTQ	51 (33)	INSRELMM	35 (23)
INSFLAG	63 (3F)	INSRELRR	33 (21)
INSFWD	1 (1)	INSRELVV	31 (1F)
INSID	0 (0)	INSSERV	63 X'04'
INSINCOR	63 X'20'	INSSOFT	63 X'10'
INSLIBF	57 (39)		

IKRINX

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRINX	INSTALLATION EXTENSION RECORD
0	(0) CHARACTER	1	INXID	IDENTIFICATION := 'Y'
1	(1) CHARACTER	6	INXQUEUE	INST. EXTENSION RECORD QUEUE
1	(1) A-ADDRESS	3	INXFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	INXBWD	BACKWARD CHAIN
7	(7) UNSIGNED	1	INXQCNT	NUMBER OF ASSOCIATED QUEUES
8	(8) CHARACTER	6	INXGENQ	GENERATED MEMBERS ENTRY QUEU
8	(8) A-ADDRESS	3	INXGENF	1ST GEN. MEMBERS ENTRY RECORD
11	(B) A-ADDRESS	3	INXGENL	X'FFFFFF' RBA
14	(E) CHARACTER	6	INXIMTQ	INCLUDED MACRO TABLE RECORD Q
14	(E) A-ADDRESS	3	INXIMTF	1ST INCL.MACRO TABLE RECORD
17	(11) A-ADDRESS	3	INXIMTL	X'FFFFFF' RBA
20	(14) CHARACTER	44		RESERVED

CROSS-REFERENCE

IKRINX	0 (0)	INXID	0 (0)
INXBWD	4 (4)	INXIMTF	14 (E)
INXFWD	1 (1)	INXIMTL	17 (11)
INXGENF	8 (8)	INXIMTQ	14 (E)
INXGENL	11 (B)	INXQCNT	7 (7)
INXGENQ	8 (8)	INXQUEUE	1 (1)

IKRLIB

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRLIB	LIBRARY-NAME HISTORY FILE RECORD
0	(0) CHARACTER	1	LIBID	IDENTIFIER 'B'
1	(1) CHARACTER	6	LIBQUEUE	QUEUE CHAIN POINTERS
1	(1) A-ADDRESS	3	LIBFWD	FORWARD POINTER
4	(4) A-ADDRESS	3	LIBBWD	BACKWARD POINTER
7	(7) UNSIGNED	1	LIB#	NUMBER OF LIBRARY INFOS
8	(8) CHARACTER	48	LIBNAME	PROD/SERV LIBRARY INFO
8	(8) CHARACTER	1	LIBTP	LIBRARY TYPE (PROD/SERV)
9	(9) CHARACTER	7	LIBLBNM	LIBRARY NAME
16	(10) CHARACTER	8	LIBSLNM	SUBLIBRARY NAME
56	(38) CHARACTER	8		RESERVED

CROSS-REFERENCE

IKRLIB	0 (0)	LIBLBNM	9 (9)
LIB#	7 (7)	LIBNAME	8 (8)
LIBBWD	4 (4)	LIBQUEUE	1 (1)
LIBFWD	1 (1)	LIBSLNM	16 (10)
LIBID	0 (0)	LIBTP	8 (8)

IKRLSI

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	12	IKRLSI	LIBR SEND APPENDAGE INTERFACE
0	(0) UNSIGNED	1	LSIFUNC	REQUESTED FUNCTION
4	(4) A-ADDRESS	4	LSIPARM1	BASE FOR 1ST PARAMETER
8	(8) A-ADDRESS	4	LSIPARM2	BASE FOR 2ND PARAMETER

CROSS-REFERENCE

IKRLSI	0 (0)	LSIPARM1	4 (4)
LSIFUNC	0 (0)	LSIPARM2	8 (8)

IKRMIA

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRMIA	MODULE-IN-APAR RECORD
0	(0) CHARACTER	1	MIAID	IDENTIFICATION := 'N'
1	(1) CHARACTER	6	MIAQUEUE	MODULE-IN-APAR RECORD QUEUE
1	(1) A-ADDRESS	3	MIAFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	MIABWD	BACKWARD CHAIN
7	(7) CHARACTER	10	MIANAME	MODULE IDENTIFIER
7	(7) CHARACTER	1	MIALB	LIBRARY (PHASE:=C, MODULE:=R, SUBLIBRARY QUALIFIER FOR MACROS)
8	(8) CHARACTER	1	MIACL	CLASS (MACRO:='.', LIOCS:='+' OTHER:='-')
9	(9) CHARACTER	8	MIANM	MODULE NAME
17	(11) CHARACTER	41	MIAFIX	FIX DESCRIPTIONS
58	(3A) CHARACTER	6	MIAFIXQ	FIX-COMMENT QUEUE (ORIGIN)
58	(3A) A-ADDRESS	3	MIAFIXF	1ST COMMENT RECORD
61	(3D) A-ADDRESS	3	MIAFIXL	X'FFFFFF' RBA

CROSS-REFERENCE

IKRMIA	0 (0)	MIAFWD	1 (1)
MIABWD	4 (4)	MIAID	0 (0)
MIACL	8 (8)	MIALB	7 (7)
MIAFIX	17 (11)	MIANAME	7 (7)
MIAFIXF	58 (3A)	MIANM	9 (9)
MIAFIXL	61 (3D)	MIAQUEUE	1 (1)
MIAFIXQ	58 (3A)		

IKRMIP

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRMIP	MODULES-IN-PTF RECORD
0	(0) CHARACTER	1	MIPID	IDENTIFICATION:='M','U'
1	(1) CHARACTER	6	MIPQUEUE	MODULES-IN-PTF RECORD QUEUE
1	(1) A-ADDRESS	3	MIPFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	MIPBWD	BACKWARD CHAIN
7	(7) CHARACTER	50	MIPENTRY	1-5 MODULE-IN-PTF ENTRIES
7	(7) CHARACTER	1	MIPLB	LIBRARY: PHASE:='C', MODULE:='R', INVOLVES:='I' SUBLIB QUALIFIER FOR MACROS
8	(8) CHARACTER	1	MIPCL	CLASS: MACRO:='.', LIOCS:='+' OTHER:='- ', LINKBOOK:='+'
9	(9) CHARACTER	8	MIPNM	MODULE OR LINKBOOK NAME
57	(39) CHARACTER	7		NOT USED

CROSS-REFERENCE

IKRMIP	0 (0)	MIPID	0 (0)
MIPBWD	4 (4)	MIPLB	7 (7)
MIPCL	8 (8)	MIPNM	9 (9)
MIPENTRY	7 (7)	MIPQUEUE	1 (1)
MIPFWD	1 (1)		

IKRPRD

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRPRD	PRODUCT RECORD
0	(0) CHARACTER	1	PRDID	IDENTIFICATION := 'D'
1	(1) CHARACTER	6	PRDQUEUE	PRODUCT RECORD QUEUE
1	(1) A-ADDRESS	3	PRDFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	PRDBWD	BACKWARD CHAIN
7	(7) UNSIGNED	1	PRDQCNT	NUMBER OF ASSOCIATED QUEUES
8	(8) CHARACTER	6	PRDMODQ	MODULE RECORD QUEUE (ORIGIN)
8	(8) A-ADDRESS	3	PRDMODF	1ST MODULE RECORD
11	(B) A-ADDRESS	3	PRDMODL	X'FFFFFF' RBA
14	(E) CHARACTER	6	PRDREPQ	REPLACED COMPON. RECORD QUEUE (ORIGIN)
14	(E) A-ADDRESS	3	PRDREPF	1ST REPL. COMPONENT RECORD
17	(11) A-ADDRESS	3	PRDREPL	X'FFFFFF' RBA
20	(14) CHARACTER	6	PRDPREQ	PRE/CO-REQUISITE-INFO RECORD QUEUE (ORIGIN)
20	(14) A-ADDRESS	3	PRDPREF	1ST PRE/CO-REQUISITE-INFO RECORD
23	(17) A-ADDRESS	3	PRDPREL	X'FFFFFF' RBA
26	(1A) CHARACTER	6	PRDCOMMQ	PRODUCT COMMENT RECORD QUEUE (ORIGIN)
26	(1A) A-ADDRESS	3	PRDCOMMF	1ST PRODUCT COMMENT RECORD
29	(1D) A-ADDRESS	3	PRDCOMML	X'FFFFFF' RBA
32	(20) CHARACTER	6	PRDDATE	APPLICATION DATE
38	(26) CHARACTER	7	PRDBLKS	REQUIRED BLOCKS FOR LIBRARIES
38	(26) CHARACTER	2	PRDCIL	BLOCKS FOR CORE IMAGE LIBRARY
40	(28) CHARACTER	2	PRDRL	BLOCKS FOR RELOCATE LIBRARY
42	(2A) CHARACTER	3	PRDSL	BLOCKS FOR SOURCE LIBRARY
45	(2D) CHARACTER	3	PRDNO	PRODUCT NUMBER
48	(30) CHARACTER	6	PRDPL	PRODUCT ID
48	(30) CHARACTER	3	PRDPPP	PRODUCT CODE
51	(33) CHARACTER	3	PRDRFF	PRODUCT LEVEL
54	(36) CHARACTER	10		NOT USED

CROSS-REFERENCE

IKRPRD	0 (0)	PRDPL	48 (30)
PRDBLKS	38 (26)	PRDPPP	48 (30)
PRDBWD	4 (4)	PRDPREF	20 (14)
PRDCIL	38 (26)	PRDPREL	23 (17)
PRDCOMMF	26 (1A)	PRDPREQ	20 (14)
PRDCOMML	29 (1D)	PRDQCNT	7 (7)
PRDCOMMQ	26 (1A)	PRDQUEUE	1 (1)
PRDDATE	32 (20)	PRDREPF	14 (E)
PRDFWD	1 (1)	PRDREPL	17 (11)
PRDID	0 (0)	PRDREPQ	14 (E)
PRDMODF	8 (8)	PRDRFF	51 (33)
PRDMODL	11 (B)	PRDRL	40 (28)
PRDMODQ	8 (8)	PRDSL	42 (2A)
PRDNO	45 (2D)		

