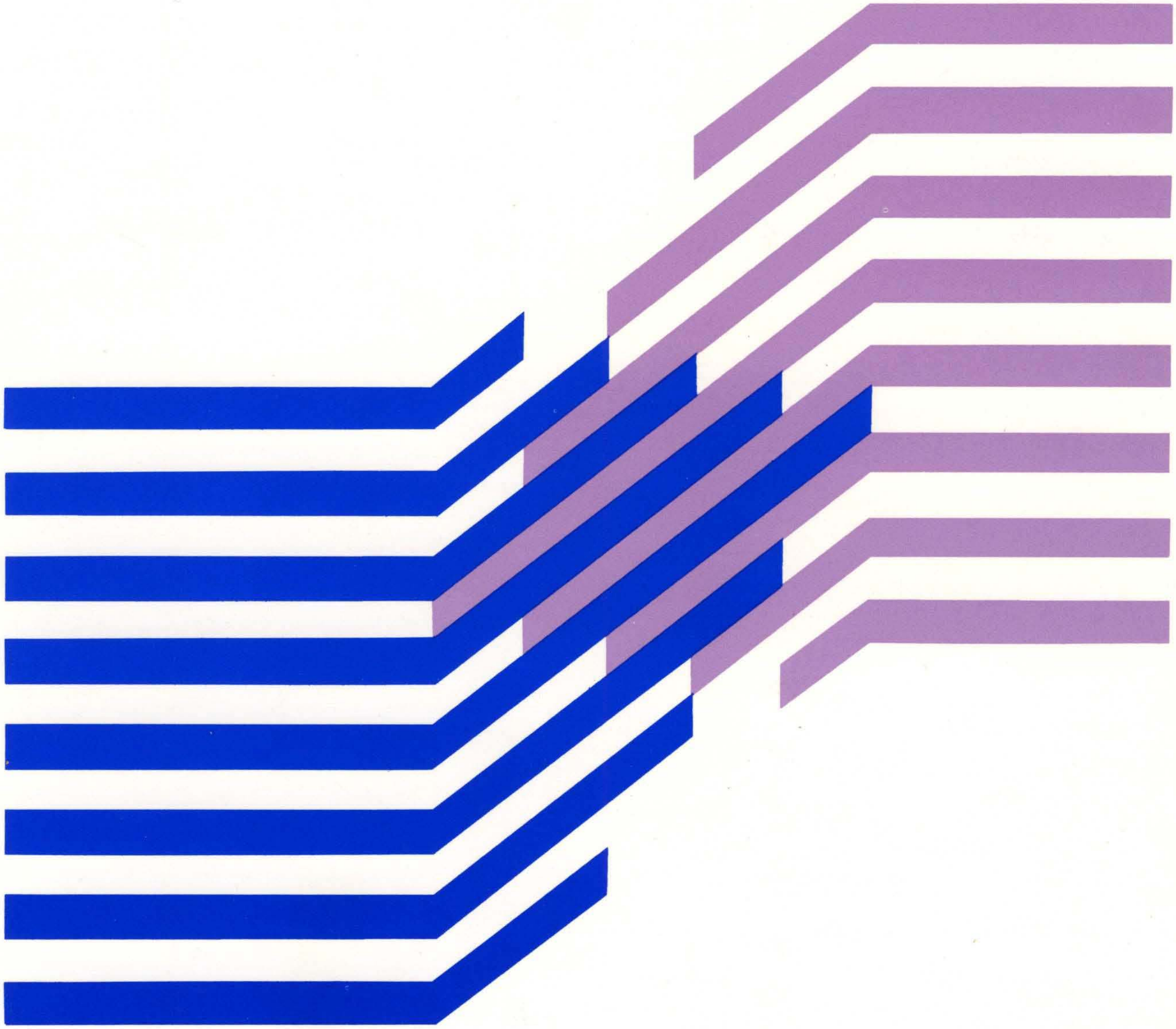
LY28-1420-2

MVS/ESA
Component Diagnosis and Logic: Module Descriptions

MVS/System Product
JES2 Version 3
JES3 Version 3



LY28-1420-2

MVS/ESA
Component Diagnosis and Logic: Module Descriptions

MVS/System Product
JES2 Version 3
JES3 Version 3

Third Edition (December, 1989)

This is a major revision of, and obsoletes, LY28-1420-1. See the Summary of Changes following the Contents for a summary of the changes made to this manual. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

This edition applies to Version 3 of MVS/System Product 5685-001 or 5685-002 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 Bibliography*, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products or services do not imply that IBM intends to make these available in all countries in which IBM operates. References to IBM products in this document do not imply that functionally equivalent products may be used. The security certification of the trusted computing base that includes the products discussed herein covers certain IBM products. Please contact the manufacturer of any product you may consider to be functionally equivalent for information on that product's security classification. This statement does not expressly or implicitly waive any intellectual property right IBM may hold in any product mentioned herein.

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Information Development, Department D58, Building 921-2, PO Box 950, Poughkeepsie, NY 12602. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

PROGRAMMING INTERFACES

This book does NOT contain any programming interface information.

About This Book

This books provides module descriptions for the modules documented in the following *MVS/ESA™ Component Diagnosis and Logic* books.

Book Titles

MVS/ESA Comp Diag and Logic: Address Space Control
MVS/ESA Comp Diag and Logic: Allocation/Unallocation
MVS/ESA Comp Diag and Logic: Alternate CPU Recovery
MVS/ESA Comp Diag and Logic: Auxiliary Storage Manager
MVS/ESA Comp Diag and Logic: Command Processing
MVS/ESA Comp Diag and Logic: Communications Task
MVS/ESA Comp Diag and Logic: Contents Supervision
MVS/ESA Comp Diag and Logic: Converter/Interpreter
MVS/ESA Comp Diag and Logic: Dump Analysis and Elimination
MVS/ESA Comp Diag and Logic: Dumping Services
MVS/ESA Comp Diag and Logic: Dynamic Device Reconfiguration
MVS/ESA Comp Diag and Logic: Event Notification Facility
MVS/ESA Comp Diag and Logic: EXCP Processor
MVS/ESA Comp Diag and Logic: Functional Subsystem Interface
MVS/ESA Comp Diag and Logic: Global Resource Serialization
MVS/ESA Comp Diag and Logic: Initiator/Terminator
MVS/ESA Comp Diag and Logic: Input/Output Supervisor
MVS/ESA Comp Diag and Logic: JES/Scheduler Services
MVS/ESA Comp Diag and Logic: Machine Check Handler
MVS/ESA Comp Diag and Logic: Master Subsystem/Subsystem
MVS/ESA Comp Diag and Logic: Recovery Termination Manager
MVS/ESA Comp Diag and Logic: Scheduler JCL Facility
MVS/ESA Comp Diag and Logic: Scheduler Restart
MVS/ESA Comp Diag and Logic: Supervisor Control and Dispatcher
MVS/ESA Comp Diag and Logic: SWA Manager
MVS/ESA Comp Diag and Logic: Symptom Record
MVS/ESA Comp Diag and Logic: System Initialization
MVS/ESA Comp Diag and Logic: System Management Facilities
MVS/ESA Comp Diag and Logic: System Resource Manager
MVS/ESA Comp Diag and Logic: Task Management
MVS/ESA Comp Diag and Logic: Timer Supervision
MVS/ESA Comp Diag and Logic: Trace
MVS/ESA Comp Diag and Logic: Virtual Storage Management

Trademarks

The following is a trademark of International Business Machines Corporation:

MVS/ESA™

Who This Book Is For

The books listed above are intended for people who debug or modify the control program. This book is designed for anyone who wants to gain a general understanding of the modules that make up the processing for any component listed above.

How This Book Is Organized

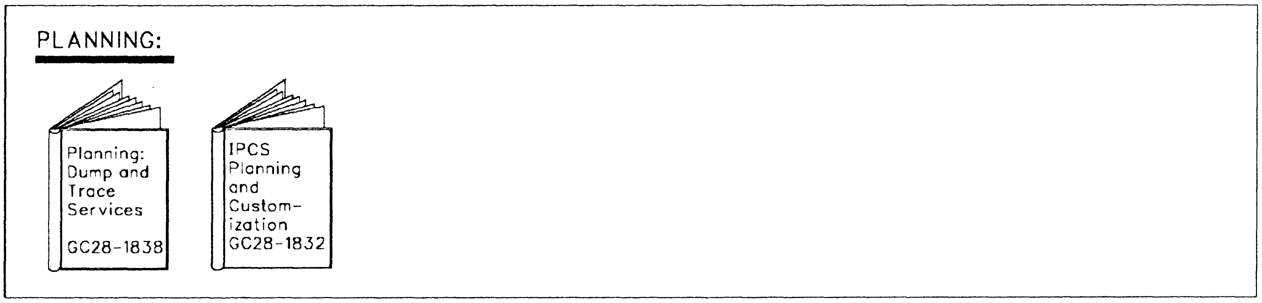
The module descriptions are arranged alphanumerically by module name. The description of each object module includes:

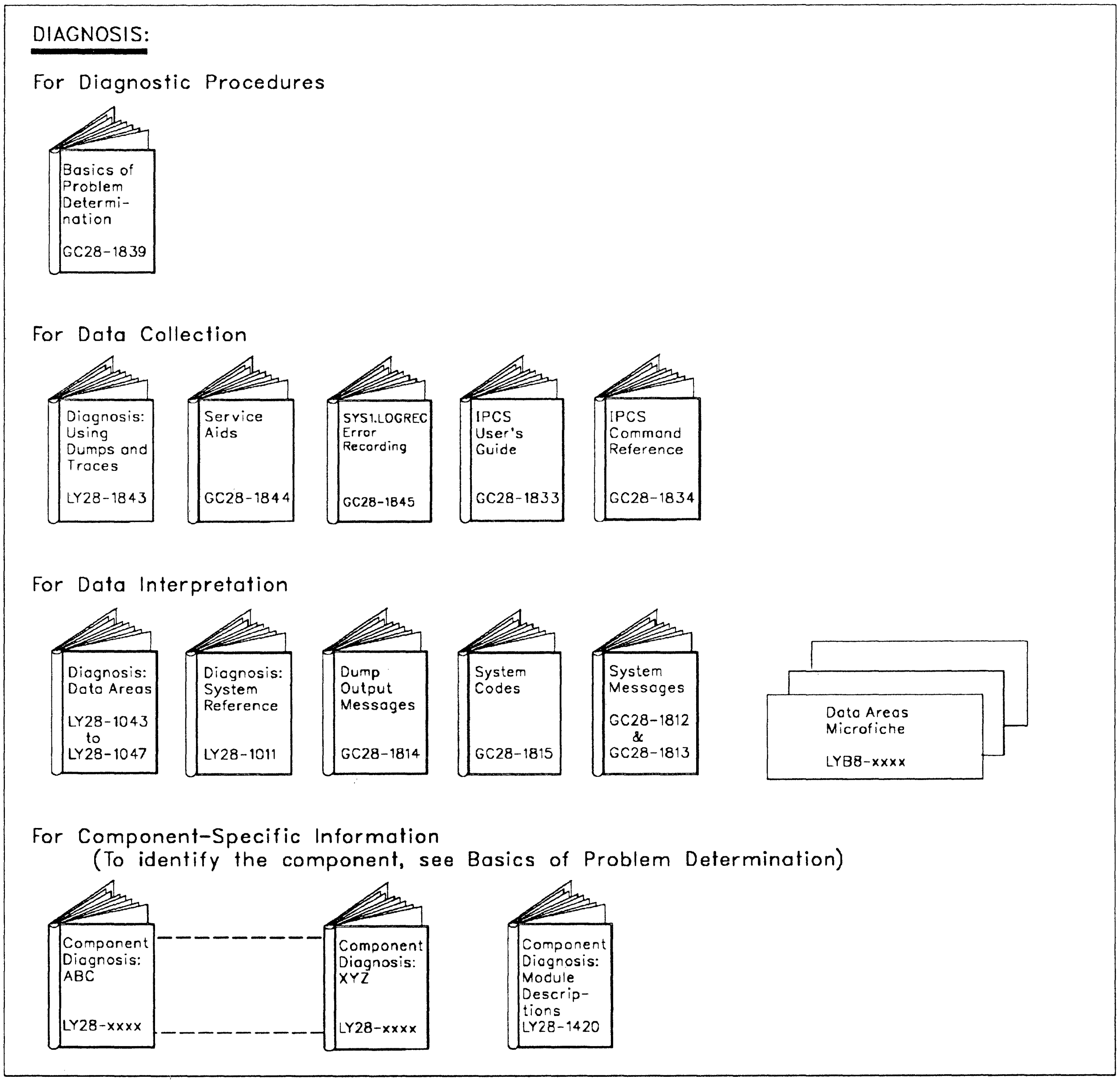
- A summary of the function of the module.
- The names of the modules that pass control to it.
- The names of the modules that receive control from it.