IKRPRE

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	53	IKRPRE	PRE/CO-REQUISITE-INFO RECORD
0	(0) CHARACTER	1	PREID	IDENTIFICATION := 'Q'
1	(1) CHARACTER	6	PREQUEUE	PRE/CO-REQUISITE-INFO RECORD QUEUE
1	(1) A-ADDRESS	3	PREFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	PREBWD	BACKWARD CHAIN
7	(7) CHARACTER	46	PREENTRY	1-2 PRE/CO-REQUISITE-INFO ENTRIES
7	(7) SIGNED	2	PREORQ	INDEX OF FIRST OR ONLY 'OR' REQUIREMENT
9	(9) SIGNED	2	PREANDQ	INDEX OF NEXT 'AND' REQUIR.
11	(B) UNSIGNED	1	PRESWTCH	'1' := PRE-REQ INFORMATION '3' := NEGATIVE PRE-REQ INFO '2' := CO-REQ INFORMATION
12	(C) UNSIGNED	1	PREOWNER	'2' := PTF OR APAR '4' := FEATURE '1' := COMPONENT
13	(D) CHARACTER	9	PRECMPNO	COMPONENT-ID
13	(D) CHARACTER	12	PRECMPID	COMPONENT AND LEVEL
13	(D) CHARACTER	6	PREPRDNO	PRODUCT WITH LEVEL-ID
13	(D) CHARACTER	3	PREPRPPP	PRODUCT ID
16	(10) CHARACTER	3	PREPRRFF	LEVEL ID
19	(13) CHARACTER	3		
22	(16) CHARACTER	8	PREPTFNO	PTF/APAR# OR ZEROS

CROSS-REFERENCE

IKRPRE	0 (0)	PREORQ	7 (7)
PREANDQ	9 (9)	PREOWNER	12 (C)
PREBWD	4 (4)	PREPRDNO	13 (D)
PRECMPID	13 (D)	PREPRPPP	13 (D)
PRECMPNO	13 (D)	PREPRRFF	16 (10)
PREENTRY	7 (7)	PREPTFNO	22 (16)
PREFWD	1 (1)	PREQUEUE	1 (1)
PREID	0 (0)	PRESWTCH	11 (B)

IKRPTF

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRPTF	PTF RECORD
0	(0) CHARACTER	1	PTFID	IDENTIFICATION := 'P'
1	(1) CHARACTER	6	PTFQUEUE	PTF RECORD QUEUE
1	(1) A-ADDRESS	3	PTFFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	PTFBWD	BACKWARD CHAIN
7	(7) UNSIGNED	1	PTFQCNT	NUMBER OF ASSOCIATED QUEUES
8	(8) CHARACTER	6	PTFAPARQ	APARS-IN-PTF RECORD QUEUE (ORIGIN)
8	(8) A-ADDRESS	3	PTFAPARF	1ST APARS-IN-PTF RECORD
11	(B) A-ADDRESS	3	PTFAPARL	X'FFFFFF' RBA
14	(E) CHARACTER	6	PTFMODQ	MODULE IN PTF RECORD QUEUE (ORIGIN)
14	(E) A-ADDRESS	3	PTFMODF	1ST MODULES-IN-PTF RECORD
17	(11) A-ADDRESS	3	PTFMODL	X'FFFFFF' RBA
20	(14) CHARACTER	6	PTFSUPQ	SUPERCEDED-PTF'S RECORD QUEUE (ORIGIN)
20	(14) A-ADDRESS	3	PTFSUPF	1ST SUPERCEDED-PTF'S RECORD
23	(17) A-ADDRESS	3	PTFSUPL	X'FFFFFF' RBA
26	(1A) CHARACTER	6	PTFPREQ	PRE/CO-REQUISITE-INFO RECORD QUEUE (ORIGIN)
26	(1A) A-ADDRESS	3	PTFPREF	1ST PRE/CO-REQUISITE-INFO RECORD
29	(1D) A-ADDRESS	3	PTFPREL	X'FFFFFF' RBA
32	(20) CHARACTER	6	PTFCOMMQ	PTF-COMMENT RECORD QUEUE ANCHOR, USED AS ANCHOR TO INFLUENCED MEMBERS, IF PTFEXT='1'B
32	(20) A-ADDRESS	3	PTFCOMMF	1ST PTF-COMMENT RECORD
35	(23) A-ADDRESS	3	PTFCOMML	X'FFFFFF' RBA
38	(26) CHARACTER	6	PTFDATE	APPLICATION DATE
44	(2C) CHARACTER	7	PTFBLKS	REQUIRED BLOCKS FOR LIBRARIES
44	(2C) CHARACTER	2	PTFCIL	BLOCKS FOR CORE IMAGE LIBRARY
46	(2E) CHARACTER	2	PTFRL	BLOCKS FOR RELOCATE LIBRARY
48	(30) CHARACTER	3	PTFSL	BLOCKS FOR SOURCE LIBRARY
51	(33) CHARACTER	8	PTFNO	PTF NUMBER
59	(3B) CHARACTER	4		NOT USED
63	(3F) BITSTRING	1	PTFFLAG	STATUS FLAG
	1... ..		PTFBACK	1 := PTF IS BACKOUT'ED
	.1.. ..		PTFSUPER	1 := PTF IS SUPERCEDED
	..1.		PTFINCOR	1 := PTF INCORRECT APPLIED
	...1		PTFEXT	1 := PTF EXTENDED
 1...		PTFNOLK	1 := PTF NOT LINKED

CROSS-REFERENCE

IKRPTF	0 (0)	PTFMODF	14 (E)
PTFAPARF	8 (8)	PTFMODL	17 (11)
PTFAPARL	11 (B)	PTFMODQ	14 (E)
PTFAPARQ	8 (8)	PTFNO	51 (33)
PTFBACK	63 X'80'	PTFNOTLK	63 X'08'
PTFBLKS	44 (2C)	PTFPREF	26 (1A)
PTFBWD	4 (4)	PTFPREL	29 (1D)
PTFCIL	44 (2C)	PTFPREQ	26 (1A)
PTFCOMMF	32 (20)	PTFQCNT	7 (7)
PTFCOMML	35 (23)	PTFQUEUE	1 (1)
PTFCOMMQ	32 (20)	PTFRL	46 (2E)
PTFDATE	38 (26)	PTFSL	48 (30)
PTFEXT	63 X'10'	PTFSUPER	63 X'40'
PTFFLAG	63 (3F)	PTFSUPF	20 (14)
PTFFWD	1 (1)	PTFSUPL	23 (17)
PTFID	0 (0)	PTFSUPQ	20 (14)
PTFINCOR	63 X'20'		

IKRPTX

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRPTX	PTF EXTENSION RECORD
0	(0) CHARACTER	1	PTXID	IDENTIFICATION := 'Z'
1	(1) CHARACTER	6	PTXQUEUE	PTF EXTENSION RECORD QUEUE
1	(1) A-ADDRESS	3	PTXFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	PTXBWD	BACKWARD CHAIN
7	(7) UNSIGNED	1	PTXQCNT	NUMBER OF ASSOCIATED QUEUES
8	(8) CHARACTER	6	PTXCOMQ	COMMENT RECORD QUEUE ANCHOR
8	(8) A-ADDRESS	3	PTXCOMF	1ST COMMENT RECORD
11	(B) A-ADDRESS	3	PTXCOML	X'FFFFFF' RBA
14	(E) CHARACTER	6	PTXINFQ	INFLUENCED MEMBER RECORD ANCH.
14	(E) A-ADDRESS	3	PTXINFF	1ST INFLUENCED MEMBER RECORD
17	(11) A-ADDRESS	3	PTXINFL	X'FFFFFF' RBA
20	(14) CHARACTER	44		RESERVED

CROSS-REFERENCE

IKRPTX	0 (0)	PTXID	0 (0)
PTXBWD	4 (4)	PTXINFF	14 (E)
PTXCOMF	8 (8)	PTXINFL	17 (11)
PTXCOML	11 (B)	PTXINFQ	14 (E)
PTXCOMQ	8 (8)	PTXQCNT	7 (7)
PTXFWD	1 (1)	PTXQUEUE	1 (1)

IKRSUP

<u>OFFSETS</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0	(0) STRUCTURE	64	IKRSUP	SUPERCEDED-PTF'S RECORD
0	(0) CHARACTER	1	SUPID	IDENTIFICATION := 'S'
1	(1) CHARACTER	6	SUPQUEUE	SUPERCEDED-PTF'S RECORD QUEUE
1	(1) A-ADDRESS	3	SUPFWD	FORWARD CHAIN
4	(4) A-ADDRESS	3	SUPBDW	BACKWARD CHAIN
7	(7) CHARACTER	56	SUPENTRY	1-7 SUPERCEDED PTF ENTRIES
63	(3F) CHARACTER	1		NOT USED

CROSS-REFERENCE

IKRSUP	0 (0)	SUPFWD	1 (1)
SUPBDW	4 (4)	SUPID	0 (0)
SUPENTRY	7 (7)	SUPQUEUE	1 (1)

Chapter 5: Diagnostic Aids

This chapter shows the relationship between modules and messages, and explains the dump of a history file.

Module-Message Cross-Reference Lists

Part 1 : Sorted by Modules

This chart lists in alphameric order all modules which issue messages and the applicable message number(s).

Part 2 : Sorted by Messages

This chart lists in alphameric order all messages issued by modules and the name of the issuing module(s).

Messages issued by the parser follow the format 4Dxxx. For modules that issue these messages refer to the **VSE/Advanced Functions Diagnosis Reference: Service Aids**.

Module-Message Interrelations

Part 1: Sorted by Modules

IKRAAPAR	CAUSES	MSG100	MSG103			
IKRACOMP	CAUSES	MSG103				
IKRAGEN	CAUSES	MSG067	MSG100	MSG419	MSG420	
IKRAPPLY	CAUSES	MSG100	MSG121	MSG275	MSG276	
IKRAPRD	CAUSES	MSG103	MSG114			
IKRAPTF	CAUSES	MSG100	MSG103			
IKRKBKPR	CAUSES	MSG066	MSG081	MSG083	MSG100	MSG101
		MSG102	MSG112	MSG131	MSG241	MSG242
		MSG243	MSG244	MSG246	MSG250	MSG306
		MSG307				
IKRKBKUP	CAUSES	MSG061	MSG081	MSG083		
IKRCGT	CAUSES	MSG093				
IKRCHECK	CAUSES	MSG146	MSG148	MSG152	MSG153	MSG154
IKRCKASS	CAUSES	MSG071	MSG078			
IKRCKCMP	CAUSES	MSG083	MSG109	MSG110		
IKRCKCTL	CAUSES	MSG394				
IKRCKD	CAUSES	MSG091				
IKRCKFIX	CAUSES	MSG133	MSG134			
IKRCKLV2	CAUSES	MSG014	MSG312	MSG313	MSG321	MSG323
		MSG327	MSG329	MSG330	MSG331	MSG332
		MSG333	MSG334	MSG336	MSG337	MSG338
		MSG339	MSG340	MSG341	MSG342	MSG343
		MSG346	MSG349	MSG351	MSG358	MSG380
		MSG381	MSG391	MSG392	MSG393	