If there are two names in a title, the first is the CSECT name and the second (in parentheses) is the assembler module name. If there is only one name in the title, it is the name of both the CSECT and the assembler module.

Note: Module descriptions are not included for components whose modules are not available in source code.

Diagnosis Library


PLANNING:
Planning: Dump and Trace Services
GC28-1838
IPCS Planning and Custom-ization
GC28-1832



Summary of Changes

Summary of Changes
for LY28-1420-2
MVS/System Product Version 3 Release 1.3

Changes have been made throughout this book to support MVS/SP Version 3 Release 1.3.

New Information: The following module descriptions have been added:

IEATTUSD
IEAVADSP
IEECB813
IEECB815
IEECB929
IEECB930
IEE8C03D
IEFAB4C0
IEFAB4C1
IEFAB4C2
IEFAB4C3
IEFAB4C4
IEFAB4FG
IEFDB475
IEFDB476
IEFDB477
IEFDB478

Changed Information: The following module descriptions have been changed:

COF
IEAVAD01
IEAVTABD
IEECB816
IEEDMCHP
IEEDMCPU
IEEDMMSG
IEEMPDEV
IEEMPDM
IEEVCPR
IEEVOLSI
IEEVORGI
IEEVSTEE
IEEVSTFA
IEEVSTGL
IEFAB42A

Deleted Information: The following module descriptions have been removed:

IEAVTSYS
IEECB813
IEFAB4A2

In addition, the Diagnosis Library figure has been updated to reflect the deletion of the *Component Diagnosis: Master Index* from the library. See *MVS/ESA Master Index for SP Version 3, GC28-1827* for master index information.

Summary of Changes
for LY28-1420-1
MVS/System Product Version 3 Release 1.0e

This major revision consists of changes to support MVS/System Product Version 3 Release 1.0e.

New Information: The following module descriptions have been added:

IEAVMP08
IEAVNPCD
IEAVNP47
IEAVSPDM

The following modules have been added:

IEAVEEXP
ISGNBRSE
IEAVLKRM
ISGNFINT
ISGDQDMP
ISGDRSAQ
ISGDQDEQ

Changed Information: The following module descriptions have been changed:

IEAVEBBR
IEAVECMS
IEAVEDR
IEAVEDRS
IEAVEDS0
IEAVEEE0
IEAVEEXT
IEAVEEE0
IEAVEGR

IEAVEINT
IEAVEIO
IEAVEJST
IEAVELKR
IEAVEMS0
IEAVEMS3
IEAVEMS6
IEAVEMS3
IEAVEMS7

IEAVEMS8
IEAVEMS9
IEAVEPC
IEAVEPCO
IEAVEPDR
IEAVEQV0
IEAVEQV4
IEAVEQV0
IEAVEQV5

IEAVERI
IEAVERP
IEAVESC0
IEAVESGP
IEAVESLK
IEAVESRT
IEAVESVC
IEAVESRT
IEAVETCL

IEAVLSCR
IEAVLSIH
IEAVLSLC
IEAVVJST

In addition, the Index has been removed, as well as, minor technical and editorial changes.

Deleted Information: The following modules have been removed.

CSVLLCRE CSVLLSCH CSVLLS01 CSVLLTCH CSVLLTSS
CSVLLXIT

The following module descriptions have been removed.

IEAVEF00 IEAVEGLU ISGBRF ISGBSM ISGGQWB0
ISGGRP00

Summary of Changes
for LY28-1420-0
MVS/System Product Version 3 Release 1.0

This book contains information presented in *MVS/Extended Architecture System Logic Library: Module Descriptions*, LY28-1605-1.

New Information: The following modules are new:

CBPUS023
CBPUS024
CBPUS025
CBPUS026
CBPUS031
CBPUS050
IEAMLT22
IEAMLT24
IEAMLT27
IEAMLT33
IEAVAR08
IEAVNP00
IEAVNP27
IEAVTSDS
IEAVTSD2
IEAMTSD3
IEAMTSD4
IEAMTSD5
IEAVWSAT
IECVDDT2
IECVDDT4
IECVDDT5
IECVDDT7
IECVEXSM
IECVOID
IECVURVL
IEEDMCHP
IEEDMCPU
IEEDMDEV
IEEDMSTR
IEERDUMP
IEERESTA
IEERSYMR
IEESAFF
IEESCPVF
IEEVCPH
IEEVCPLF
IEEVCPLN
IEEVCPRA
IEEVCVSR
IEEVVFLF
IEEVVFLF
IEEVVFLN
IEE7603D
IEFAB4C0
IEFAB4C1
IEFAB4C2
IEFAB4C3
IEFAB4C4
IEFAB4FG
ILRAB42A
ILRASHCH
ILRPDPAG
ILRPDSWP
ILRPDVIO
ILRPOS01
ILRSWSCN
IOSCONSL
IOSCPARZ
IOSERDTE
IOSERDTS
IOSEREVH
IOSERIT
IOSEROUT
IOSVRMIHI
IOSVATTN
IOSVDATA
IOSVDEVN
IOSVDSTF
IOSVFMTI
IOSVFMTU
IOSVIEDN
IOSVIOQM
IOSVMODI
IOSVMODU
IOSVMOD1
IOSVMOD2
IOSVUCBS
IOSVUCBV
IRARMWTO

Changed Information: The following modules have been updated:

CSVFRR
CSVGETMD
CSVSBRTN
CSVSYNCH
CSVVFCRE
CSVVFCR1
CSVVFMEM
CSVVFNDE
CSVVFTCH
CSVXCTL
CSVXXCTL
IARVTSSD
IEAVAD00
IEAVEGLU
IEAVID00
IEAVTSDX
IEAVTSRB
IEAVTSSD
IECIOFMT
IECIOFT1
IECVEXCP
IECVEXPR
IECVTCCW
IEECB806
IEECB808
IEECB813
IEEDMSID
IEEMB884
IEEMPDEV
IEEVCPR
IEEVFTFA
IEEVSTEE
IEEVSTEL
IEEVSTGL
IEEVSTGP
IEEVSTPE
IEEVWKUP
IEE0303D
IEE0703D
IEE1603D
IEE40110
IEE4303D
IEE5403D
IEE5703D
IEE5903D
IEE6403D
IEE6803D
IEE7103D
IEE7503D
IEE7730D
IEE7803D
IEE8B03D
IEE8603D
IEFAB475
IEFAB476
IEFAB477
IEFAB478
IEXVEXFR
IGFPMRTH
IGFPXMFA
IGVRRTR
IGVRSQA
IGVRVSM
ILRACT
ILRCMP
ILRCMP01
ILRCMSRB
ILRCPBLD
ILRDRV01
ILRFRS01
ILRFTASM
ILRFTMAN
ILRFTRTN
ILRGOS
ILRGOS01
ILRIODRV
ILRIOFRR
ILROPS00
ILRPAGCM
ILRPOS
ILRPREAD
ILRRLG
ILRRSMRB
ILRSAV
ILRSRBC
ILRSRB01
ILRSWAP
ILRSWP01
ILRTERMR
ILRTMI01
ILRTMRLG
IRARMCPM
IRARMST2
IRARMSWP
ISGCMDI
ISGCRET1

Minor technical and editorial changes throughout the publication.

Deleted Information: The following modules have been deleted:

IEAIPL03
IEAIPL40
IEAVNPF2
IEAVNP02
IECDPRFX
IECDPRF2
IECLMSGD
IECLMSGT
IECVXDAS
IECVXDAT
IECVXDAU
IECVXDRS
IECVXDRT
IECVXPRE
IECVXTAT
IECVXT3S
IEECB814
IEECB815
IEECB929
IEECB930
IEE8C03D

Note: Module descriptions for components whose modules are not available in source code are not included in this publication.

Module Descriptions

ADYEXT – DAE Symptom Extraction Routine

Function: ADYEXT provides the visible keywords and the symptom data for one or more MVS and RETAIN symptoms.

Entry-Point ADYEXT:

Callers: ADYHDFMT, ADYPRED.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: ADYEXTT.

ADYEXTT – DAE Keyword Data Access Routine

Function: ADYEXTT accesses dump analysis and elimination keyword data and returns it to the caller in the symptom extraction parameter list (DSX).

Entry-Point ADYEXTT:

Callers: ADYEXT.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: None.

ADYHDFMT – DAE Dump Header Record Format Routine

Function: ADYHDFMT formats and prints the DAE data in the dump header record for either an SDUMP or a SYSMDUMP.

Entry-Point ADYHDFMT:

Callers: AMDPRDMP.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: ADYEXT, AMDPRUIM.

ADYIO – DAE Input/Output Processor Routine

Function: ADYIO handles a subset of I/O operations for any DASD data set that conforms to a queued sequential access method (QSAM) data control block (DCB).

Entry-Point ADYIO:

Callers: ADYTRNS.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: None.

ADYMSG – DAE Message Generation Routine

Function: ADYMSG generates and writes dump analysis and elimination (DAE) messages to the operator console.

Entry-Point ADYMSG:

Callers: ADYSETP, ADYTRNS.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: None.

ADYPARS – DAE Text Parsing Routine

Function: For each character string found in the input text (TXALL), ADYPARS creates an entry in the output table (TXAZZ). Each entry contains the following: a classification of the field type, the length of the field, and the pointer to the start of the field.

Entry-Point ADYPARS:

Callers: ADYSETP.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: None.

ADYPRED – DAE Pre-Dump Routine for SDUMP and SYSMDUMP

Function: ADYPRED constructs a symptom string in the dump header record. ADYPRED also determines whether the dump is a duplicate, and, if DAE is active and all suppression criteria are met, suppresses any duplicate dumps.

Entry-Point ADYPRED:

Callers: IEAVAD00, IEAVTSDX.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point ADYPREDR: Provides recovery for pre-dump processing.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

External-References: ADYEXT.

ADYPSTD — DAE Post-Dump Exit Routine

Function: ADYPSTD schedules a transaction on the DAE transaction processor queue whenever an SDUMP or a SYSMDUMP completes or is suppressed.

Entry-Point ADYPSTD:

Callers: IEAVTSEP.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point ADYPSTDR: Provides recovery for entry point ADYPSTD.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

External-References: ADYTRNS.

ADYSETP — DAE SET Command Processor

Function: ADYSETP reads, parses, and interprets DAE parameter records from members of SYS1.PARMLIB that are indicated by the SETDAE command. ADYSETP creates a transaction entry to represent the DAE commands.

Entry-Point ADYSETP:

Callers: IEEMB811.

Exit-Normal: Returns to Master Scheduler.

Exit-Error: Returns to Master Scheduler.

Entry-Point ADYSETPR: Provides recovery for entry point ADYSETP.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

External-References: ADYMSG, ADYPARS, ADYTRNS, IEEMB878.

ADYTRNS — DAE Transaction Processor

Function: ADYTRNS processes all the entries on the DAE high-priority and low-priority transaction queues.

Entry-Point ADYTRNS:

Callers: IEAVTDSV and IEAVTDSE attach ADYTRNS; ADYPSTD and ADYSETP post ADYTRNS.

Exit-Normal: There is no exit-normal condition. ADYTRNS is a never-ending task.

Exit-Error: Returns to the caller.

Entry-Point ADYTRNSR: Provides recovery for errors in DAE transaction processing.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

External-References: ADYIO, ADYMSG, ADYDFLT.

ASE — Address Space Services

Function: All modules and macros with this prefix are part of the availability manager. The diagnosis information for this component is in *MVS/ESA Component Diagnosis: Address Space Services.*

ASRFDE — SYMREC Detail Edit Report Formatter

Function: This module produces the EREP Detail Edit report for symptom record type LOGREC records.

Entry-Point ASRFDE: See function.

Callers: ASRROUT

Exit-Normal: Returns to caller.

Exit-Error: There are no exit error conditions.

External-References: None.

ASRFDS — SYMREC Detail Summary Report Formatter

Function: This module produces the EREP Detail Summary report for symptom record type LOGREC records.

Entry-Point ASRFDS: See function.