IKRCLMSH	CAUSES	MSG081				
IKRCPIN	CAUSES	MSG068	MSG092	MSG095	MSG131	
IKRDEQ	CAUSES	MSG081				
IKRDIAGN	CAUSES	MSG160	MSG161	MSG162		
IKRDLIN	CAUSES	MSG100				
IKREAPL	CAUSES	MSG014	MSG030	MSG041	MSG326	
IKREARC	CAUSES	MSG014	MSG027	MSG041	MSG321	MSG322
		MSG324	MSG325	MSG326	MSG359	MSG392
		MSG395				
IKREBCK	CAUSES	MSG014	MSG023	MSG041		
IKRECOR	CAUSES	MSG014	MSG041	MSG063	MSG067	MSG083
		MSG100	MSG103	MSG104	MSG105	MSG106
		MSG107	MSG124	MSG125	MSG126	MSG130
		MSG131	MSG137	MSG142	MSG143	MSG178
		MSG179	MSG180	MSG211	MSG217	MSG272
		MSG326				
IKRECPY	CAUSES	MSG014	MSG025	MSG041		
IKRECRE	CAUSES	MSG014	MSG021	MSG041		
IKREDMP	CAUSES	MSG014	MSG029	MSG041		
IKREINC	CAUSES	MSG003	MSG014	MSG031	MSG041	MSG077
		MSG225	MSG312	MSG326		
IKRELKP	CAUSES	MSG014	MSG033	MSG041	MSG312	
IKRELST	CAUSES	MSG014	MSG034	MSG041	MSG063	MSG069
IKREMBDY	CAUSES	MSG002	MSG083	MSG111	MSG113	MSG123
		MSG128	MSG131	MSG275	MSG276	
IKREMRG	CAUSES	MSG014	MSG026	MSG041		
IKREMU	CAUSES	MSG0003	MSG097			
IKREPER	CAUSES	MSG014	MSG022	MSG041		
IKRERDC	CAUSES	MSG003	MSG014	MSG036	MSG041	
IKRERMV	CAUSES	MSG014	MSG028	MSG041	MSG322	
IKRERST	CAUSES	MSG014	MSG024	MSG041	MSG345	MSG347
IKRERTC	CAUSES	MSG014	MSG039	MSG041	MSG460	
IKRERVK	CAUSES	MSG014	MSG032	MSG041	MSG326	
IKRESEL	CAUSES	MSG014	MSG035	MSG042	MSG326	MSG396
IKRESTL	CAUSES	MSG003	MSG014	MSG038	MSG041	MSG063
		MSG224	MSG312	MSG345	MSG347	MSG348
IKRETLR	CAUSES	MSG014	MSG037	MSG041	MSG392	MSG395
IKREUND	CAUSES	MSG014	MSG041	MSG083		
IKREXIT	CAUSES	MSG041	MSG045			
IKRFBA	CAUSES	MSG091				
IKRGTBUF	CAUSES	MSG063				
IKRGTINF	CAUSES	MSG081				
IKRHDLV2	CAUSES	MSG186	MSG328	MSG353	MSG357	MSG386
		MSG387	MSG389			
IKRHELP	CAUSES	MSG017				
IKRINSTL	CAUSES	MSG083	MSG089	MSG131	MSG222	MSG223
		MSG226	MSG227	MSG228	MSG229	MSG230
		MSG231	MSG232	MSG233	MSG234	MSG235
		MSG236	MSG238	MSG239	MSG240	MSG248
		MSG249	MSG256			
IKRISLNK	CAUSES	MSG256	MSG272	MSG277	MSG278	

IKRISVBK	CAUSES	MSG083	MSG089	MSG131	MSG259	MSG268
		MSG269	MSG270	MSG271	MSG275	MSG276
		MSG362	MSG363	MSG365	MSG366	MSG416
		MSG368	MSG369	MSG370	MSG416	
IKRLGNRC	CAUSES	MSG081				
IKRLIBST	CAUSES	MSG063	MSG212			
IKRLINCM	CAUSES	MSG081				
IKRLKUP	CAUSES	MSG100				
IKRLOAD	CAUSES	MSG060				
IKRLOFI	CAUSES	MSG081				
IKRMAIN	CAUSES	MSG001	MSG003	MSG004	MSG005	MSG006
		MSG009	MSG061	MSG063	MSG081	MSG083
IKRMRGIN	CAUSES	MSG300	MSG301			
IKROPEN	CAUSES	MSG081	MSG092			
IKROPHF	CAUSES	MSG083	MSG090	MSG091	MSG094	
IKROPLNK	CAUSES	MSG076	MSG078	MSG081		
IKROPMSH	CAUSES	MSG081				
IKRPARSE	CAUSES	MSG007	MSG008	MSG010	MSG016	MSG080
IKRPDZAP	CAUSES	MSG014	MSG083	MSG087	MSG089	MSG108
		MSG132	MSG133	MSG134	MSG135	MSG136
		MSG138	MSG130	MSG140	MSG144	MSG145
		MSG147	MSG165	MSG170	MSG183	MSG326
		MSG338	MSG356			
IKRPERS	CAUSES	MSG311				
IKRPRAPL	CAUSES	MSG371	MSG403	MSG405	MSG406	
		MSG407	MSG408	MSG409	MSG410	MSG411
		MSG412	MSG413	MSG414	MSG417	
IKRQUEUE	CAUSES	MSG081				
IKRRDCE	CAUSES	MSG100	MSG101	MSG112		
IKRRDLV2	CAUSES	MSG004	MSG011	MSG012	MSG015	MSG083
IKRRDNXT	CAUSES	MSG013				
IKRREGCK	CAUSES	MSG081	MSG424	MSG425		
IKRREQFR	CAUSES	MSG067				
IKRRLZAP	CAUSES	MSG014	MSG108	MSG131	MSG132	MSG135
		MSG136	MSG138	MSG139	MSG140	MSG141
		MSG156	MSG165	MSG170	MSG220	MSG326
IKRRMGEN	CAUSES	MSG081				
IKRRTRVE	CAUSES	MSG066	MSG081			
IKRSAASM	CAUSES	MSG081				
IKRSAMNT	CAUSES	MSG113	MSG123	MSG415		
IKRSAPL	CAUSES	MSG368	MSG369	MSG370	MSG427	
IKRSERR	CAUSES	MSG209				
IKRSET	CAUSES	MSG163				
IKRSLA	CAUSES	MSG064	MSG065	MSG081		
IKRSLZAP	CAUSES	MSG014	MSG071	MSG124	MSG132	MSG165
		MSG172	MSG174	MSG175	MSG177	MSG181
		MSG182	MSG186	MSG265		
IKRSREPQ	CAUSES	MSG063				
IKRSTMTS	CAUSES	MSG403	MSG415	MSG416	MSG417	
IKRSVCMD	CAUSES	MSG081	MSG084			

IKRTAIL	CAUSES	MSG045	MSG067	MSG083	MSG088	MSG100
		MSG131	MSG210	MSG212	MSG217	MSG397
		MSG420	MSG421	MSG422	MSG423	MSG428
		MSG429	MSG430	MSG431		
IKRTPMSH	CAUSES	MSG247				
IKRUJRP	CAUSES	MSG066				
IKRUNDO	CAUSES	MSG100	MSG124	MSG125	MSG126	MSG127
		MSG131	MSG176	MSG211	MSG217	MSG272
IKRUPGR	CAUSES	MSG083	MSG088	MSG089	MSG111	MSG116
		MSG117	MSG118	MSG119	MSG120	MSG121
		MSG129	MSG201	MSG202	MSG204	MSG205
		MSG206	MSG207	MSG208	MSG251	MSG255
		MSG258	MSG262	MSG263	MSG264	MSG265
		MSG273	MSG274	MSG279	MSG360	MSG367
		MSG426				
IKRWRSQ	CAUSES	MSG095				

Part 2: Sorted by Messages

MSG001	IS CAUSED BY	IKRMAIN				
MSG002	IS CAUSED BY	IKREMBDY				
MSG003	IS CAUSED BY	IKREINC	IKREMU	IKRERDC	IKRESTL	IKRMAIN
MSG004	IS CAUSED BY	IKRMAIN	IKRRDLV2			
MSG005	IS CAUSED BY	IKRMAIN				
MSG006	IS CAUSED BY	IKRMAIN				
MSG007	IS CAUSED BY	IKRPARSE				
MSG008	IS CAUSED BY	IKRPARSE				
MSG009	IS CAUSED BY	IKRMAIN				
MSG010	IS CAUSED BY	IKRPARSE				
MSG011	IS CAUSED BY	IKRRDLV2				
MSG012	IS CAUSED BY	IKRRDLV2				
MSG013	IS CAUSED BY	IKRRDNXT				
MSG014	IS CAUSED BY	IKRCKLV2	IKREAPL	IKREARC	IKREBCK	
		IKRECOR	IKRECPY	IKRECRE	IKREDMP	
		IKREINC	IKRELKP	IKRELST	IKREMRG	
		IKREPER	IKRERDC	IKRERMV	IKRERST	
		IKRERTC	IKRERVK	IKRESEL	IKRESTL	
		IKRETLR	IKREUND	IKRPDZAP	IKRRLZAP	
		IKRSLZAP				
MSG015	IS CAUSED BY	IKRRDLV2				
MSG016	IS CAUSED BY	IKRPARSE				
MSG017	IS CAUSED BY	IKRHELP				
MSG021	IS CAUSED BY	IKRECRE				
MSG022	IS CAUSED BY	IKREPER				
MSG023	IS CAUSED BY	IKREBCK				
MSG024	IS CAUSED BY	IKRERST				
MSG025	IS CAUSED BY	IKRECPY				
MSG026	IS CAUSED BY	IKREMRG				
MSG027	IS CAUSED BY	IKREARC				
MSG028	IS CAUSED BY	IKRERMV				
MSG029	IS CAUSED BY	IKREDMP				
MSG030	IS CAUSED BY	IKREAPL				
MSG031	IS CAUSED BY	IKREINC				
MSG032	IS CAUSED BY	IKRERVK				
MSG033	IS CAUSED BY	IKRELKP				
MSG034	IS CAUSED BY	IKRELST				
MSG035	IS CAUSED BY	IKRESEL				
MSG036	IS CAUSED BY	IKRERDC				
MSG037	IS CAUSED BY	IKRETLR				
MSG038	IS CAUSED BY	IKRESTL				
MSG039	IS CAUSED BY	IKRERTC				
MSG041	IS CAUSED BY	IKREAPL	IKREARC	IKREBCK	IKRECOR	
		IKRECPY	IKRECRE	IKREDMP	IKREINC	
		IKRELKP	IKRELST	IKREMRG	IKREPER	
		IKRERDC	IKRERMV	IKRERST	IKRERTC	
		IKRERVK	IKRESTL	IKRETLR	IKREUND	
		IKREXIT				
MSG042	IS CAUSED BY	IKRESEL				
MSG045	IS CAUSED BY	IKREXIT	IKRTAIL			
MSG060	IS CAUSED BY	IKRLOAD				
MSG061	IS CAUSED BY	IKRBKUP	IKRMAIN			
MSG063	IS CAUSED BY	IKRECOR	IKRELST	IKRESTL	IKRGTBUF	