Callers: ASRROUT

Exit-Normal: Returns to caller.

Exit-Error: There are no exit error conditions.

External-References: ASRPSRV

ASRFEH – SYMREC Event History Report Formatter

Function: This module produces the EREP Event History report for symptom record type LOGREC records.

Entry-Point ASRFEH: See function.

Callers: ASRROUT

Exit-Normal: Returns to caller.

Exit-Error: There are no exit error conditions.

External-References: None.

ASRKEYS – Symptom Key Table

Function: This module searches an internal table to determine the appropriate description for a symptom key, and returns the description to the caller.

Entry-Point ASRKEYS: See function.

Callers: ASRSYMV, ASRFDE

Exit-Normal: Returns to caller.

Exit-Error: There are no exit error conditions.

External-References: None.

ASRPSRV – SYMREC Formatter Print Routine

Function: This module prints a line of formatted data to the EREP report data set.

Entry-Point ASRPSRV: See function.

Callers: ASRFDE, ASRFDS

Exit-Normal: Returns to caller.

Exit-Error: There are no exit error conditions.

External-References: IFCPRINT.

ASRROUT – SYMREC Formatter Routing Module

Function: This module formats symptom records for the Detail Edit, Detail Summary, or Event History EREP reports.

Entry-Point ASRROUT: See function.

Callers: EREP

Exit-Normal: Returns to caller.

Exit-Error: There are no exit error conditions.

External-References: ASRPSRV.

ASRSYMV – Symptom Dump Formatting Routine

Function: This module is a dump formatting exit that can be called from either IPCS or PRDMP.

Entry-Point ASRSYMV: This module formats the symptom strings contained in the second half of the dump header record.

Callers: IPCS and PRDMP.

Exit-Normal: Returns to caller.

Exit-Error: There are no exit error conditions.

External-References: ASRKEYS.

ASRSERVP – SYMREC Macro Service Routine

Function: This module is the MVS service routine for the SYMREC macro.

Entry-Point ASRSERVP: See function.

Callers: Any.

Exit-Normal: Returns to the caller via PT instruction.

Exit-Error: There are no exit error conditions.

Entry-Point ASRSERVR: This entry point is both the first level FRR and nested FRR for ASRSERVP.

Callers: RTM1

Exit-Normal: Returns to RTM1.

Exit-Error: Returns to RTM1.

Entry-Point ASRSERVM: This entry point is the resource manager for the SYMREC service and it gets control of every address space termination.

Callers: IEAVTMMT

Exit-Normal: Returns to caller.

Exit-Error: Returns to caller.

External-References: ASRGLTAB.

AVM — Availability Manager

Function: All modules and macros with this prefix are part of the availability manager. The diagnosis information for this component is in *MVS/ESA Component Diagnosis: Availability Manager.*

COF — Virtual Lookaside Facility

| **Function:** All modules and macros with this prefix
| are part of the Virtual Lookaside Facility. The diagnosis information for this component is in *MVS/ESA Component Diagnosis: Virtual Lookaside Facility.*

CBPUS014 — Unit Information Module for CTC devices

Function: CBPUS014 defines the device dependent support for the CTC device.

Entry-Point CBPUS014: The function of this entry point is to define the device dependent support for the CTC device.

Callers: CBPIFIR - IOS Functional Initialization Routine and CBPICBBR - IOS Control Block Build Routine

Exit-Normal: Returns to caller.

Exit-Error: If abend occurs the ESTAE routine established in module CBPMVSCP will provide the diagnostic information.

External-References:

- CBPADIT
- CBPIDFT
- CBPIFEAT
- CBPIPARM

CBPUS023 — Unit Information Module for Communication Devices

Function: This module defines the device dependent support for the following:

- 3704 Communication Control Unit.
- 3705 Communication Control Unit.
- 3791L Controller.
- 7770 Audio Response Device.

Callers: CBPIFIR and CBPICBBR.

Exit-Normal: Returns to caller.

External-References:

- CBPADIT
- CBPIDFT
- CBPIPARM

CBPUS024 — Unit Information Module for the 1030, 1050, 1050X, 115A, 2740, 2740C, and 2740X

Function: This module defines the device dependent support for the 1030, 1050, 1050X, 115A, 2740, 2740C, and 2740X.

Callers: CBPIFIR and CBPICBBR.

Exit-Normal: Returns to the caller.

External-References:

- CBPADIT
- CBPIDFT
- CBPIFEAT
- CBPIPARM

CBPUS025 — Unit Information Module for Teleprocessing Devices

Function: This module defines the device dependent support for the following miscellaneous teleprocessing devices:

- 2741P terminal.
- 2741C terminal.
- 83B3 Selective calling terminal.
- TWX teletype.
- WTTA telegraph terminal.

Callers: CBPIFIR and CBPICBBR.

Exit-Normal: Returns to the caller.

External-References:

- CBPADIT
- CBPIDFT
- CBPIFEAT
- CBPIPARM

CBPUS026 — Unit Information Module for Binary Synchronous Communication System 1, 2 or 3

Function: This module defines the device dependent support for the following:

- BSC1.
- BSC2.
- BSC3.

Callers: CBPIFIR and CBPICBBR.

Exit-Normal: Returns to the caller.

External-References:

- CBPADIT

- CBPIDFT
- CBPIFEAT
- CBPIPARM

CBPUS031 – Unit Information Module for the 3284 and 3286 Printers

Function: This module defines the device dependent support for the 3284 and 3286 printers.

Callers: CBPIFIR and CBPICBBR.

Exit-Normal: Returns to the caller.

External-References:

- CBPADIT
- CBPIDFT
- CBPIFEAT
- CBPIPARM

CBPUS050 – Unit Information Module for the DUMMY Device

Function: This module defines the device dependent support for the DUMMY device.

Callers: CBPIFIR and CBPICBBR.

Exit-Normal: Returns to the caller.

External-References:

- CBPADIT
- CBPIDFT
- CBPIPARM

CSR – Callable Service Requests

Function: All modules with this prefix are part of callable service requests. The diagnosis information for this component is in *MVS/ESA Component Diagnosis: Callable Service Requests.*

CSVABEND – ABEND Message Module

Function: This module issues the appropriate diagnostic message and, if necessary, the ABEND SVC for any error condition that is detected by contents supervisor.

Entry-Point CSVABEND:

Callers: CSVATTCH, CSVLINK, CSVLOAD, CSVSYNCH, CSVXCTL, CSVXLINK, CSVXLOAD, CSVXXCTL.

Exit-Normal: An appropriate ABEND SVC for the error condition.

Exit-Normal: Returns to the caller.

Exit-Error: C0D ABEND SVC.

External-References: CSVRFREE.

CSVARMGR – ABEND Resource Manager

Function: This module cleans up resources associated with a task when the task is terminating abnormally.

Entry-Point CSVARMGR:

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

External-References: CSVSBCHK, IEA0PT01.

CSVATTCH – ATTACH SVC Service Branch Entry Routine

Function: As one phase of the ATTACH service, CSVATTCH obtains a usable copy of a requested module and builds control blocks needed by the dispatcher to pass control to the module.

Entry-Point CSVATTCH or IEAQCS01 (alias):

Callers: Users of the ATTACH SVC service.

Exit-Normal: Transfer control to EXIT prolog via a branch entry.

Exit-Error: ABEND SVC via CSVABEND.

External-References: CSVABEND, CSVS2GWK, CSVRBPRB, CSVSBMFD.

CSVDELET – DELETE SVC Service Routine

Function: DELETE cancels the effect of a previous LOAD request. If DELETE cancels the only outstanding LOAD request for the module and no other requirements exist for the module, the virtual storage occupied by the module is released for reuse.

Entry-Point CSVDELET or IGC009 (alias):

Callers: Users of the DELETE SVC service.

Exit-Normal: Transfer control to EXIT prolog via a branch entry.

Exit-Error: No exit error conditions.

External-References: CSVSBLDL.

CSVEXIT – Exit Resource Manager

Function: This module cleans up resources when a PRB exits or when a task is about to terminate.

Entry-Point CSVEXIT:

Callers: EXIT service (SVC 3).

Exit-Normal: Returns control to the exit routine.

Exit-Error: No exit error conditions.

External-References: CSVSBCHK, IEA0PT01.

CSVFRR – Contents Supervision Functional Recovery Routines

Function: This module provides recovery capability and queue validation for contents supervision.

Entry-Point CSVFRR: Intercepts unwanted ABEND conditions that occur during the main processing of contents supervision.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

Entry-Point CSVFRR2: Intercepts unwanted ABEND conditions that occur during the execution of the recovery module CSVFRR.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

Entry-Point CSVFRRET: Intercepts unwanted ABEND conditions that occur during the execution of the exit resource manager, CSVEXIT.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

Entry-Point CSVFRCDV: Validates individual job pack area (JPA) queue contents directory entries (CDEs).

Callers: IEAVEQV0.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point CSVFRLLV: Validates individual load list elements.

Callers: IEAVEQV0.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IEAVSPER, IEAVEQV1.

CSVGETMD – BLDL and Fetch Interface Routine

Function: This module controls the library search sequence for a request and, if necessary, invokes the BLDL and fetch services to obtain a usable copy of a requested module.

Entry-Point CSVGETMD:

Callers: CSVSBRTN.

Exit-Normal: Returns to the caller.

Exit-Error: Returns to the caller, but flags may be set in the SVRB extended save area to indicate that an ABEND should occur.

Entry-Point CSVGEXLD: Processes explicit load requests.

Callers: CSVXLOAD.

Exit-Normal: Returns to the caller.

Exit-Error: Returns to the caller, but flags may be set in the SVRB extended save area to indicate that an ABEND should occur.

External-References:

- CSVS2CDQ
- CSVS2QWK
- CSVS2LSR
- CSVS2ROQ
- FETCHRTN
- IEAVTPVL
- TSOTEST

CSVLINK – LINK SVC Service Routine

Function: This module obtains a usable copy of a requested program and builds the control blocks necessary for the dispatcher to transfer control to the program. After the requested program executes, control is returned to the issuer of the LINK macro.

Entry-Point CSVLINK or IGC006 (alias):

Callers: Users of the LINK SVC service.

Exit-Normal: Transfers control to EXIT prolog (IEAVEEXP) via a branch entry.

Exit-Error: Transfers control to EXIT prolog (IEAVEEXP) via a branch entry. A user-supplied ERRET routine will eventually receive control.

Exit-Error: ABEND SVC via CSVABEND.

External-References: CSVABEND, CSVFRR, CSVS2GWK, CSVSBMFD, CSVRBPRB.

CSVLL – Contents Supervision

Function: All modules with this prefix are part of contents supervision. The diagnosis information for this component is in *MVS/ESA Component Diagnosis: Contents Supervision.*

CSVLOAD – LOAD SVC Service Routine

Function: Loads a usable copy of a requested module into virtual storage if a usable copy is not already available in virtual storage. LOAD does not pass control to the requested module.

Entry-Point CSVLOAD or IGC008 (alias):

Callers: Users of the LOAD SVC service routine.

Exit-Normal: Transfer control to EXIT prolog (IEAVEEXP) via a branch entry.

Exit-Error: Transfer control to EXIT prolog (IEAVEEXP) via a branch entry. An ERRET address specified by the ERRET operand receives control after the return to the caller. Control is not transferred directly to the ERRET address by this module, but is handled by the LOAD macro expansion.

Exit-Error: ABEND SVC via CSVABEND.

External-References: CSVABEND, CSVS2GWK, CSVFRR, CSVSBLDF.

CSVMEM – Address Space Termination Resource Manager

Function: During address space termination, CSVMEM frees the common service area (CSA) storage represented by the list of global extent blocks.

Entry-Point CSVMEM:

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

External-References: None.

CSVRBBLD – RB Build Routine

Function: This module obtains and initializes the program request block (PRB) and the extended status block (XSB), for a LINK, ATTACH, or XCTL request.

Entry-Point CSVRBBLD or CSVRBPRB (alias): Builds and enqueues a PRB and XSB for a LINK or an ATTACH request.

Callers: CSVATTCH, CSVLINK, CSVXLINK.

Exit-Normal: Returns to the caller.

Exit-Error: Returns to the caller, but flags are set in the SVRB extended save area to indicate that an ABEND should occur.

Entry-Point CSVRBXRB: Builds and enqueues a PRB and an XSB for an XCTL request where the requestor is not running under an SVRB.

Callers: CSVXCTL, CSVXXCTL.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: None.