		IKRLIBST	IKRMAIN	IKRSREPQ	
MSG064	IS CAUSED BY	IKRSLA			
MSG065	IS CAUSED BY	IKRSLA			
MSG066	IS CAUSED BY	IKRBKPR	IKRRTRVE	IKRUIRP	
MSG067	IS CAUSED BY	IKRAGEN	IKRECOR	IKRREQFR	IKRTAIL
MSG068	IS CAUSED BY	IKRCPIN			
MSG069	IS CAUSED BY	IKRELST			
MSG071	IS CAUSED BY	IKRCKASS	IKRSLZAP		
MSG076	IS CAUSED BY	IKROPLNK			
MSG077	IS CAUSED BY	IKREINC			
MSG078	IS CAUSED BY	IKRCKASS	IKROPLNK		
MSG080	IS CAUSED BY	IKRPARSE			
MSG081	IS CAUSED BY	IKRBKPR	IKRBKUP	IKRCLMSH	IKRDEQ
		IKRGTINF	IKRLGNRC	IKRLINCM	IKRLOFI
		IKRMAIN	IKROPEN	IKROPLNK	IKROPMSH
		IKRQUEUE	IKRREGCK	IKRRMGEN	IKRRTRVE
		IKRSAASM	IKRSLA	IKRSVCMD	
MSG083	IS CAUSED BY	IKRBKPR	IKRBKUP	IKRCKCMP	IKRECOR
		IKREUND	IKRINSTL	IKRISVBK	IKRMAIN
		IKROPHF	IKRPDZAP	IKRRDLV2	IKRTAIL
					IKRUPGR
MSG084	IS CAUSED BY	IKRSVCMD			
MSG087	IS CAUSED BY	IKRPDZAP			
MSG088	IS CAUSED BY	IKRTAIL	IKRUPGR		
MSG089	IS CAUSED BY	IKRBKPR	IKRINSTL	IKRPDZAP	IKRISVBK
MSG090	IS CAUSED BY	IKROPHF			
MSG091	IS CAUSED BY	IKRCKD	IKRFBA	IKROPHF	
MSG092	IS CAUSED BY	IKRCPIN	IKROPEN		
MSG093	IS CAUSED BY	IKRCGT			
MSG094	IS CAUSED BY	IKROPHF			
MSG095	IS CAUSED BY	IKRCPIN	IKRWRSQ		
MSG097	IS CAUSED BY	IKREMU			
MSG100	IS CAUSED BY	IKRAAPAR	IKRAGEN	IKRAPPLY	IKRAPTF
		IKRBKPR	IKRDLIN	IKRECOR	IKRLKUP
		IKRRDCE	IKRTAIL	IKRUNDO	
MSG101	IS CAUSED BY	IKRBKPR	IKRRDCE		
MSG102	IS CAUSED BY	IKRBKPR			
MSG103	IS CAUSED BY	IKRAAPAR	IKRACOMP	IKRAPRD	IKRAPTF
		IKRECOR			
MSG104	IS CAUSED BY	IKRECOR			
MSG105	IS CAUSED BY	IKRECOR			
MSG106	IS CAUSED BY	IKRECOR			
MSG107	IS CAUSED BY	IKRECOR			
MSG108	IS CAUSED BY	IKRPDZAP	IKRRLZAP		
MSG109	IS CAUSED BY	IKRCKCMP			
MSG110	IS CAUSED BY	IKRCKCMP			
MSG111	IS CAUSED BY	IKREMBDY	IKRUPGR		
MSG112	IS CAUSED BY	IKRBKPR	IKRRDCE		
MSG113	IS CAUSED BY	IKREMBDY	IKRSAMNT		
MSG114	IS CAUSED BY	IKRAPRD			
MSG116	IS CAUSED BY	IKRUPGR			
MSG117	IS CAUSED BY	IKRUPGR			
MSG118	IS CAUSED BY	IKRUPGR			
MSG119	IS CAUSED BY	IKRUPGR			
MSG120	IS CAUSED BY	IKRUPGR			

MSG121	IS	CAUSED	BY	IKRAPPLY	IKRUPGR		
MSG123	IS	CAUSED	BY	IKREMBDY	IKRSAMNT		
MSG124	IS	CAUSED	BY	IKRECOR	IKRSLZAP	IKRUNDO	
MSG125	IS	CAUSED	BY	IKRECOR	IKRUNDO		
MSG126	IS	CAUSED	BY	IKRECOR	IKRUNDO		
MSG127	IS	CAUSED	BY	IKRUNDO			
MSG128	IS	CAUSED	BY	IKREMBDY			
MSG129	IS	CAUSED	BY	IKRUPGR			
MSG130	IS	CAUSED	BY	IKRECOR			
MSG131	IS	CAUSED	BY	IKRKBPR	IKRCPIN	IKRECOR	IKREMBDY IKRINSTL
				IKRISVBK	IKRRLZAP	IKRTAIL	IKRUNDO
MSG132	IS	CAUSED	BY	IKRPDZAP	IKRRLZAP	IKRSLZAP	
MSG133	IS	CAUSED	BY	IKRCKFIX	IKRPDZAP		
MSG134	IS	CAUSED	BY	IKRCKFIX	IKRPDZAP		
MSG135	IS	CAUSED	BY	IKRPDZAP	IKRRLZAP		
MSG136	IS	CAUSED	BY	IKRPDZAP	IKRRLZAP		
MSG137	IS	CAUSED	BY	IKRECOR			
MSG138	IS	CAUSED	BY	IKRPDZAP	IKRRLZAP		
MSG139	IS	CAUSED	BY	IKRPDZAP	IKRRLZAP		
MSG140	IS	CAUSED	BY	IKRPDZAP	IKRRLZAP		
MSG141	IS	CAUSED	BY	IKRRLZAP			
MSG142	IS	CAUSED	BY	IKRECOR			
MSG143	IS	CAUSED	BY	IKRECOR			
MSG144	IS	CAUSED	BY	IKRPDZAP			
MSG145	IS	CAUSED	BY	IKRPDZAP			
MSG146	IS	CAUSED	BY	IKRCHECK			
MSG147	IS	CAUSED	BY	IKRPDZAP			
MSG148	IS	CAUSED	BY	IKRCHECK			
MSG152	IS	CAUSED	BY	IKRCHECK			
MSG153	IS	CAUSED	BY	IKRCHECK			
MSG154	IS	CAUSED	BY	IKRCHECK			
MSG156	IS	CAUSED	BY	IKRRLZAP			
MSG160	IS	CAUSED	BY	IKRDIAGN			
MSG161	IS	CAUSED	BY	IKRDIAGN			
MSG162	IS	CAUSED	BY	IKRDIAGN			
MSG163	IS	CAUSED	BY	IKRSET			
MSG165	IS	CAUSED	BY	IKRPDZAP	IKRRLZAP	IKRSLZAP	
MSG170	IS	CAUSED	BY	IKRPDZAP	IKRRLZAP		
MSG172	IS	CAUSED	BY	IKRSLZAP			
MSG174	IS	CAUSED	BY	IKRSLZAP			
MSG175	IS	CAUSED	BY	IKRSLZAP			
MSG176	IS	CAUSED	BY	IKRUNDO			
MSG177	IS	CAUSED	BY	IKRSLZAP			
MSG178	IS	CAUSED	BY	IKRECOR			
MSG179	IS	CAUSED	BY	IKRECOR			
MSG180	IS	CAUSED	BY	IKRECOR			
MSG181	IS	CAUSED	BY	IKRSLZAP			
MSG182	IS	CAUSED	BY	IKRSLZAP			
MSG183	IS	CAUSED	BY	IKRPDZAP			
MSG186	IS	CAUSED	BY	IKRHDLV2	IKRSLZAP		
MSG201	IS	CAUSED	BY	IKRUPGR			
MSG202	IS	CAUSED	BY	IKRUPGR			
MSG204	IS	CAUSED	BY	IKRUPGR			
MSG205	IS	CAUSED	BY	IKRUPGR			

MSG206	IS CAUSED BY	IKRUPGR		
MSG207	IS CAUSED BY	IKRUPGR		
MSG208	IS CAUSED BY	IKRUPGR		
MSG209	IS CAUSED BY	IKRSERR		
MSG210	IS CAUSED BY	IKRTAIL		
MSG211	IS CAUSED BY	IKRECOR	IKRUNDO	
MSG212	IS CAUSED BY	IKRLIBST	IKRTAIL	
MSG217	IS CAUSED BY	IKRECOR	IKRTAIL	IKRUNDO
MSG220	IS CAUSED BY	IKRRLZAP		
MSG222	IS CAUSED BY	IKRINSTL		
MSG223	IS CAUSED BY	IKRINSTL		
MSG224	IS CAUSED BY	IKRESTL		
MSG225	IS CAUSED BY	IKREINC		
MSG226	IS CAUSED BY	IKRINSTL		
MSG227	IS CAUSED BY	IKRINSTL		
MSG228	IS CAUSED BY	IKRINSTL		
MSG229	IS CAUSED BY	IKRINSTL		
MSG230	IS CAUSED BY	IKRINSTL		
MSG231	IS CAUSED BY	IKRINSTL		
MSG232	IS CAUSED BY	IKRINSTL		
MSG233	IS CAUSED BY	IKRINSTL		
MSG234	IS CAUSED BY	IKRINSTL		
MSG235	IS CAUSED BY	IKRINSTL		
MSG236	IS CAUSED BY	IKRINSTL		
MSG238	IS CAUSED BY	IKRINSTL		
MSG239	IS CAUSED BY	IKRINSTL		
MSG240	IS CAUSED BY	IKRINSTL		
MSG241	IS CAUSED BY	IKRBKPR		
MSG242	IS CAUSED BY	IKRBKPR		
MSG243	IS CAUSED BY	IKRBKPR		
MSG244	IS CAUSED BY	IKRBKPR		
MSG246	IS CAUSED BY	IKRBKPR		
MSG247	IS CAUSED BY	IKRTPMSH		
MSG248	IS CAUSED BY	IKRINSTL		
MSG249	IS CAUSED BY	IKRINSTL		
MSG250	IS CAUSED BY	IKRBKPR		
MSG251	IS CAUSED BY	IKRUPGR		
MSG255	IS CAUSED BY	IKRUPGR		
MSG256	IS CAUSED BY	IKRINSTL	IKRISLNK	
MSG258	IS CAUSED BY	IKRUPGR		
MSG259	IS CAUSED BY	IKRISVBK		
MSG262	IS CAUSED BY	IKRUPGR		
MSG263	IS CAUSED BY	IKRUPGR		
MSG264	IS CAUSED BY	IKRUPGR		
MSG265	IS CAUSED BY	IKRSLZAP	IKRUPGR	
MSG268	IS CAUSED BY	IKRISVBK		
MSG269	IS CAUSED BY	IKRISVBK		
MSG270	IS CAUSED BY	IKRISVBK		
MSG271	IS CAUSED BY	IKRISVBK		
MSG272	IS CAUSED BY	IKRECOR	IKRISLNK	IKRUNDO
MSG273	IS CAUSED BY	IKRUPGR		
MSG274	IS CAUSED BY	IKRUPGR		
MSG275	IS CAUSED BY	IKRAPPLY	IKREMBDY	IKRISVBK
MSG276	IS CAUSED BY	IKRAPPLY	IKREMBDY	IKRISVBK