CSVRBENQ – RB Enqueue Routine

Function: This module places a contents supervisor SVRB on a suspend queue when a request for a module cannot be immediately satisfied, and waits for the SVRB to be POSTed.

Entry-Point CSVRBENQ:

Callers: CSVSBRTN.

Exit-Normal: Returns to the caller.

Exit-Error: Returns to the caller, but flags are set in the SVRB extended save area to indicate that an ABEND should occur.

External-References: IEAVWAIT.

CSVRFREE – Error Return Resource Manager

Function: This routine cleans up resources held by a job step task so that control may be given to a user-defined error routine.

Entry-Point CSVRFREE:

Callers: CSVABEND.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: CSVS2CDQ, CSVS2RDQ.

CSVSBRTN – Contents Supervision Subroutines - Part 1

Function: CSVSBRTN contains a set of common subroutines that are used by contents supervision. These routines assist in module searching and in the maintenance of contents supervision control blocks.

Entry-Point CSVSBRTN or CSVSBLDF (alias): On a LOAD request, CSVSBLDF searches for the requested module.

Callers: CSVLOAD, CSVXLOAD.

Exit-Normal: Returns to the caller.

Exit-Error: Returns to the caller, but flags are set in the SVRB extended save area to indicate that an ABEND should occur.

Entry-Point CSVSBMFD: For a LINK, XCTL, or an ATTACH request, CSVSBMFD searches for the requested module.

Callers: CSVATTCH, CSVLINK, CSVXCTL, CSVXLINK, CSVXXCTL.

Exit-Normal: Returns to the caller.

Exit-Error: Returns to the caller, but flags are set in the SVRB extended save area to indicate that an ABEND should occur.

Entry-Point CSVSBLDL: On a DELETE request, CSVSBLDL decreases the count in the LLE and, if the LLE count becomes zero, purges the LLE.

Callers: CSVDELET.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point CSVSBCHK: CSVSBCHK decreases the CDE use count and, when necessary, frees the CDE and its associated resources: extent list, global extent list, minor CDEs, and the module itself. CSVSBCHK is called on a DELETE request or when a module that had been in use becomes available.

Callers: CSVARMGR, CSVEXIT.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References:

- CSVRBENQ
- CSVS2LIB
- CSVGETMD
- POSTRTN
- IEAVTPVD

CSVRBENQ, CSVS2LIB, CSVGETMD, IEAVTPVD, POSTRTN.

CSVSRCH – CDE and LPDE Search Routines

Function: This module contains CDE queue and link pack directory search routines that can be called by services other than contents supervision.

Entry-Point IEAQCDSR: Scans the contents directory for an entry representing the module name requested by the caller.

Callers: This routine is available to any caller.

Exit-Normal: If the module name is not found, returns to the caller 4 bytes beyond the return address indicated by register 14.

Exit-Error: No exit error conditions.

Entry-Point IEAVVMSR: Scans the link pack directory for an entry representing the module name requested by the caller.

Callers: This routine is available to any caller.

Exit-Normal: If the module name is not found, returns to the caller 4 bytes beyond the return address indicated by register 14.

Exit-Error: No exit error conditions.

External-References: None.

CSVSYNCH – SYNCH SVC Service Routine

Function: SYNCH allows a program to take a synchronous exit to another program, giving a supervisor state routine the ability to pass control to a problem state problem and receive control back in supervisor state.

Entry-Point CSVSYNCH or IGC012 (alias): The SYNCH SVC service has several uses:

1. Provides a supervisor routine the ability to take a synchronous exit to a user subroutine.
2. Avoids using LINK, which eliminates overhead, since no searching is performed for the requested module.
3. Allows a supervisor routine the ability to regain control in supervisor state after the user exit routine completes in problem state mode.

4. Allows the STAE and ESTAE macros the ability to schedule a synchronous exit to a routine in a privileged mode.

   Callers: Users of the SYNCH service routine.

   Exit-Normal: Transfer control to EXIT prolog via a branch entry.

   Exit-Error: ABEND SVC via CSVABEND.

External-References:

- CSVABEND
- SPIERTN
- IEAVTSSX.

CSVS2RTN – Contents Supervision Subroutines - Part 2

Function: This module contains common subroutines used by contents supervision.

Entry-Point CSVS2RTN or CSVS2LIB (alias): CSVS2LIB determines which library is to be searched first when attempting to locate the module. If either SYS1.LINKLIB or SYS1.SVCLIB is to be searched, CSVS2LIB calls the internal routine CSVS2LPA in order to search the link pack area.

Callers: CSVSBRTN.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point CSVS2CDQ: CSVS2CDQ searches the job pack area queue for a requested CDE and, when found, dequeues the CDE.

Callers: CSVGETMD, CSVRFREE.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point CSVS2GWK: Obtains and initializes a fetch work area and copies, if present, the user-supplied directory entry into the fetch work area.

Callers: CSVGETMD, CSVLINK, CSVLOAD, CSVXCTL, CSVXLINK, CSVXLOAD, CSVXXCTL.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point CSVS2LSR: CSVS2LSR serves as an external entry-point for the CSVS2LPA subroutine, which is used to search the link pack area.

Callers: CSVGETMD, CSVXCTL, CSVXXCTL.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point CSVS2RDQ: Dequeues RDs that are waiting on a module.

Callers: CSVGETMD, CSVRFREE.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IEAOPT02.

CSVVFCRE – Virtual Fetch Service Initialization Routine

Function: Sets up a cross memory environment that allows a user of virtual fetch, executing in another address space, to access the hash table in the virtual fetch service address space. Builds and refreshes a hash table of directory entries for all of the modules to be placed in the virtual fetch VIO data set. Builds and refreshes the virtual fetch VIO data set, which contains the reformatted load modules for all of the modules in the virtual fetch libraries.

Entry-Point CSVVFCRE:

Callers: IEEPRWI2 (Started task control).

Exit-Normal: Cancelled by operator (S222 ABEND). (Note: CSVVFCRE is a never-ending task operating in the virtual fetch address space. CSVVFCRE executes until cancelled by the operator.)

Exit-Error: Expected errors, returns to the caller.

Exit-Error: RTM.

Entry-Point CSVVFCES: ESTAE covering CSVVFCRE.

Callers: IEAVTRTM (RTM).

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point CSVVFCFR: FRR covering CSVVFCES while it holds the LOCAL lock.

Callers: IEAVTRTM (RTM).

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References:

- CSVVFCR1
- IARXVIO
- ILRGOS
- IARXWVFC

CSVVFCR1 – Virtual Fetch Load Module Formatter

Function: This module reformats a load module and its directory entry (DE) so that virtual fetch in LINKLST lookaside (LLA) can subsequently relocate the module.

Entry-Point CSVVFCR1:

Callers: CSVVFCRE and LINKLST lookaside module staging.

Exit-Normal: Returns to the caller.

Exit-Error::

- I/O errors and expected errors are trapped and reflected in the return code.
- Other errors are trapped by the caller's recovery routine.

External-References: None.

CSVVFGLU – Virtual Fetch Addressing Mode Interface Routine

Function: This module is the interface routine that handles addressing mode changes for virtual fetch. Virtual fetch runs in 31-bit addressing mode. Virtual fetch's caller can be in either 24-bit or 31-bit addressing mode. The called routine may run in either 24-bit or 31-bit addressing mode but must be invoked in the addressing mode of the caller of virtual fetch as if the caller were invoking the module.

Entry-Point CSVVFIND: Saves the addressing mode of the caller of the BUILD or FIND function of virtual fetch and passes control to CSVVFNDE in 31-bit addressing mode.

Callers: Users of virtual fetch's BUILD and FIND functions.

Exit-Normal: BSM to CSVVFNDE.

Exit-Error: No exit error conditions.

Entry-Point CSVVFGET: Saves the addressing mode of the caller of the GET function of virtual fetch and passes control to CSVVFGTE in 31-bit addressing mode.

Callers: Users of virtual fetch's GET function.

Exit-Normal: BSM to CSVVFGTE.

Exit-Error: No exit error conditions.

Entry-Point CSVVFSNC: Issues a SYNCH to a requested module. The SYNCH is performed in the addressing mode of the caller of CSVVFGET from below the 16-megabyte virtual storage line. If necessary, switches back to 31-bit addressing mode on return to CSVVFGTE.

Callers: CSVVFGTE.

Exit-Normal: BSM to CSVVFGTE.

Exit-Error: No exit error conditions.

External-References: CSVVFGTE, CSVVFNDE.

CSVVFGTE – Virtual Fetch Module GET Routine

Function: CSVVFGTE processes virtual fetch GET requests. That is, CSVVFGTE obtains and executes modules managed by virtual fetch when they are requested by the user of virtual fetch.

Entry-Point CSVVFGTE: Entry-point to process GET requests in 31-bit addressing mode.

Callers: CSVVFGET (an entry-point in module CSVVFGLU).

Exit-Normal: Passes control to the caller of CSVVFGET (an entry-point in module CSVVFGLU) or, if the module invoked for the caller abnormally terminated, to RTM.

Exit-Error: To RTM.

Entry-Point CSVVFEP2: Authorized section of CSVVFGTE to process GET requests.

Callers: CSVVFORK (a subroutine in module CSVVFNDE).

Exit-Normal: PT to label CSVVFRT2 for return to the caller of CSVVFGET in the caller's state, key and addressing mode.

Exit-Error: To RTM.

Entry-Point CSVVFGES: To perform ESTAE recovery for CSVVFGTE and CSVVFTCH.

Callers: RTM.

Exit-Normal: Returns to RTM requesting a retry to label RTRYSTUB in module CSVVFGTE with register one pointing to the LWK or requesting that RTM percolate the error.

Exit-Error: To RTM.

Entry-Point CSVVFRR: To perform FRR recovery for the JPA queue dequeuing and CSVVFRM routines within the ESTAE routine (CSVVFGES).

Callers: RTM.

Exit-Normal: Returns to RTM requesting a retry to label RTRYFRR1 or to label RTRYFRR2 in CSVVFGES with register one pointing to the LWK, or requesting that RTM percolate the error.

Exit-Error: To RTM.

Entry-Point CSVVFRM: Virtual fetch's routine to free the module and the VCBs storage associated with a VFWK.

Callers: CSVVFEP1 (an entry-point in module CSVVFNDE), CSVVFEP2 (an entry-point in module CSVVFGTE), CSVVFGES (an entry-point in module CSVVFGTE), CSVVFTCH.

Exit-Normal: Returns to the caller.

Exit-Error: To RTM.

External-References: CSVVFORK, CSVVFSNC, CSVVFTCH.

CSVVFMEM – Virtual Fetch Task and Memory Termination Manager Routine

Function: During address space termination for the virtual fetch service, CSVVFMEM marks the virtual fetch control block (VFCB) &q1.not active&eq1.. During any job step task termination, CSVVFMEM clears the pointer to virtual fetch's local control blocks (ASXBVFVT) and releases any resources owned by the terminating task.

Entry-Point CSVVFMEM: Performs recovery for user modules that abended under the terminating sub-task.

Callers: RTM (IEAVTMMT).

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point CSVVFMFR:

Callers: RTM

Exit-Normal: Returns to RTM requesting a retry to label RTRYDFRR.

Exit-Error: To RTM.

External-References:

- CSVVFRM
- IEAVTPVD
- IEAVTPVR

CSVVFNDE – Virtual Fetch BUILD and FIND Routine

Function: This module processes BUILD and FIND requests as follows:

1. For BUILD requests, CSVVFNDE creates a virtual fetch work area (VFWK) for the requested module indicating that virtual fetch is managing this module. On the first BUILD request, CSVVFNDE also creates a virtual fetch vector table (VFVT) for the user.
2. For FIND requests, CSVVFNDE finds the VFWK for a module and determines whether it can be used for a virtual fetch GET request.

Entry-Point CSVVFNDE: Module entry-point to process BUILD and FIND requests in 31-bit addressing mode.

Callers: CSVVFIND (entry-point in CSVVFGLU).

Exit-Normal: Returns to the caller of CSVVFIND in CSVVFIND's caller's addressing mode.

Exit-Error: To RTM.

Entry-Point CSVVFORK: Changes callers to supervisor state and dispatches callers to the authorized sections of CSVVFNDE and CSVVFGTE as appropriate.

Callers: CSVVFGTE, CSVVFNDE.

Exit-Normal: Branch to CSVVFEP1 in CSVVFNDE if CSVVFORK is invoked from CSVVFNDE or branch to CSVVFEP2 in CSVVFGTE if CSVVFORK is invoked from CSVVFGTE.

Exit-Error: If invoked from other than CSVVFNDE or CSVVFGTE, the caller receives control at the instruction following the invoking PC in the same state as the caller was in when the caller executed the PC instruction.

Entry-Point CSVVFEP1: Authorized section of CSVVFNDE to process BUILD and FIND requests.

Callers: CSVVFORK (entry-point in CSVVFNDE).

Exit-Normal: Control is passed to label CSVVFRT1 via a PT instruction which restores the caller's state. At CSVVFRT1, CSVVFNDE returns to the caller of CSVVFIND in the caller's addressing mode.

Exit-Error: To RTM.

Entry-Point CSVVFFES: ESTAE established by CSVVFNDE to cover CSVVFNDE and CSVVFSCH.

Callers: RTM.

Exit-Normal: Control is returned to RTM requesting a retry to label RTRYSTUB in module CSVVFNDE with register 1 pointing to the LWK. If retry is not requested, RTM continues with task termination processing.

Exit-Error: To RTM.

External-References:

- CSVVFRM
- CSVVFSCH.
- IEAVAPVD

CSVVFRSH – Virtual Fetch Address Space Refresh Routine

Function: CSVVFRSH issues a cross memory POST to the event control block (ECB) in the virtual fetch control block to request that refresh processing begin. Refresh processing consists of CSVVFCRE rebuilding the VIO data set and its hashed directory.

Entry-Point CSVVFRSH:

Callers: Initiator.

Exit-Normal: Returns to the caller.

Exit-Error: Returns from the POST error exit routine.

External-References: None.

CSVVFSCH – Virtual Fetch Hash Table Search Routine

Function: This module copies data from the virtual fetch directory entry (VFDE) of the named module to the address space of the caller.

Entry-Point CSVVFSCH:

Callers: CSVVFNDE.

Exit-Normal: PT back to the caller.

Exit-Error: The caller's recovery routine handles unexpected errors. RTM ensures that the PCLINK stack element created by CSVVFSCH is removed and the CML lock of the virtual fetch address space is released.

External-References: None.

CSVVFTCH – Virtual Fetch Module-Obtain Routine

Function: This module obtains a fresh copy of the requested module from virtual fetch's VIO data set and relocates the module's address constants (ADCONs).

Entry-Point CSVVFTCH:

Callers: CSVVFGTE.

Exit-Normal: Returns to the caller.

Exit-Error: Unexpected errors are trapped by the caller's recovery routine.

External-References:

- IARXVFA
- CSVVFRM

CSVXCTL – XCTL SVC Service Routine

Function: XCTL obtains a usable copy of a requested module and builds the control blocks necessary for the dispatcher to pass control to that module. Control is not returned to the issuer of the XCTL macro.

Entry-Point CSVXCTL or IGC007 (alias):

Callers: Users of the XCTL SVC Service.

Exit-Normal: Transfer control to EXIT prolog (IEAVEEXP) via a branch entry.

Exit-Error: ABEND SVC via CSVABEND.

External-References:

- CSVABEND
- CSVS2LSR
- CSVS2GWK
- CSVRBXRB
- CSVSBMFD
- SCBPURGE
- IEAVEXP3

CSVXLINK – Extended LINK SVC Service Routine

Function: CSVXLINK obtains a usable copy of a module and builds the control blocks necessary for the dispatcher to pass control to the module. After the requested program executes, control is returned to the issuer of the LINK macro.

Entry-Point CSVXLINK Or IGC0X6 (alias):

Callers: None.

Exit-Normal: Transfers control to EXIT prolog via a branch entry.

Exit-Error: Transfers control to EXIT prolog via a branch entry. Eventually, a user-supplied ERRET exit will receive control.

Exit-Error: ABEND SVC via CSVABEND.

External-References: CSVABEND, CSVFRR, CSVS2GWK, CSVSBMFD, CSVRBPRB.

CSVXLOAD – Extended LOAD SVC Service Routine

Function: Loads a usable copy of a requested module into virtual storage if a usable copy is not already present. LOAD does not pass control to the requested module.

Entry-Point CSVXLOAD or IGC0X8 (alias):

Callers: Users of the extended LOAD SVC service routine.

Exit-Normal: Transfers control to EXIT prolog (IEAVEEXP) via a branch entry.

Exit-Error: Transfers control to EXIT prolog (IEAVEEXP) via a branch entry. An ERRET address specified by the ERRET operand receives control after the return to the caller. CSVXLOAD does not transfer control directly to the ERRET address, for this is done by the LOAD macro expansion.

Exit-Error: ABEND SVC via CSVABEND.

External-References: CSVABEND, CSVFRR, CSVS2GWK, CSVGEXLD, CSVSBLDF.

CSVXXCTL – Extended XCTL SVC Service Routine

Function: Obtains a usable copy of a requested module and builds the control blocks necessary for the dispatcher to pass control to that module. Control is not returned to the issuer of the XCTL macro.

Entry-Point CSVXXCTL or IGC0X7 (alias):

Callers: Issuers of an XCTL SVC.

Exit-Normal: Transfers control to EXIT prolog (IEAVEEXP) via a branch entry.

Exit-Error: ABEND SVC via CSVABEND.

External-References:

- CSVABEND
- CSVS2LSR
- CSVS2GWK
- CSVRBXRB
- CSVSBMFD
- SCBPURGE
- IEAVEXP3

FSIREQ – JES Functional Subsystem Interface Request Macro Instruction

Function: FSIREQ invokes JES functional subsystem interface (FSI) services.

Entry From JES2, any functional subsystem (FSS)

Exit-Normal: Returns to the caller.

External-References: None.

IAR – Real Storage Manager

Function: All modules and macros with this prefix are part of the real storage manager component. The diagnosis information for this component is in *MVS/ESA Component Diagnosis: Real Storage Manager*.

IEAFTRT2 – ESA-SDWA Format Module

Function: This module formats the SVRB's extended save area's bits (via entry point IEAFTESA), that contain information indicating reasons for entry to RTM1 or RTM2 and formats the registers saved in the SDWA at the time of the error (via entry point IEAFSDW). IEAFTRT2 sends a return code to its caller indicating success or failure in formatting.

Entry from: IEAVTFMT.

Exit to: Caller.

IEAIHSAF — IHSA Control Block Format Module

Function: This module formats the IHSA (using IEAIHSAF's formatting model CSECT IEAISAFP), formats the FRR stack (using IEAVTRF3's formatting model CSECT IEAVTRP3), formats RT1W (using IEAVTRF3's formatting model CSECT IEAVTRP1), and summarizes RT1W's bit flags. IEAIHSAF sends back a return code to its caller indicating failure or success in formatting.

Entry from: IEAVTFMT via BLSQROUT and BLSQCFMT

Exit to: Caller.

IEAIPL04 — VSM IPL Resource Initialization Module

Function: This module allocates the page frame table (PFT), the initial portion of the system queue, area (SQA, ESQA), and the extended local system queue area (ELSQA).

Callers: IEAIPL00

Exit-Normal: IEAIPL00

Exit-Error: Disabled wait state (via IPL SVC 1)

External-References:

- IEAIPL00 — Issues SVCS during NIP
- IGVVSFMT — VSM cell pool formatter
- IGVEAAQA (entry point in IGVAAQAT) — turn the AQAT allocation bits on

IEALIMIT — Default Region Routine

Function: This module sets the limit values for the private area below the 16-megabyte line.

Entry-Point IEALIMIT:

Callers: IGVLIMIT.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: None.

IEAMLT05 - Module Lists Table (MLT) for the Console Display devices

Function: IEAMLT05 is a load module that contains the list of nucleus device dependent module names and the list of Link Pack Area (LPA) device dependent module names 3277, 3278, 3279, 3284, and 3286.

IEAMLT22 — Module Lists Table (MLT) for Miscellaneous Communication Devices

Function: IEAMLT22 is a load module that contains the list of nucleus device dependent module names and the list of Link Pack Area (LPA) device dependent module names required for miscellaneous communication devices.

IEAMLT24 — Module Lists Table (MLT) for the CTC Device

Function: IEAMLT24 is a load module that contains the list of nucleus device dependent module names and the list of Link Pack Area (LPA) device dependent module names required for the CTC device. devices.

IEAMLT27 — Module Lists Table (MLT) for the 3704 and 3705 Devices

Function: IEAMLT27 is a load module that contains the list of nucleus device dependent module names and the list of Link Pack Area (LPA) device dependent module names required for the 3704 and 3705 devices.

IEAMLT33 — Module Lists Table (MLT) for the DUMMY Devices

Function: IEAMLT33 is a load module that contains the list of nucleus device dependent module names and the list of Link Pack Area (LPA) device dependent module names required for the DUMMY device.

IEASMFEX — SMF EXCP Counter Routine

Function: This module counts all EXCPs and device connect time and places them into the TCTIOT and the OUCB.

Entry from: I/O drivers.

Exit to: Caller.

Called routines: IFGDEBUR.

IEASMFSP – SMF Memterm Resource Manager

Function: This module cleans up SMF resources for an address space being terminated. If SMF is terminating, the SMF timer elements are dequeued and freed along with other SMF areas (SST, EXIT tables, ACT and storage for SMFPRMXX). For other terminating address spaces, the SMF timer queue element for that address space, if one exists, is dequeued and freed.

Entry from: RTM.

Exit to: Caller for reset.

Called routines: TMRDEQ (TQEDEQUE via BALR).

IEATLEXT – SMF Time Limit Extension

Function: This module is entered if a job, job step, or wait-time limit has expired. It interfaces with the SMF user's exit routine, IEFUTL. If the exit grants an extension, IEATLEXT extends the expired limit by the amount specified by the exit. If the exit does not grant an extension, IEATLEXT will ABEND the expired task with a code 322 for a job or job step limit expiration, or 522 for a wait-time limit expiration.

Entry from: Dispatcher (IEAVEDS0).

Exit to: Dispatcher (IEAVEDS0).

IEATTUSD

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Timer Supervision.*

IEAV1443 – Communications Task Device Support Processor for 1403 and 3211 Devices

Function: This routine is used as the output portion of composite consoles or as the output-only consoles.

Entry from: IEAVVCTR.

Exit-Normal: Returns to caller.

Exit-Error: Exits to IEAVSWCH to perform an automatic console switch.

Called routines: IEEMB885

IEAV2540 – Communications Task Console Support Processor

Function: This routine is used as the input portion of a composite console or as the output-only consoles.

Entry from: IEAVVCTR.

Exit-Normal: Returns to the caller.

Exit-Error: Exits to IEAVSWCH to perform an automatic console switch.

Called routines: IEEMB885

IEAVADFM – User Formatter CSECT

Function: This module contains no executable code, only installation formatter names.

IEAVADIN – SNAP Invoker CSECT

Function: Invokes SNAP formatter CSECTs based on the dump options requested.

Entry from: IEAVAD01.

Exit to: IEAVAD01.

IEAVADSM – Display Summary Dump Storage

Function: This module displays selected key areas of user storage. These include the contents of the active load module if PSW points to an active module, 2K of storage around the PSW, 2K of storage around the registers, and the PRB module if one is found.

Entry from: IEAVADIH.

Exit normal: To caller.

Exit error: To caller.

IEAVADSP – Dump Data Spaces Routine

Function: This routine displays the data space storage requested by the caller of SNAP/ABDUMP.

Callers: IEAVADIN - SNAP/ABDUMP routine.

Exit-Normal: Caller.

External References: IARQDSPD, IEAVAD71, IEQVADGL.

IEAVAD0A – Display SQA and LSQA

Function: This module displays the allocated SQA and LSQA.

Entry from: IEAVADIN via BALR.

Exit to: IEAVADIN.

IEAVAD0B – Display LPA, JPA, and Active SVC Modules

Function: This module conditionally displays those LPA/JPA modules that are on the active RB queue or load list of the task being dumped, and the active SVC modules for the task. This module also displays the error header information which contains the load module name and PSW address, and the error offset into the module if the PSW at the time of error points to an active load module.

Entry from: IEAVADIN via BALR.

Exit to: Caller.

IEAVAD0C – Supervisor Trace Formatting

Function: This module invokes module IEAVETFC, which displays the supervisor trace table.

Entry from: IEAVADIN via BALR.

Exit to: Caller.