MSG277	IS	CAUSED	BY	IKRISLNK			
MSG278	IS	CAUSED	BY	IKRISLNK			
MSG279	IS	CAUSED	BY	IKRUPGR			
MSG300	IS	CAUSED	BY	IKRMRGIN			
MSG301	IS	CAUSED	BY	IKRMRGIN			
MSG306	IS	CAUSED	BY	IKRBKPR			
MSG307	IS	CAUSED	BY	IKRBKPR			
MSG311	IS	CAUSED	BY	IKRPERS			
MSG312	IS	CAUSED	BY	IKRCKLV2	IKREINC	IKRELKP	IKRESTL
MSG313	IS	CAUSED	BY	IKRCKLV2			
MSG321	IS	CAUSED	BY	IKRCKLV2	IKREARC		
MSG322	IS	CAUSED	BY	IKREARC	IKRERMV		
MSG323	IS	CAUSED	BY	IKRCKLV2			
MSG324	IS	CAUSED	BY	IKREARC			
MSG325	IS	CAUSED	BY	IKREARC			
MSG326	IS	CAUSED	BY	IKREAPL	IKREARC	IKRECOR	IKREINC
				IKRERVK	IKRESEL	IKRPDZAP	IKRRLZAP
MSG327	IS	CAUSED	BY	IKRCKLV2			
MSG328	IS	CAUSED	BY	IKRHDLV2			
MSG329	IS	CAUSED	BY	IKRCKLV2			
MSG330	IS	CAUSED	BY	IKRCKLV2			
MSG331	IS	CAUSED	BY	IKRCKLV2			
MSG332	IS	CAUSED	BY	IKRCKLV2			
MSG333	IS	CAUSED	BY	IKRCKLV2			
MSG334	IS	CAUSED	BY	IKRCKLV2			
MSG336	IS	CAUSED	BY	IKRCKLV2			
MSG337	IS	CAUSED	BY	IKRCKLV2			
MSG338	IS	CAUSED	BY	IKRCKLV2	IKRPDZAP		
MSG339	IS	CAUSED	BY	IKRCKLV2			
MSG340	IS	CAUSED	BY	IKRCKLV2			
MSG341	IS	CAUSED	BY	IKRCKLV2			
MSG342	IS	CAUSED	BY	IKRCKLV2			
MSG343	IS	CAUSED	BY	IKRCKLV2			
MSG345	IS	CAUSED	BY	IKRERST	IKRESTL		
MSG346	IS	CAUSED	BY	IKRCKLV2			
MSG347	IS	CAUSED	BY	IKRERST	IKRESTL		
MSG348	IS	CAUSED	BY	IKRESTL			
MSG349	IS	CAUSED	BY	IKRCKLV2			
MSG351	IS	CAUSED	BY	IKRCKLV2			
MSG353	IS	CAUSED	BY	IKRHDLV2			
MSG356	IS	CAUSED	BY	IKRPDZAP			
MSG357	IS	CAUSED	BY	IKRHDLV2			
MSG358	IS	CAUSED	BY	IKRCKLV2			
MSG359	IS	CAUSED	BY	IKREARC			
MSG360	IS	CAUSED	BY	IKRUPGR			
MSG362	IS	CAUSED	BY	IKRISVBK			
MSG363	IS	CAUSED	BY	IKRISVBK			
MSG365	IS	CAUSED	BY	IKRISVBK			
MSG366	IS	CAUSED	BY	IKRISVBK			
MSG367	IS	CAUSED	BY	IKRUPGR			
MSG368	IS	CAUSED	BY	IKRISVBK	IKRSAPL		
MSG369	IS	CAUSED	BY	IKRISVBK	IKRSAPL		
MSG370	IS	CAUSED	BY	IKRISVBK	IKRSAPL		
MSG371	IS	CAUSED	BY	IKRPRAPL			

MSG380	IS CAUSED BY	IKRCKLV2	
MSG381	IS CAUSED BY	IKRCKLV2	
MSG386	IS CAUSED BY	IKRHDLV2	
MSG387	IS CAUSED BY	IKRHDLV2	
MSG389	IS CAUSED BY	IKRHDLV2	
MSG391	IS CAUSED BY	IKRCKLV2	
MSG392	IS CAUSED BY	IKRCKLV2	IKREARC IKRETLR
MSG393	IS CAUSED BY	IKRCKLV2	
MSG394	IS CAUSED BY	IKRCKCTL	
MSG395	IS CAUSED BY	IKREARC	IKRETLR
MSG396	IS CAUSED BY	IKRESEL	
MSG397	IS CAUSED BY	IKRTAIL	
MSG403	IS CAUSED BY	IKRPRAPL	IKRSTMTS
MSG405	IS CAUSED BY	IKRPRAPL	
MSG406	IS CAUSED BY	IKRPRAPL	
MSG407	IS CAUSED BY	IKRPRAPL	
MSG408	IS CAUSED BY	IKRPRAPL	
MSG409	IS CAUSED BY	IKRPRAPL	
MSG410	IS CAUSED BY	IKRPRAPL	
MSG411	IS CAUSED BY	IKRPRAPL	
MSG412	IS CAUSED BY	IKRPRAPL	
MSG413	IS CAUSED BY	IKRPRAPL	
MSG414	IS CAUSED BY	IKRPRAPL	
MSG415	IS CAUSED BY	IKRSAMNT	IKRSTMTS
MSG416	IS CAUSED BY	IKRISVBK	IKRPRAPL IKRSTMTS
MSG417	IS CAUSED BY	IKRPRAPL	IKRSTMTS
MSG419	IS CAUSED BY	IKRAGEN	
MSG420	IS CAUSED BY	IKRAGEN	IKRTAIL
MSG421	IS CAUSED BY	IKRTAIL	
MSG422	IS CAUSED BY	IKRTAIL	
MSG423	IS CAUSED BY	IKRTAIL	
MSG424	IS CAUSED BY	IKRREGCK	
MSG425	IS CAUSED BY	IKRREGCK	
MSG426	IS CAUSED BY	IKRUPGR	
MSG427	IS CAUSED BY	IKRSAPL	
MSG428	IS CAUSED BY	IKRTAIL	
MSG429	IS CAUSED BY	IKRTAIL	
MSG430	IS CAUSED BY	IKRTAIL	
MSG431	IS CAUSED BY	IKRTAIL	
MSG460	IS CAUSED BY	IKRERTC	

History File Dump Analysis Example

This section, together with Figure 8 on page 13, shows the major areas of history file records and the relationship between different record types.

Figure 16 on page 418 is a sample dump of a history file produced by the DUMP function of MSHP. Figure 17 on page 432 shows the structure of the dumped file; to demonstrate the relationship between the records, only those pointer locations and pointer values that are used to form the structure are shown. This figure is followed by a description of the records.

In Figure 17 on page 432 and the description of the records, all displacements within the records are in decimal, all RBAs (relative byte addresses) are hexadecimal.

For more information on record types, identifiers, chains, and DSECT names, refer to "History File Records" in Chapter 4; for information on the structure of the history file, refer to Figure 8 on page 13 in Chapter 2.

DATE 12/12/83 (MM/DD/YY) JOB EXAMPLE MSHP-FUNCTION = DUMP

```
RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0041 000A00 CHAR V.. ..E.DISP .E.SGNUM .E.SGCCWT .E.SGCCWF .E.ISTAVT .
ZONE E00400C00C4CCED4440C4ECDEC4440C4ECCCEE440C4ECCCEC440C4CEECEE440
NUMR 50A0090515B4927000015B2754300015B2733630015B2733660015B923153000

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0042 000A40 CHAR V.....E.SGPMR .E.SGPREAL .E.SMICR .E.SGAP .E.SGEFCH .
ZONE E00800000C4ECDD4440C4ECDDCC40C4EDCCD4440C4ECCD44440C4ECCCC440
NUMR 50A00A0515B2774900015B2779513015B2493900015B2717000015B275638000

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0043 000A80 CHAR V.....E.ASYCODE .E.ASYTAB .E.SGATAB .E.SGPK .E.SGAFCH .
ZONE E00C00400C4CEECDCC40C4CEECC440C4ECCECC440C4ECDCD4440C4ECCCC440
NUMR 50A00A0515B1283645015B1283120015B2713120015B2773200015B271638000

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0044 000AC0 CHAR V.....E.SGDFCH .E.SGSVC .E.SGSVCX .E.MCRAS .E.SGSTAR .
ZONE E00000800C4ECCCC440C4EEEC4440C4EEECE440C4DCDCE4440C4EEECD440
NUMR 50B00A0515B2746380015B2725300015B2725370015B4391200015B272319000

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0045 000B00 CHAR V.. ..E.SGIOS .E.SGCFCH .E.SGERP .E.SGTINF .E.DTSSVIC.
ZONE E00400C00C4ECCDE4440C4ECCCC440C4ECCDD4440C4ECECDC440C4EEEEECCC0
NUMR 50B00A0515B2796200015B2736380015B2759700015B2739560015B432253930

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0046 000B40 CHAR V.....E.DTSSVCIN.E.SGRM .E.SCBATAB .E.SGLOCK .E.SGAM .
ZONE E00800000C4EEEEECCD0C4ECDD44440C4ECCCECC40C4ECDDCD440C4ECCD44440
NUMR 50B00B0515B4322539515B2794000015B2321312015B2736320015B271400000

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0047 000B80 CHAR V.....E.SGNPGR .E.SGBFCH .E.SGSER .E.SGACF .E.SGXECB .
ZONE E00C00400C4ECDDCD440C4ECCCC440C4ECECD4440C4ECCCC4440C4EECCC440
NUMR 50B00B0515B2757790015B2726380015B2725900015B2713600015B277532000

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0048 000BC0 CHAR V.....E.SGACCT .E.SGINF .E.SGIUCV .E.SGPDATA .E.SGDSECTF.
ZONE E00000800C4ECCCE440C4ECCDC4440C4ECECE440C4ECDCCE40C4EECCCEC0
NUMR 50C00B0515B2713330015B2795600015B2794350015B2774131015B274253360

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0049 000C00 CHAR V.. ..E.MAPPUB .E.MAPCHNTB.E.EXTENT .E.EXNTBLK .E.AVRLIST .
ZONE E00400C00C4DCDDEC440C4DCDCDEC0C4CEECDE440C4CEDECDD40C4CEDDCEE40
NUMR 50C00B0515B4177420015B4173853215B5735530015B5753232015B159392300

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0050 000C40 CHAR V.....E.SGCHQ .E.ICM$COM .E.MAPDIBF .E.MAPPUBX .E.MAPERPIB.
ZONE E00800000C4ECCCD4440C4CCD5CDD40C4DCDCCC40C4DCDDECE40C4DCDDCC0
NUMR 50C00C0515B2738800015B934B364015B4174926015B4177427015B417597920
```