IEAVAD0D – Display Allocated Space Within User Subpools

Function: This module displays allocated space in user subpools (0-127) and subpool 252.

Entry from: IEAVADIN via BALR.

Exit to: Caller.

IEAVAD0E – Display SWA

Function: This module displays allocated space in the SWA. This module is also called to display subpools 229 and 230 when LSQA is requested.

Entry from: IEAVADIN via BALR.

Exit to: Caller.

IEAVAD0F – Display SNAP Storage

Function: This module displays the storage represented by the list supplied by the caller. This module also adds headers to the storage list if supplied by the user.

Entry from: IEAVADIN via BALR.

Exit to: Caller.

IEAVAD00 – SVC 51

Function: SVC 51 is the top segment for two distinct dump functions, SNAP and SVC dump. The basic function of SVC 51 is to identify which function is requested and route control to the corresponding routine.

Caller: None

Exit-Normal: Return to caller.

Exit-Error: Abends caller.

External-References: TESTAUTH, ABEND, IEAVTSGL, IEAVAD01, IEAVTSGL, IEAVTSDM, IEAVTSDH, ICAVTSSE, IEAVTSDG, IEAVTSDL, IEAVTSDF, IEAVTSDC, IEAVTSD1, ADYPRED, IEAVTSDB, and IEAVTSDE.

IEAVAD01 – SNAP Mainline Processing Routine

Function: IEAVAD0 initializes buffers and checks the validity of the SNAP parameter list. It also calls IEAVADIN which routes control to the appropriate functional display routine.

Callers: IEAVADIN.

Entry from: IEAVAD00 via BALR.

Exit to: Return to caller.

IEAVAD02 – Header, PSW Formatting

Function: This module formats and prints the dump ID, job name, step name, time, CPUID (serial number), and completion code.

Entry from: IEAVADIN via BALR.

Exit to: Caller.

IEAVAD03 — Control Blocks I

Function: This module formats and prints the ASCB, TCB, RB, LLE, CDE, extent lists (XTLIST), and the TIOT for the task being dumped.

Entry from: IEAVADIN via BALR.

Exit to: Caller.

IEAVAD05 — Control Blocks II

Function: This module formats and displays the VSM storage map and the TCB summary.

Entry from: IEAVADIN via BALR.

Exit to: Caller.

IEAVAD07 — Display Save Areas, PCLINK Stacks

Function: This module provides the display of user save areas and PCLINK stacks.

Entry from: IEAVADIN via BALR.

Exit to: Caller.

IEAVAD08 — SNAP, TCAM, GTF, VSAM, Shared Resources Interface Module, and Installation Formatters

Function: This module is responsible for the interfaces to the SNAP, TCAM, GTF, VSAM, shared resources formatting routines, and installation format routines.

Entry from: IEAVADIN via BALR.

Exit to: IEAVADIN, IBGDSNAP via BALR for GRSQ, IDA0195A via BALR for VSAM, IGA0E05A via LOAD and BALR for TCAM, IGC0F05A via LOAD and BALR for GTF, IGC0905A via LOAD and BALR for shared resources. ISTRAFD1 via LOAD and BALR for VTAM, and installation format routines whose names appear in the format CSECT, IEAVADFM via BALR.

IEAVAD09 — Display Nucleus and PSA

Function: This module displays the control program nucleus and PSA.

Entry from: IEAVADIN via BALR.

Exit to: Caller.

IEAVAD10 — Display PSW and Registers

Function: This module conditionally formats and prints the PSW, ILC, and interruption code, and displays the registers at entry to SNAP or ABEND.

Entry from: IEAVADIN via BALR.

Exit to: Caller.

IEAVAD11 — SNAP Output Routines

Function: This module writes the lines of the dump on the output device.

> Entry-point IEAVAD21 forces output of any records remaining in buffer after the dump is complete.
>
> Entry-point IEAVAD81 is the interface between the subsystem formatting routines and the output module IEAVAD11.

Entry from: IEAVADIN via BALR.

Exit to: Caller.

IEAVAD31 — SNAP Format and Format 01 Routines

Function: This module unpacks data into the output line while it associates a label with each piece of data.

> Entry-point IEAVAD31 converts the data to decimal notation and unpacks it into the output line associating labels with the fields. It also allows an indentation factor to the location in an output line where the data is to be placed.
>
> Entry-point IEAVAD41 performs the same functions as IEAVAD31 except that no indentation factor is anticipated in the layout line.

Entry from: Caller via BALR.

Exit to: Caller.

Error exit RTM via on X'0C4' ABEND after an unexpected UPR occurs.

IEAVAD51 — SNAP Format 20 and Format 22 Routines

Function: Format 20 translates the line into printable characters and unpacks four bytes of data into the beginning of the line. Format 22 unpacks up to four bytes of data into any positions of the line.

Entry from:

For entry-point IEAVAD51 — format 20 routine: IEAVAD71 via BALR.

For entry-point IEAVAD61 — format 22 routine: IEAVAD71 via BALR.

Exit to: IEAVAD71.

Error exit IEAVAD71 with register 15 set to 8.

IEAVAD71 — SNAP Format Router

Function: This routine controls the routing of other format routines to produce the print line that consists of the address of the data, the hex representation, and the EBCDIC translation of the data at that location.

Entry from: IEAVADSM, IEAVAD03, IEAVAD0B, IEAVAD0D via BALR.

Exit to: Caller.

Error exit Caller with a return code of eight if the space for a save area is not available, or RTM with an X'0C4ehex. ABEND if an unexpected UPR occurs.

IEAVALCR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines.*

IEAVAR00 — RCT Initialization/Termination

Function: This module performs address space initialization functions for a new address space and attaches the dump task and started task control. When a task terminates, this module frees address space resources before termination and detaches the dump task and the started task control task.

Entry from:

For entry-point IEAVAR00: IEAVEMIN.
For entry-point IEAVAER0: IEAVTAS1.

Exit to: Caller.

IEAVAR01 — RCT Common Processing

Function: This modules routes control within the RCT modules to wait for a functional request in the quiesce module and passes control to other routines to perform the requested functions. These functions are performed until a termination request is received, this module then returns control.

Entry from:

For entry-point IEAVAR01: IEAVAR00.
For entry-point IEAVARIA: IEAVTRT2.

Exit to: Caller.

IEAVAR02 — Quiesce Routine

Function: This module waits for a functional request to be posted and determines the function requested. If posted by SRM, this module either prepares an address for swap-out, or verifies that all tasks under the RCT are in long wait prior to returning. If a swap-out is possible, this module breaks active addressing binds to the address space and prohibits new ones. If posted for a termination or attention exit request, this module returns control.

Entry from:

For entry-point IEAVAR02: IEAVAR01.
For entry-point IEAVAFR2: IEAVTRTS.

Exit to: Caller.

Called routines: IEAVEBBR, IEAVESPM, IARSOUT, IEAVWAIT.

IEAVAR03 — Restore Routine

Function: This module either prepares an address space for execution after swap-in or handles quiesce backout processing.

Entry from:

For entry-point IEAVAR03: IEAVAR01.
For entry-point IEAVAR3A: IEAVTAS1.
For entry-point IEAVAFR3: IEAVTRTS.

Exit to:

For entry-points IEAVAR03 and IEAVAR3A: IEAVAR01.

For entry-point IEAVAFR3: IEAVTRTS.

IEAVAR04 — Attention Exit Scheduler Routine

Function: This module determines the attention level to be scheduled and schedules the associated attention exit.

Entry from:

For entry-point IEAVAR04: IEAVAR01.
For entry-point IEAVAFR4: IEAVTRTS.

Exit to: Caller.

IEAVAR05 – Attention Exit Prolog Routine

Function: This module acts as a front end of the user attention exit. It issues TPUT or TGET to the terminal, creates and initializes the TAIE, and goes to the attention exit in the user's key and state.

Entry from:

For entry-point IEAVAR05: IEAVEDS0.
For entry-point IEAVART5: IEAVTRT2.
For entry-point IEAVAFR5: IEAVTRTS.

Exit to:

For entry-points IEAVAR05 and IEAVART5: User attention exit or IEAVEOR.

For entry-point IEAVAFR5: Caller.

IEAVAR06 – Attention Exit Epilog Routine

Function: This module acts as a resource manager for the TAXE queue. It frees the TAIE, restarts the subtasks of the TCB for which the attention exit has completed, performs housekeeping on the TAXE queue, and updates the available TAXE count.

Entry from:

For entry-point IEAVAR06: IEAVEOR.
For entry-point IEAVAFR6: IEAVTRTS.

Exit to: Caller.

IEAVAR07 – Attention Exit Purge Routine

Function: This module acts as a resource manager for the TAXE queue. It is entered at task termination time so that inactive TAXEs for the terminating TCB can be dequeued. In addition to the TAXE queue housekeeping, this module updates the available TAXE count.

Entry from:

For entry-point IEAVAR07: IEAVTSKT.
For entry-point IEAVAFR7: IEAVTRTS.

Exit to: Caller.

IEAVAR08 – RCT Real Address Conversion Routine

Function: IEAVAR08 is called during swap-in of an address space and runs in the master address space. IEAVAR08 sets the correct values in the real address designation fields in the ASSB, ASTE, DUCTs and STCBs of the address space being swapped in.

Caller RSM

Exit to: Caller.

IEAVAX00 – STAX SVC Service Routine

Function: This routine creates, maintains, and releases terminal attention exit elements (TAXEs) which contain information for scheduling terminal attention exits. These TAXEs are queued from the Region Control Task Data Area (RCTD) as they are created.

Entry from:

For entry-point IGC0009F: IEAVESVC.
For entry-point STXFRR: IEAVTRTS.

Exit to:

For entry-point IGC0009F: IEAVEOR.
For entry-point STXFRR: Caller.

IEAVBLDP – Build Quickcell Pool Routine

Function: This module builds, extends, or reformats a quickcell pool.

Entry-Point IEAVBLDP:

Callers: Users of the BLDCPOOL macro.

Exit-Normal: BLDERR00 returns to the caller.

Exit-Normal: BLDERR08 indicates that the CPID is invalid.

Exit-Normal: BLDERR12 indicates that the requested subpool is invalid.

Exit-Normal: BLDERR16 indicates that the cell size is invalid.

Exit-Normal: BDLERR20 indicates that there are conflicting requests operating simultaneously on this extent.

Exit-Error: No exit error conditions.

External-References: None.

IEAVCKRS – Checkpoint/Restart Exit Routine Router

Function: This module invokes exit routines that perform MVS control program processing for checkpoint/restart.

Entry from: Checkpoint/Restart

Exit to: Checkpoint/Restart

IEAVCRVF — Vector Checkpoint/Restart Exit Routine

Function: This module saves and restores a task's vector-related status during a checkpoint or restart operation.

Entry from: IEAVCKRS

Exit to: IEAVCKRS or, for recovery, RTM

IEAVC700 — Communications Task Command Queuer

Function: This module queues a command for execution by the communications task, ensuring that the command executes in the communications task's address space and under its TCB. After the command is processed, this module dequeues the command.

Entry from: IEAVMQWR, IEE6703D.

Exit to: Caller.

IEAVC701 — Comm Task Initialization Routine

Function: Issue commands to initialize various console functions.

Entry-Point: IEAVC701

Callers:

Attached by IEEMB860 during Master Scheduler region initialization.

Exit-Normal:

Return to caller via BR 14

Exit-Error: None

External References:

Routines:

- IEEMB811 - SET command processor
- IEAVC700 - Comm Task Command Queue Routine

IEAVDELP — Delete Quickcell Pool

Function: This module removes all or part of a pool of quickcells, either freeing the storage or enqueuing the storage to be freed by the user of the DELCPOOL macro.

Entry-Point IEAVDELP:

Callers: Users of the DELCPOOL macro.

Exit-Normal: DELEXIT0 returns to the caller.

Exit-Normal: DELERR04 indicates there was an error when the storage from a deleted extent was being freed.

Exit-Normal: DELERR08 indicates that a permanent pool was to be deleted.

Exit-Normal: DELERR12 indicates that either an extent on a pool that has no extents or an unformatted pool was to be deleted.

Exit-Normal: DELERR16 indicates that the CPID was either invalid or zero.

Exit-Normal: DELERR20 indicates that there are conflicting requests operating simultaneously on this extent.

Exit-Error: No exit error conditions.

External-References: None.

IEAVD20

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVD30

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVD31

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVD32

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVD33

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVD34

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVD35

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVD36

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVEAC0

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Task Management.*

IEAVEADV – Address Verifier

Function: This module investigates a piece of storage to determine:

1. Whether that storage is associated with a storage check.
2. Whether a page fault or segment exception will occur when the storage range is referenced.

For entry-point IGAVEADV: From various callers, via the CVTADV field of the CVT.

Exit to: Caller.

IEAVEAT0

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Task Management.*

IEAVEBBR

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVECBV – Control Block Verification

Function: This module determines whether the input address is a valid ASCB, SRB, or TCB. All entry points are available to nucleus resident routines via VCONs:

> IEAVECAS -- Perform tests to determine if the input address points to the current ASCB.
>
> IEAVEGAS -- Perform tests to determine if the input address points to an ASCB.
>
> IEAVESRB -- Perform tests to determine if the input address points to an SRB.
>
> IEAVETCB -- Perform tests to determine if the input address points to an TCB.

Exit to: Caller.

Called routine IEAVEADV (Address Verification Routine)

IEAVECH0

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Task Management.*

IEAVECMS

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVEDFT

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVEDR

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVEDSR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Dispatcher.*

IEAVEDS0

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Dispatcher.*

IEAVEED0

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Task Management.*

IEAVEEEP – IQE Purge Routine

Function: Removes the IQEs for the terminating task from the asynchronous exit queue.

Entry from: End-of-task processing or ABEND processing.

Exit to: Caller.

IEAVEEER

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVEEE0

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVEEE2

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVEES – Emergency Signal Second Level Interrupt Handler

Function: This module receives control from the external FLIH and gives control to the appropriate receiving routine whenever emergency signal interrupt occurs.

Entry from: IEAVEEXT by using a BALR 2, 10.

Exit to: Caller.

IEAVEEXP

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVEEXT

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVEE1R

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVEE2R

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVEE3R

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVEF00

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVEGLT – Task Management Addressing Mode Interface

Function: IEAVEGLT provides the interface that allows callers executing in 24-bit addressing mode to use two task management modules, IEAVSETS and IEAVESSI, which execute in 31-bit addressing mode. Instead of specifying an entry-point in IEAVSETS or IEAVESSI, the caller specifies a corresponding entry-point in IEAVEGLT. This entry-point in IEAVEGLT changes the addressing mode of the caller and enters IEAVSETS or IEAVESSI at the proper point to perform the required function.

Entry from:

For entry-point IGC07902: A system module executing in PSW key 0 and supervisor state to obtain a STATUS service.

For entry-point IEATRSCN: A system module executing in PSW key 0 and supervisor state to obtain a TCB tree scan.

For entry-point IEAVESAS: A user defined subsystem that issues the SSAF macro to place data in the SSAT.

For entry-point IEAVESAF: A user defined subsystem that issues the SSAFF macro to obtain data from the SSAT.

Exit to:

For entry-point IGC07902: IEAVSETS at entry-point IGC07905.

For entry-point IEATRSCN: IEAVSETS at entry-point IEATRSC1.

For entry-point IEAVESAS: IEAVESSI at entry-point IEAVESS1.

For entry-point IEAVESAF: IEAVESSI at entry-point IEAVESS2.

IEAVEGLU

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control and Dispatcher.*

IEAVEGR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control and Dispatcher.*

IEAVEINT

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVEIO

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVEJST

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control and Dispatcher.*

IEAVELCR — Low Storage Refresh

Function: This module examines and correctly sets constants and addresses in selected system control blocks for continued system execution. Any refreshed fields are logged in SYS1.LOGREC in the form: Control block name, field name, followed by the error data.

Entry from: IEAVESAR.

Exit To caller.

Called routine None.

IEAVELK

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVELKR

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVEMCR

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Command Processing.*

IEAVEMDL

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Command Processing.*

IEAVEMIN

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Command Processing.*

IEAVEMRQ

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Command Processing.*

IEAVEMS0

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control and Dispatcher.*

IEAVEMS3

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control and Dispatcher.*

IEAVEMS6

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control and Dispatcher.*

IEAVEMS7

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control and Dispatcher.*

IEAVEMS8

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control and Dispatcher.*

IEAVEMS9

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control and Dispatcher.*

IEAVENQ1 – Queue Manipulation Routine

Function: See ISGGNQDQ. (IEAVENQ1 is an alias entry-point for ISGGNQDQ.)

IEAVEOR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Task Management.*

IEAVEPC

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVEPCO

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVEPCR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVEPDR

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVEPD0

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVEPST

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Task Management.*

IEAVEQV0 – Queue Verifier

Function: The queue verifier (QV) verifies and corrects queue structures. It performs three basic functions:

- If the queue structure has been destroyed, QV reconstructs the queue from any available information.
- Once the queue structure has been verified or reconstructed, QV will remove any elements that contain bad data.
- Records all detected errors, and corrective actions taken in an output data area.

Entry from:

For entry-point IEAVEQV1: Supervisor recovery routines for single-threaded queue with header only (type 1).

For entry-point IEAVEQV2: Supervisor recovery routines for single-threaded queue with header and trailer (type 2).

For entry-point IEAVEQV3: Supervisor recovery routines for double-threaded queue (type 3).

Exit to: Caller.

IEAVEQV4

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVEQV5

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVERER – Restart IH Recovery Routine

Function: This module makes the restart FLIH recovery function operable.

Entry from: Super FRR (IEAVESPR) at entry point IEAVERER.

Exit to: RTM via ABEND.

IEAVERES – Restart Interrupt Handler

Function: This entry saves the status of the interrupted program and proceeds to restart FLIH extension (IEAVEREX).

Entry from: The restart new PSW.

Exit to: RTM via IEAVEREX or resume interrupted program.

Called routines: IEAVEREO, IEAVEREX.

IEAVEREX – Restart FLIH Extension

Function: This module processes the restart interrupt fielded by IEAVERES. Processing includes handling the operator-selected restart reason on processor consoles that support the restart reason function. Depending on the selected reason, this module:

- ABENDs the interrupted program after displaying job information to the operator for verification.
- Performs high-level system diagnosis and repair in the following areas:
  - system nondispatchability.
  - exhausted WTO buffer elements.
  - no active batch or TSO jobs.
  - I/O problems (via IECVRSTS).

Entry from: IEAVERES.

Exit to: IEAVERES to resume interrupted work, RTM if interrupted work is to be abended.

Called routines: IEEVDCCR, IEAVSETS, CMSET, SETLOCK.

Entry-point IEAVERXI (this entry-point is a non-executable text string to be displayed on processor consoles that support the restart text function. A pointer to this entry-point is contained in the CVT (CVTRESTX). IPL/NIP and VARY CPU, ONLINE interface with the console and pass this text.)

IEAVERI

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVERP

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVESAR – Supervisor Analysis Router

Function: Routes control to analysis/repair routines. It performs the following basic functions:

- Verifies system fields required for processing in this module.
- Verifies that the environment is appropriate for analysis repair processing (not in ACR on this processor and not a recursive entry).
- Sets up recovery that remains in place over this and all called routines.
- Calls analysis/repair routines:

  IEAVELCR - Low Storage Refresh

  IEAVELKR - Spin Lock Repair Routine

  IEAVEVRR - ASVT/AFT Verification/Reconstruction Routine
- Initializes a pseudo SDWA into which any repair action taken by this or any called routine can be logged.
- Records any errors logged in the pseudo SDWA into the SYS1.LOGREC.

Entry from:

- IEAVTRTM on SLIH mode entry to RTM.
- IEAVEREX if the operator invokes Restart for system analysis and repair.

Exit to: Caller.

IEAVESC0

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVESCR – SCHEDULE Recovery Routine

Function: This module:

- Verifies the SRB journal queue and reschedules any remaining SRBs.
- Uses the queue verifier (IEAVEQV0) to verify the GSPL and LSPL queues for every address space in the system.

Entry from:

For entry-point IEAVESCR: Supervisor Control and Dispatcher recovery (IEAVEDSR) to clean up for the module (IEAVESC0).

For entry-point IEAVESQV: PURGEDQ FRR (IEAVEPDR) for queue verification.

Exit to:

For entry-point IEAVESCR: Supervisor Control and Dispatcher recovery.

For entry-point IEAVESQV: PURGEDQ FRR.

IEAVESGP

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVESLK

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVESLR – Suspend Lock Manager Repair and Lock Manager FRR Routines

Function: This module corrects correlations between the locking hierarchy mask and the suspend lock words.

Entry from:

For entry-point IEAVESLR: IEAVELKR, to verify and reconstruct suspend locks, the lock table, and the hierarchy mask.

For entry-point IEAVLKRR: Recovery routine for lock manager (IEAVELK and IEAVESLK) to determine if SETLOCK recovery is applicable and, if so, whether to retry or percolate.

Exit to:

For entry-point IEAVESLR: To caller (IEAVELKR).

For entry-point IEAVLKRR: To RTM.

IEAVESLX – Suspend Lock Analysis Exit for IPCS

Function: IEAVESLX examines a dump and creates a contention queue element (CQE) for the owner of and waiters for the LOCAL lock of each address space in the dump, CMS lock, CMS ENQ/DEQ lock, and CMS SMF lock.

Entry from: BLSQROUT. BLSQROUT is called by BLSACGTH which call all ANALYZE exits.

Exit to: Caller

External References BLSRESGU, BLSRESGE, BLSRESSA, BLSRACCQ, ADPLESRV.

IEAVESPI – SPIE/ESPIE Exit, SRB and PCFLIH Processors

Function: When entered by PCFLIH, the routine schedules the SPIE/ESPIE SRB, when appropriate. When entered by the SPIE/ESPIE SRB, the routine sets up the exit environment and passes control to it. Upon return, the routine cleans up the exit and schedules the previous exit if there is one.

Entry from:

For entry-point IEAVSPPF: PCFLIH via BALR.

For entry-point IEAVSPEX: SVC 3 via BALR.

For entry-point IEAVPSRB: An SRB via BALR.

For entry-point IEAVPSRR: IEAVPSRB via BALR.

For entry-point IEAVSPER: Contents supervisor's FRR, CSVFRR.

Exit to: Caller in 31-bit addressing mode.

IEAVESPR – Supervisor Control FRR (Super FRR)

Function: This module is a router routine for supervisor control. It determines retry and recovery addresses. In particular it can cause control to be routed to:

Module Name	Function
IEAVERER	Restart FLIH recovery
IEAVTRTF	RTM recovery
IEAVEPCR	Program FLIH recovery
IEAVEE1R	External FLIH 1 recovery
IEAVEE2R	External FLIH 2 recovery
IEAVEE3R	External FLIH 3 recovery
IEAVEIOR	I/O FLIH recovery
IEAVESVR	SVC FLIH recovery
IEA0DS (default)	Dispatcher mainline

Entry from:

For entry-point IEAVESPR: RTM for recovery control by means of an LPSW, if an error occurred while one of the supervisor control FRR stacks was in control.

For entry-point IEAVERTN: Mainline IEAVESPR.

Exit to: RTM, which will route control to one of the functions mentioned above.

IEAVESRT

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVESSI – SSAF Set/Obtain Routine

Function: This module sets a data word or obtains the data word from a specified entry in a TCB's subsystem affinity table (SSAT).

Entry from:

For entry-point IEAVESS1: Issuer of a SSAFF SET macro via IEAVEGLT.

For entry-point IEAVESS2: Issuer of a SSAFF OBTAIN macro via IEAVEGLT.

Exit to: Caller.

Error exit RTM via an ABEND macro.

Called routines: TCB treescan routine in IEAVSETS (IEATRSC1), IEAVELK.

IEAVESTS – SRB Status SAVE/RESTORE/MODIFY Routine

Function: IEAVESTS processes SRBSTAT macros. Depending on whether the macro specifies SAVE, RESTORE, or MODIFY, IEAVESTS:

- Saves the current status or an SRB in a specified SRB status save area (STSV). The saved status includes:
  - General purpose and floating point registers.
  - Processor and SRB timing information.
  - Processor affinity.
  - The purge ASID and the purge TCB address.
  - The PCLINK stack header.
  - Cross memory status.
  - The normal FRR stack.

  If the caller provided the address of an FRR routine, IEAVESTS also adds the FRR to the normal FRR stack (SRBSTAT SAVE).
- Restores status from values saved in a specified STSV (SRBSTAT RESTORE).
- Stores a new purge ASID and purge TCB address in either the STSV or the LCCA (SRBSTAT MODIFY).