Figure 16 (Part 5 of 14). History File Sample Dump

DATE 12/12/83 (MM/DD/YY) JOB EXAMPLE MSHP-FUNCTION = DUMP

```
RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0051 000C80 CHAR V......E.MAPSAVAR.E.MAPSCB .E.MAPPCB .E.MAPTIB .E.MAPTCB .
ZONE E00C00400C4DCDECECD0C4DCDECC440C4DCDDCC440C4DCDECC440C4DCDECC440
NUMR 50C00C0515B4172151915B4172320015B4177320015B4173920015B417332000

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0052 000CC0 CHAR V......E.MAPJATB .E.MAPPUB2 .E.MAPCBF .E.IPW$DPA .E.IPW$DPD .
ZONE E00000800C4DCDDCEC40C4DCDDECF40C4DCDCCC440C4CDE5CDC40C4CDE5CDC40
NUMR 50D00C0515B4171132015B4177422015B4173260015B976B471015B976B47400

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0053 000D00 CHAR V.. ....E.MAPCOMR .E.ORE .E.MAPBDY .E.MAPBDYVR.E.DTSVECTB.
ZONE E00400C00C4DCDCDD40C4DDC444440C4DCDCCE440C4DCDCCEED0C4CEEECECE0
NUMR 50D00C0515B4173649015B6950000015B4172480015B4172485915B432553320

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0054 000D40 CHAR V......E.SCYVECTB.E.DTSAPL .E.MAPDEVTY.E.SDAGBL .E.SGPDSECT.
ZONE E00800000C4ECECECE0C4CEECDD440C4DCDCCEE0C4ECCCCD440C4ECDCECE0
NUMR 50D00D0515B2385533215B4321730015B417453815B2417230015B277425330

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0055 000D80 CHAR V......E.SMDATA .E.NPGRST .E.SGHFCH .E.SGSRVLVL.E.MODFLD .
ZONE E00C00400C4EDCCEC440C4DDCDDEE40C4ECCCC440C4ECEDEDED0C4DDCCDC440
NUMR 50D00D0515B2441310015B5779323015B2786380015B2729535315B464634000

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0056 000DC0 CHAR V......E.SUPAVT .E.SGSETUP .E.CRTGEN .E.IPTE .E.SGPSVC .
ZONE E00000800C4EEDCEE440C4ECECEED40C4CDECCD440C4CDEC44440C4ECDEEC440
NUMR 50E00D0515B2471530015B2725347015B3937550015B9735000015B277253000

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0057 000E00 CHAR V.. ....E.SGPLLEV .E.SGPFIX .E.SGPOPT .E.CRSECT .E.SGETVIS .
ZONE E00400C00C4ECDDCE40C4ECDCE440C4ECDDDE440C4CDECCE440C4ECECECE40
NUMR 50E00D0515B2773355015B2776970015B2776730015B3925330015B275359200

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0058 000E40 CHAR V......E.SFREEVIS.E.CRSPID .E.EXCP .E.WAIT .E.SYSIO .
ZONE E00800000C4ECDCECE0C4CDEDCC440C4CED44440C4ECCE44440C4EEECD4440
NUMR 50E00E0515B2695559215B3927940015B5737000015B6193000015B282960000

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0059 000E80 CHAR V......E.CCB .E.CRSAV .E.SMCOM .E.SDAGDT .E.DTSMCIC .
ZONE E00C00400C4CC444440C4CDEECE440C4EDCD4440C4ECCCE440C4CEDCCC40
NUMR 50E00E0515B3320000015B3932150015B2436400015B2417430015B432439300

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0060 000EC0 CHAR V......E.DTL .E.SGLDUP .E.VIO .E.TSTOP .E.LOAD .
ZONE E00000800C4CED444440C4ECEDCED40C4ECD444440C4EEEDD4440C4DDCC44440
NUMR 50F00E0515B4330000015B2723447015B5960000015B3236700015B361400000
```

Figure 16 (Part 6 of 14). History File Sample Dump

DATE 12/12/83 (MM/DD/YY) JOB EXAMPLE MSHP-FUNCTION = DUMP

```
RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0061 000F00 CHAR V.. ....E.SGSCHED .E.IOINTER .E.SGDSK .E.SGSERI .E.MAPDLF .
ZONE E00400C00C4ECECCCC40C4CDCECD40C4ECCED4440C4ECECDC440C4DCDCDC440
NUMR 50F00E0515B2723854015B9695359015B2742200015B2725990015B417436000

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0062 000F40 CHAR V.....E.SGAMSUBR.E.GETFLD .E.TREADY .E.STXIT .E.EOJ .
ZONE E00800000C4ECCDEECD0C4CCECDC440C4EDCCCE440C4EEECE4440C4CDD444440
NUMR 50F00F0515B2714242915B7536340015B3951480015B2379300015B561000000

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0063 000F80 CHAR V.....E.EXIT .E.DSPLPAR .E.DSPLOG .E.IUCV .E.VSIUCVU .
ZONE E00C00400C4CECE44440C4CEDDDC40C4CEDDDC440C4CECE44440C4EEECEEE40
NUMR 50F00F0515B5793000015B4273719015B4273670015B9435000015B529435400

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0064 000FC0 CHAR V.....E.VSIUCVPL.E.IPARML .E.PMCOM .E.VIOCM .E.REMOVE .
ZONE E01000800C4EECECEDD0C4CDCDD440C4DDCDD4440C4ECDCD4440C4DCDDEC440
NUMR 50000F0515B5294357315B9719430015B7436400015B5963400015B954655000

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0065 001000 CHAR V.. ....E.INSBOT .E.DCTENTRY.E.MAPPFTE .E.MAPSTE .E.MAPPTE .
ZONE E01400C00C4CDECD440C4CCECDEDE0C4DCDCEC40C4DCDEEC440C4DCDDEC440
NUMR 50000F0515B9522630015B4335539815B4177635015B4172350015B417735000

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0066 001040 CHAR V.....E.MAPPTASE.E.MAPVIO .E.MAPDPD .E.MAPDEVCB.E.MAPGVCTL.
ZONE E01801000C4DCDDECEC0C4DCDECD440C4DCDCDC440C4DCDCECC0C4DCDCECED0
NUMR 5000000515B4177312515B4175960015B4174740015B4174553215B417753330

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0067 001080 CHAR V.....E.MAPANCH .E.MAPDNTRY.E.INLCDDTE.E.INLCEDTE.E.INLCDENT.
ZONE E01C01400C4DCDCDC40C4DCDCDEDE0C4CDDCCCEC0C4CDDCCCEC0C4CDDCCDE0
NUMR 5000000515B4171538015B4174539815B9533443515B9533543515B953345530

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0068 0010C0 CHAR V.....E.INLCLBCF.E.INLCLDES.E.INLCMBRX.E.INLCLDTE.E.INLCLOT .
ZONE E01001800C4CDDCDECC0C4CDDCCE0C4CDDCDE0C4CDDCDE0C4CDDCDE0C4CDDCDE40
NUMR 5010000515B9533323615B9533345215B9533429715B9533343515B953336300

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0069 001100 CHAR V.. ....E.INLCLPT .E.INLCSLTE.E.INLCSLDE.E.INLCSLXE.E.MAPSVAH.
ZONE E01401C00C4CDDCDE40C4CDDCECEC0C4CDDCEDCC0C4CDDCEDEC0C4DCDEECC0
NUMR 5010000515B9533373015B9533243515B9533234515B9533237515B417251840

RECORD RBA      0....*....1....*....2....*....3....*....4....*....5....*....6...
0070 001140 CHAR V.....E.SETIME .E.ASYSCOM .E.INLMLAMB.E.INLMLRPL.E.INLMSCON.
ZONE E01801000C4ECECDC440C4CEEECD40C4CDDDDC0C4CDDDDDD0C4CDDDECD0
NUMR 5010010515B2539450015B1282364015B9534314215B9534397315B953423650
```

Figure 16 (Part 7 of 14). History File Sample Dump

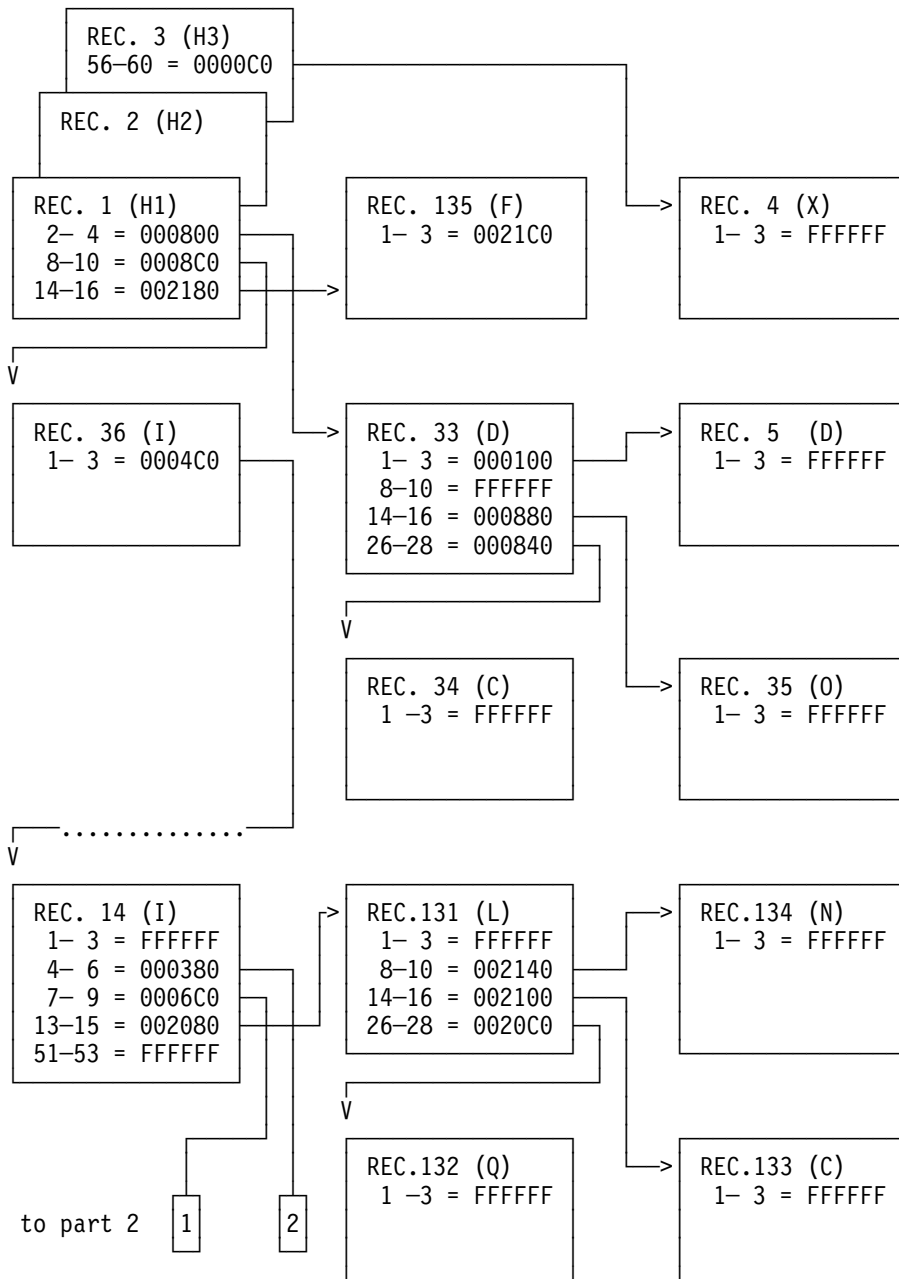


Figure 17 (Part 1 of 2). Sample Dump Queue Structure

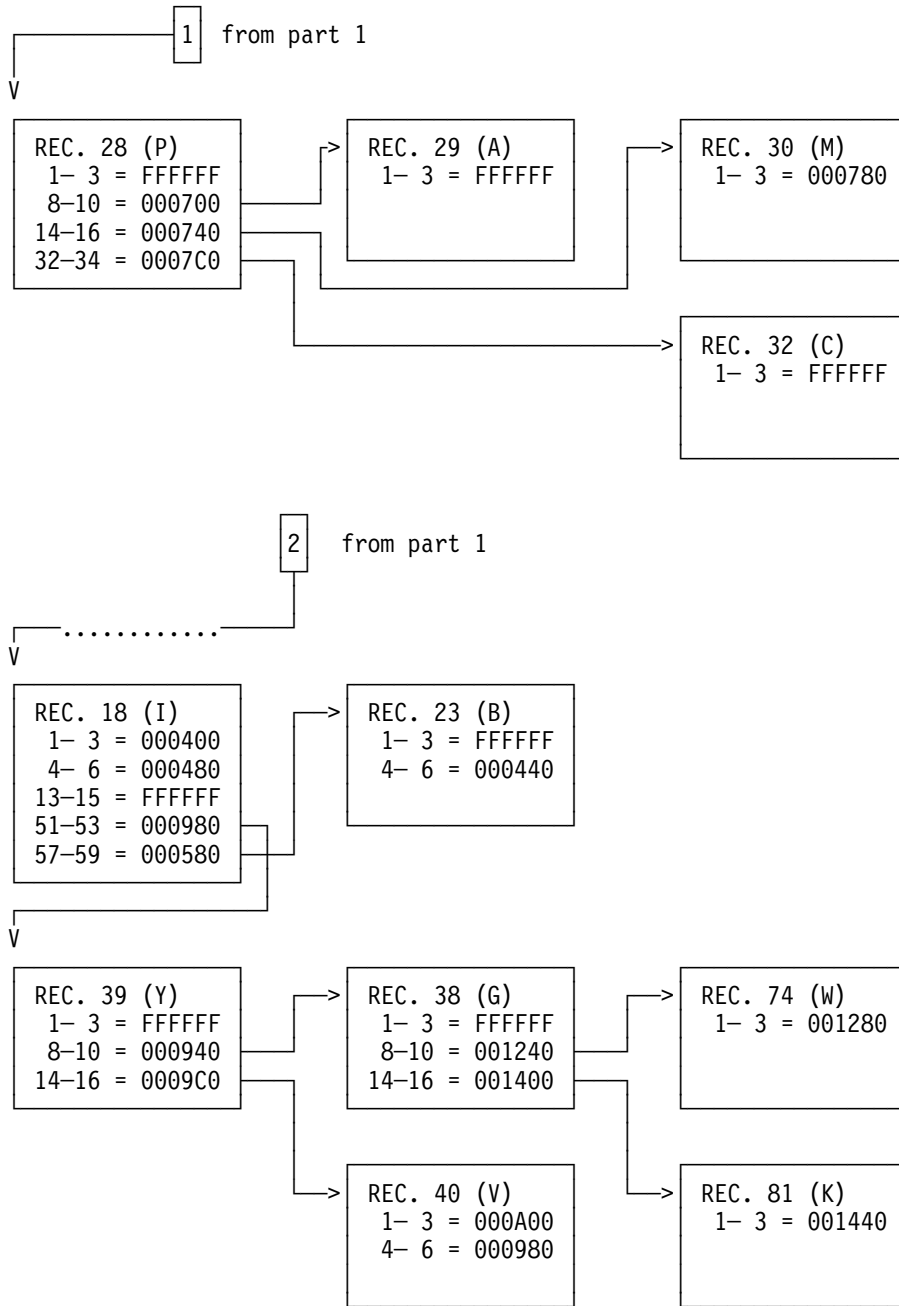


Figure 17 (Part 2 of 2). Sample Dump Queue Structure

Description of History File Records

The following is a description of history file records in the sequence of their logical queue structure, as shown in Figure 16 on page 418.

Record 1 (H1) HEADER 1 History File Record

This record (RBA=000000) in conjunction with records 2 and 3 describes the customer environment and contains relative byte addresses (RBAs) of the first D-, I-, and F-records.

Byte(s) 0 - 1	= C'H1'	Record ID
2 - 4	= X'000800'	RBA of first Product Record
8 -10	= X'0008C0'	RBA of first Installation Record
14 -16	= X'002180'	RBA of first Free Record

Record 4 (X) HEADER EXTENSION History File Record

This record (RBA=0000C0) contains additional information about the customer environment.

Byte(s) 0	= C'X'	Record ID
1 - 3	= X'FFFFFF'	Last in Queue

Record 135 (F) FREE History File Record

This record is the first in a chain of free records. All free records are chained by forward and backward pointers.

Byte(s) 0	= C'F'	Record ID
1 - 3	= X'0021C0'	RBA of F-Record in queue

Record 33 (D) PRODUCT History File Record

This record describes a product installed and contains pointers to associated queues.

Byte(s) 0	= C'D'	Record ID
1 - 3	= X'000100'	RBA of next D-Record
8 -10	= X'FFFFFF'	RBA of associated M-Record
14 -16	= X'000880'	RBA of associated O-Record
26 -28	= X'000840'	RBA of associated C-Record

Record 5 (D) PRODUCT History File Record

This record describes a product installed and contains pointers to associated queues.

Byte(s) 0	= C'D'	Record ID
1 - 3	= X'FFFFFF'	Last D-Record
8 -10	= X'FFFFFF'	RBA of associated M-Record
14 -16	= X'000300'	RBA of associated O-Record

Record 34 (C) Comment Record

This record contains comments associated with a installed product.

Byte 0	= C'C'	Record ID
Byte(s) 1 - 3	= X'FFFFFF'	Last in queue

Record 35 (O) COMPONENT-OF-FEATURE History File Record

This record describes the component associated with the product number shown in the respective D-Record.

Byte(s) 0	= C'0'	Record ID
1 - 3	= X'FFFFFF'	Last in Queue
7 - 9	= X'FFFFFF'	RBA of associated M-Record

Record 14 (I) INSTALLATION History File Record

Same meaning as record 18.

Byte(s) 0	= C'I'	Record ID
1 - 3	= X'FFFFFF'	Last in queue
4 - 6	= X'000380'	RBA of previous I-Record
7 - 9	= X'0006C0'	RBA of associated P-Record
13 -15	= X'002080'	RBA of associated L-Record

Record 131 (L) APAR/LOCAL FIX History File Record

This record contains information about an APAR/Local Fix and a pointer to an associated MODULE-IN-APAR record queue.

Byte(s) 0	= C'L'	Record ID
1 - 3	= X'FFFFFF'	Last in Queue
8-10	= X'002140'	RBA of associated N-Record
14-16	= X'002100'	RBA of associated C-Record
26-28	= X'0020C0'	RBA of associated Q-Record

Record 134 (N) MODULES-IN-APAR History File Record

This record contains information about a phase, module, or macro effected by an APAR/Local Fix.

Byte(s) 0	= C'N'	Record ID
1 - 3	= X'FFFFFF'	RBA of next N-Record in queue
7	= C'C'	Phase
9 -16	= C'MSHP '	Phase Name

Record 132 (Q) PRE/CO-REQUISITE History File Record

This record contains information about Pre- or Corequisite PTFs.

Byte(s) 0	= C'Q'	Record ID
1 - 3	= X'FFFFFF'	Last in Queue
11	= X'01'	Pre-Req Information
12	= X'02'	PTF
13 -21	= C'566630101'	Component ID
22 -29	= C'UD12345 '	PTF Number

Record 133 (C) COMMENT History File Record

The comment text of this record provides further information concerning the P-Record it is chained to.

Byte(s) 0	= C'C'	Record ID
1 - 3	= X'FFFFFF'	Last in Queue
7 -63	= C'PROGRAM CHECK	Comment text

Record 28 (P) PTF History File Record

This record describes a PTF installed and contains pointers to associated queues.

Byte(s) 0	= C'P'	Record ID
1 - 3	= X'FFFFFF'	RBA of next PTF-Record
8 -10	= X'000700'	RBA of associated A-Record
14 -16	= X'000740'	RBA of associated M-Record
26 -28	= X'FFFFFF'	RBA of associated Q-Record
32 -34	= X'0007C0'	RBA of associated C-Record
51 -58	= C'UD12345 '	PTF Number

Record 29 (A) APARS-IN-PTF History File Record

This record may contain 1 to 7 APAR numbers, belonging to the P-Record, to which this A-Record is chained.

Byte(s) 0	= C'A'	Record ID
1 - 3	= X'FFFFFF'	Last in Queue
7 -14	= C'DY23456 '	APAR-Number

Record 30 (M) MODULES-IN-PTF History File Record

This record may contain 1 to 5 names of phases, modules, or macros belonging to the D-Record, to which this M-Record is chained.

Byte(s) 0	= C'M'	Record ID
1 - 3	= X'000780'	Last in Queue
7	= C'R'	Module
9 -16	= C'IKRRLZAP'	Module Name

Record 31 (M) MODULES-IN-PTF History File Record

Same meaning as Record 30.

Byte(s) 0	= C'M'	Record ID
1 - 3	= X'FFFFFF'	Last in Queue
7	= C'R'	Module
9 -16	= C'IKRGTHD '	Module Name
17	= C'R'	Module
19 -26	= C'IKRDQFR '	Module Name

Record 32 (C) COMMENT History File Record

The comment text of this record provides further information concerning the P-Record it is chained to.

Byte(s) 0	= C'C'	Record ID
1 - 3	= X'FFFFFF'	Last in Queue
7 -63	= C'PROGRAM CHECK	Comment text

Record 18 (I) INSTALLATION History File Record

This record describes a component installed and contains pointers to associated queues.

Byte(s) 0	= C'I'	Record ID
1 - 3	= X'000400'	RBA of next I-Record
51 -53	= X'000980'	RBA of associated Y-Record
57 -59	= X'000580'	RBA of associated B-Record

Record 23 (B) LIBRARY NAME History File Record

This record describes the type and name of an associated library.

Byte(s)	0	= C'B'	Record ID
	1 - 3	= X'FFFFFF'	Last RBA
	4 - 6	= X'000440'	RBA to I-Record
	8	= C'P'	Library type
	9 -15	= C'IJSYSRS'	Library name
	16-23	= C'SYSLIB '	Sublibrary name
	24-24	= C'S'	Library type
	25-31	= C'SERVLIB'	Library name
	32-39	= C'G1\$301 '	Sublibrary name

Record 39 (Y) Installation Extension Record

This record is an extension of the installation record (I-type) and is used when generated members or included macros are associated with a particular component.

Byte(s)	0	= C'Y'	Record ID
	1 - 3	= X'FFFFFF'	Last in Queue
	8 -10	= X'000940'	RBA of associated G-Record
	14 -16	= X'0009C0'	RBA of associated V-Record

Record 38 (G) Generated Member Record

This record contains the name of the generated member and the date it was generated.

Byte(s)	0	= C'G'	Record ID
	1 - 3	= X'FFFFFF'	Last in Queue
	8 -10	= X'001240'	RBA of associated W-Record
	14 -16	= X'001400'	RBA of associated K-Record
	20 -22	= X'FFFFFF'	RBA of associated C-Record

Record 40 (V) Included Macro Reference Record

This record contains the names of the included macros.

Byte(s)	0	= C'V'	Record ID
	1 - 3	= X'000A00'	RBA of next V-Record in queue
	4 - 6	= X'000980'	RBA of Y-Record

Record 74 (W) Included Macro Reference Record

This record contains references to the names of the included macro associated with the generated member.

Byte(s) 0	= C'W'	Record ID
1 - 3	= X'001280'	RBA of next W-Record in queue

Record 81 (K) Comment Record

This record contains comments in fix description format that are associated with the generated member.

0	= C'K'	Record ID
Byte(s) 1 - 3	= X'001440'	RBA of next K-Record in queue

Index

C

Common MSHP Tables and Layouts 279

D

data areas

CRGHFCKD 283
CRGHFFBA 284
ENTAFF 364
ENTAPA 361
ENTCMN 360
ENTFIX 365
ENTIML 366
ENTINF 368
ENTRELLS 359
ENTREQ 362
ENTRLM 367
ENTSUP 363
IKR#AFF 289
IKR#ALT 292
IKR#APL 293
IKR#ARC 295
IKR#BCK 296
IKR#CIS 298
IKR#COR 300
IKR#CPT 301
IKR#CPY 302
IKR#CRE 303
IKR#DAT 304
IKR#DEL 305
IKR#DFN 306
IKR#DMP 308
IKR#EXE 309
IKR#GEN 310
IKR#ICL 311
IKR#INC 312
IKR#INF 313
IKR#IST 315
IKR#IVL 316
IKR#LIST 317
IKR#LKP 319
IKR#MRG 321
IKR#OR 322
IKR#PER 323
IKR#PTF 325
IKR#RDC 326
IKR#RMV 327
IKR#RPL 329
IKR#RQR 330
IKR#RST 332
IKR#RSTA 334

data areas (*continued*)

IKR#RSV 335
IKR#RTC 336
IKR#RVK 337
IKR#SCN 338
IKR#SEL 339
IKR#STL 340
IKR#SUP 343
IKR#TLR 344
IKR#UND 346
IKR#VER 347
IKRAIP 379
IKRAPA 380
IKRBFRO 285
IKRCOF 381
IKRCOM 382
IKRCOMRG 279
IKRFRE 383
IKRGEN 384
IKRHD1 386
IKRHD2 387
IKRHD3 388
IKRHDX 389
IKRIFCTX 286
IKRIMR 390
IKRIMT 391
IKRINS 392
IKRINX 394
IKRLIB 395
IKRLSI 396
IKRMIA 397
IKRMIP 398
IKRPRD 399
IKRPRE 401
IKRPTF 402
IKRPTX 404
IKRSUP 405
IKRXCOR 349
IKRXCOPY 350
IKRXCUM 352
IKRXCUM 352
IKRXDMP 355
IKRXENT 356
IKRXLKP 369
IKRXMRG 370
IKRXPER 371
IKRXRDC 372
IKRXRMV 373
IKRXRTC 374
IKRXSTL 375
Index to Data Areas 275
information 273

data areas (*continued*)

list 274

data sets

system 3

Diagnostic Aids 407

Module-Message Interrelations 407

Module-Message X-Reference 407

diagnostic tools 4

dump

history file sample 417

E

environmental characteristics 1

error handling 3

F

function flow 7

H

history file

dump analysis of 417

records of 377

structure of 377

summary 378

History File Sample Dump 418

Description of History File Records 434

I

interface

system 3

interface control 8

System Programs Called by MSHP 13

Internal Representation 348

Introduction 1

L

Layout of Data Areas 278

M

messages

parser 3

Module Description 17

Module-IKRCOMRG Interrelations 267

Module-Macro Interrelations 255

Module-Module Interrelations 240

Modules

IKRAAPAR 18

IKRACMP1 19

IKRACOMP 20

IKRAGEN 21

IKRAPPLY 22

IKRAPRD 23

Module Description (*continued*)

Modules (*continued*)

IKRAPTF 24

IKRARCH 25

IKRBARDT 26

IKRBGRDT 27

IKRBINS 28

IKRBIO 29

IKRBKPR 30

IKRBKT 31

IKRBKUP 32

IKRBPRDT 33

IKRCGT 34

IKRCHECK 36

IKRCKASS 38

IKRCKCMP 40

IKRCKCTL 41

IKRCKD 42

IKRCKFIX 43

IKRCKLV2 44

IKRCKPTF 45

IKRCLMSH 46

IKRCLOSE 47

IKRCMDS 48

IKRCNVRT 49

IKRCNVTB 51

IKRCOMR 52

IKRCOPY 53

IKRCPIN 54

IKRCREAT 55

IKRCUTHF 56

IKRCVTLU 57

IKRDATE 58

IKRDEBUG 59

IKRDEL 60

IKRDEQ 61

IKRDIAGN 63

IKRDLELE 64

IKRDLIN 65

IKRDQFR 67

IKREAPL 68

IKREARC 69

IKREBCK 70

IKRECOR 71

IKRECPY 72

IKRECRE 73

IKREDMP 66, 74

IKREINC 75

IKRELKP 76

IKRELST 77

IKREMBDY 78

IKREMRG 80

IKREMU 81

IKREPER 83

IKRERDC 84

IKRERMV 85

IKRERST 86

Module Description (continued)

Modules (continued)

IKRERTC 87
 IKRERVK 88
 IKRESEL 89
 IKRESTL 90
 IKRETLR 91
 IKREUND 92
 IKREXIT 93
 IKRFX 94
 IKRFRBUF 95
 IKRGOGO 96
 IKRGTAPR 99
 IKRGTBUF 100
 IKRGTCMP 101
 IKRGTFR 102
 IKRGTGEN 103
 IKRGTGRP 104
 IKRGTHD 105
 IKRGTIMT 106
 IKRGTINF 107
 IKRGTMSH 108
 IKRGTPTF 109
 IKRGTREC 110
 IKRGTTAP 111
 IKRGWA 112
 IKRHDBKT 113
 IKRHDLV2 114
 IKRHSLA 116
 IKRIMSMT 117
 IKRINMSH 118
 IKRINSTL 119
 IKRISLNK 121
 IKRISVBK 122
 IKRLAPA 124
 IKRLCMPR 125
 IKRLGNRC 126
 IKRLIBST 127
 IKRLINCM 134
 IKRLKUP 135
 IKRLMLF 136
 IKRLMOD 137
 IKRLOAD 138
 IKRLOFI 140
 IKRLSUP 142
 IKRMAIN 143
 IKRMERGE 144
 IKRMOD 145
 IKRMRGIN 146
 IKRNOTE 147
 IKROPBUF 148
 IKROPEN 149
 IKROPHF 150
 IKROPLNK 151
 IKROPMSH 153
 IKRPAPAL 154
 IKRPARSE 155

Module Description (continued)

Modules (continued)

IKRPCREQ 156
 IKRPDZAP 157
 IKRPERS 158
 IKRPGENL 159
 IKRPHMSH 160
 IKRPIO 161
 IKRPPTFL 162
 IKRPRAPL 163
 IKRPRCS 164
 IKRPSDMD 165
 IKRPTLNK 166
 IKRPTMSH 167
 IKRQUEUE 168
 IKRRACSV 169
 IKRRAESV 170
 IKRRALNK 171
 IKRRASRV 172
 IKRRCV 173
 IKRRDCE 174
 IKRRDIPT 178
 IKRRDLV2 179
 IKRRDNXT 180
 IKRREGCK 181
 IKRREQFR 182
 IKRREWR 183
 IKRRL2 185
 IKRRL3 186
 IKRRL4 188
 IKRRL5 189
 IKRRL6 190
 IKRRL7 191
 IKRRLZAP 184
 IKRRMGEN 192
 IKRRTC 193
 IKRTRVE 194
 IKRRWBUF 195
 IKRSAASM 196
 IKRSALIB 197
 IKRSAMNT 198
 IKRSAPL 199
 IKRSEND 200
 IKRSERR 201
 IKRSET 202
 IKRSHAPR 207
 IKRSHCMP 208
 IKRSHGEN 209
 IKRSHGRP 210
 IKRSHPRD 211
 IKRSHPTF 212
 IKRSHREL 213
 IKRSLA 214
 IKRSLZAP 215
 IKRSPCLS 217
 IKRSREPQ 218
 IKRSRTMD 219

Module Description (*continued*)

Modules (*continued*)

IKRSTMTS 221
IKRSVCMD 222
IKRSWTHF 223
IKRTAIL 224
IKRTRLCP 226
IKRUIRP 227
IKRUNDO 228
IKRUOMT 229
IKRUPGR 230
IKRWKF 233
IKRWOF 234
IKRWOR 235
IKRWRSQ 236
IKRWTO 237
IKRWTP 238

P

parser 3
processor routines 6

S

semantic checker 5
SIF table 287
SIT table 287
standardized internal format 287
Standardized Internal Table (SIT) 287

V

value table(VTAB) 287
VTAB (value table) 287
SIF Tables 288



File Number: S370/S390-34
Program Number: 5686-007



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC33-6335-00