Entry from:

For entry-point IEAVESTS: An SRBSTAT SAVE macro via a BALR 14, 15 instruction.

For entry-point IEAVESTR: An SRBSTAT RESTORE macro via a BALR 14, 15 instruction.

For entry-point IEAVESTM: An SRBSTAT MODIFY macro via a BALR 14, 15 instruction.

Exit to: Caller.

Error exit RTM via an ABEND macro.

IEAVESTU – Dynamic SVC Table Update Service

Function: The SVC Dynamic Update Service updates entries in the system SVC table for "REPLACE/DELETE" requests. The old SVC table entry and other data about the update request is stored in the SVC update recording table. It returns information from the SVC table for "EXTRACT" requests.

Entry from:

For entry-point IEAVESTU: A SVCUPDTE macro interface generates BASSM with register 1 pointing to the parameter list.

For entry-point IEAVESTE: BASSM pointing to the parameter list.

For entry-point IEAVESTR: RTM.

Exit to: Caller

External References IEAVENLU, IEAVVMSR.

IEAVESVC

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVETAC – System Trace Address Space Create Routine

Function: IEAVETAC initiates the creation of the system trace address space (TRACE) and the initialization of the trace function.

Callers: System Trace ALTRTRC Service Interface (IEAVETAT entry-point in module IEAVETSI).

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: Issues an ABEND macro to terminate the caller with a X'09E' completion code.

External-References: IEEMB881, IEAVETTV entry-point in IEAVETSI, IEAVETAI (invoked via address space creation processing), IEAVETRM.

IEAVETAI – System Trace Address Space and Function Initialization Routine

Function: IEAVETAI initializes the system trace address space (TRACE) and the trace function.

Callers: Started Task Control (IEEPRWI2).

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: Issue an ABEND macro to terminate the caller with a X'09E' completion code.

External-References: IEEMB883, IEAVETCV, IEAVETEA, IEAVETFX, IEAVETPA, IEAVETPS, IEAVETPV, IEAVETSD, IEAVETSN, IEAVETSP.

IEAVETCL

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVETCV – System Trace Control Block Verification

Function: IEAVETCV verifies and corrects the following system trace structures:

- The Trace Vector Table (TRVT).
- The Trace Option Block (TOB).
- Trace Table Snapshot Copy Header (TTCH) queue via the queue verifier (IEAVEQV0).
- The Trace Buffer Vector Table (TBVT) queue by executing System Trace Processor Verification Routine (IEAVETPV) for each processor.

Callers:

System Trace Verification SRB (IEAVETVS entry-point in IEAVETSI).

System Trace Verification (IEAVETTV entry-point in IEAVETSI).

Exit-Normal: Returns to the caller via PT.

Exit-Error: ABEND X'09' to FRR IEAVETFR (Entry-point in IEAVETRR).

External-References: IEAVETPV, IEAVETRR entry-point in IEAVETFR, IEAVEQV3.

IEAVETEA – System Trace Environment Alteration Routine

Function: IEAVETEA provides the ALTRTRC environment alteration service, which initializes and maintains the system trace structure and state.

Callers: System Trace ALTRTRC Service Interface (IEAVETAT entry-point of IEAVETSI).

Exit-Normal: Returns to the caller via PT.

Exit-Error: Issues an ABEND macro to terminate the caller with a X'09E' completion code; or issue an ABEND macro to enter the FRR IEAVETFR (entry-point in IEAVETRR) with a X'09E' completion code.

External-References: IEAVETFR entry-point of IEAVETRR, IEAVETPA, IEAVETRR, IEAVETTV entry-point of IEAVETSI, IEAVRT02.

IEAVETEF – System Trace Table Entry Filter/Formatter

Function: IEAVETEF is part of the system trace formatter. IEAVETEF filters out (skips) or formats the current trace table entry (TTE) passed in by the system trace table formatter controller (IEAVETFC). IEAVETEF sets the length of the just-filtered TTE (TFWALJFE) so that the formatter controller knows where the next TTE begins. IEAVETEF calculates the number of words in an explicit TTE so that individual formatting routines can determine the number of output lines the TTE will require.

Callers: IEAVETFC.

Exit-Normal: Returns to the caller.

Exit-Error: To RTM via an ABEND.

External-References: ITRFBR, ITRFDEFU, ITRFEXP, ITRFHEX, ITRFPC, ITRFPT, ITRFSSAR, ITRF0001, ITRF0003, ITRF0005, ITRF0007, ITRF000B, ITRF000F, ITRF0013, ITRF0015, ITRF0017, ITRF0019, ITRF001B, ITRF007F, ITRF017F, ITRF027F, ITRF037F, ITRF047F, ITRF057F, ITRF067F, ITRF077F, ITRF087F, ITRF097F, ITRF0A7F, ITRF0B7F, ITRF0C7F, ITRF0D7F, ITRF0E7F, ITRF0F7F.

IEAVETFA – System Trace Formatter Service Routines

Function: IEAVETFA contains two service routines used within the system trace formatter:

IEAVETFA entry point informs the caller whether a specified ASID has been selected for formatting. TTEs that are not formatted are said to be filtered out. IEAVETFA determines whether branch and address space TTEs following the current TTE will be formatted when the caller requests this information.

IEAVETPB entry-point prints an output buffer. IEAVETPB prevents trace table entries from crossing an output page boundary when called by Print Dump, and prints the Print Dump TRACE verb syntax error messages to the SYSPRINT file.

Callers: Individual format routines with names of the form ITRF0nnn or ITRF0n7F.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND.

Entry-Point IEAVETPB prints the trace output buffer as part of the trace table. When called by Print Dump, this routine keeps track of the relative line on the page and prints blank lines where necessary to avoid splitting a TTE across pages.

Callers: Any CSECT in load module IEAVETFC (except IEAVETFA).

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND.

External-References: SNAP/PRDMP print routine pointed to by ADPLPRNT.

IEAVETFC – System Trace Table Formatter Controller

Function: IEAVETFC is called by SNAP (IEAVADOC) or Print Dump (AMDPRUIM) to format the system trace table. It is called by SNAP if the TRT option is specified on the SNAP macro. It is called by Print Dump if the TRACE verb is supplied in the control input. IEAVETFC controls:

- The interpretation of format request options.
- The flow of trace data from the dump data set or trace address space into work buffers associated with each processor.
- The merge of trace table entries from the individual processor trace buffers into the formatted trace table in chronological sequence.
- The filtering of trace table entries according to the format request options.

Callers: AMDPRUIM (Print Dump) or IEAVADOC (SNAP).

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

External-References: IEAVETEF, IEAVETFA, IEAVETPB, IEAVETNP, IEAVETPW, IEAVETRW, IEAVETVP, ITRFHEX.

IEAVETIH – System Trace Interrupt Handler (SLIH)

Function: IEAVETIH handles a trace program interrupt. A trace interrupt is generated when no more trace table entries can be stored in the current trace buffer. IEAVETIH causes the oldest trace buffer to become the new current trace buffer.

Callers: DAT-off program FLIH (IEAVEPCO).

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To the program FLIH via a program check (a specification exception). FLIH enters RTM to begin recovery processing. Since the program FLIH is executing under a super FRR stack, the super FRR (IEAVESPR) will be executed. The super FRR calls system trace mainline recovery (IEAVETMR) to perform IEAVETIH recovery.

External-References: None.

IEAVETMR — System Trace Mainline Recovery Routine

Function: IEAVETMR performs system trace mainline recovery following an error in the creation of a trace table entry (TTE) (IEAVETRC or an inline PTRACE macro) or an error in trace interrupt processing (IEAVETIH). IEAVETMR stops program checks which occur because of a problem in the trace structure. If a problem is detected in the trace structure, system trace structure verification and restoration is initiated to correct it.

Callers: Super FRR (IEAVESPR).

Exit-Normal: Returns to the caller via BR 14.

External-References: IEAVETOT, IEAVETRC, Label IEAVETRD in IEAVETRC, Entry-point IEAVETTV in IEAVETSI, IEAVETVT.

IEAVETNP — System Trace Formatter Dump Data Initialization Routine

Function: IEAVETNP extracts the data necessary to obtain the system trace table from the trace address space or a dump data set.

Callers: System trace table formatter controller (IEAVETFC).

Exit-Normal: Returns to the caller.

Exit-Error: To RTM via an ABEND.

External-References: AMDPRDMP storage access routine (ACCESS, address is in the field ADPLMEMA), Entry-point IEAVETPB in module IEAVETFA.

IEAVETOB — System Trace Option Block Module

Function: IEAVETOB is the system trace option block (TOB), containing the system trace configuration and state information. It contains no executable code.

IEAVETOT — System Trace Operand Table Module

Function: IEAVETOT is the system trace operand table (TOT), containing the operands for the TRACE instruction. It contains no executable code.

IEAVETPA — System Trace Processor Alteration Routine

Function: IEAVETPA alters the trace environment for one processor.

Callers:

System Trace Environment Alteration Routine (IEAVETEA).

System Trace Receiving Routine (IEAVETRR entry-point of IEAVETRR).

Exit-Normal: Returns to the caller via BSM.

Exit-Error: Issues an ABEND macro to terminate the caller with a X'09E' completion code.

External-References: None.

IEAVETPS — System Trace Processor Snapshot Routine

Function: IEAVETPS snapshots the trace data for one processor.

Callers:

System Trace Table Snapshot Routine (IEAVETSN).

System Trace Receiving Routine (IEAVETRR entry-point of IEAVETRR).

Exit-Normal: Returns to the caller via BSM.

Exit-Error: Issues a ABEND macro to terminate the caller with a X'09E' completion code.

External-References: None.

IEAVETPV — System Trace Processor Verification Routine

Function: IEAVETPV verifies and corrects the system trace buffer vector table (TBVT) queue via the queue verifier (IEAVEQV0) and reestablishes other system trace fields for a target processor.

Callers:

System Trace Control Block Verification (IEAVETCV).

System Trace Receiving Routine (IEAVETRR).

Exit-Normal: Returns to the caller via BSM.

Exit-Error: ABEND X'09E' to FRR established by caller.

External-References: Queue Verifier routine (IEAVEQV0).

IEAVETPW – System Trace Formatter Prime Work Buffer Routine

Function: IEAVETPW primes the formatter work buffers from the system trace data in the current trace address space or a dump data set.

Callers: System trace table formatter controller (IEAVETFC).

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND.

External-References: AMDPRDMP storage access routine (ACCESS, address is in the field ADPLMEMA), Entry-point IEAVETPB in module IEAVETFA, IEAVETRW.

IEAVETRC – System Trace Table Entry Creation Routines

Function: IEAVETRC contains the routines that create system trace table entries. These routines set up the general registers and issue the TRACE instruction to create the following entries in the system trace table for the issuing processor:

- Subchannel trace table entries - Start, Modify, Halt, Clear, Resume.
- External Interrupt trace table entries - General External, EMS, SS, External Call, Clock Comparator.
- SVC Interrupt trace table entries - General SVC, SVC Return, SVC Error.
- Program Interrupt trace table entry.
- I/O Interrupt trace table entry.
- Task Dispatch trace table entries - Task, Initial SRB, Suspended SRB, WAIT.
- Machine Check Interrupt trace table entry.
- Restart Interrupt trace table entry.
- Alternate CPU Recovery trace table entry.
- Suspension trace table entry.
- Trace Options Alteration trace table entry.
- User Event trace table entry.

Entry-Point IEAVETSS creates Start Subchannel (SSÇH) entry.

Callers: IOS Start Subchannel Initiation Module (IOSVSSCH).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETMS creates Modify Subchannel (MSCH) entry.

Callers: IOS Modify Subchannel Initiation Routine (IOSVMSCH).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETHS creates Halt Subchannel (HSCH) entry.

Callers: IOS Halt/Clear Subchannel Routine (IOSVHSCH).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETCS creates Clear Subchannel (CSCH) entry.

Callers: IOS Halt/Clear Subchannel Routine (IOSVHSCH).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETRS creates Resume Subchannel (RSCH) entry.

Callers: IOS Resume I/O Service Routine (IOSVRSUM).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETEX creates External Interrupt (EXT, EMS, SS, CALL, CLKC) entries.

Callers: External FLIH (IEAVEEXT).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETSV creates General SVC Interrupt (SVC) entry.

Callers: SVC FLIH (IEAVESVC).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETSR creates SVC Return (SVCR) entry.

Callers: Exit Prolog (IEAVEEXP).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETSE creates SVC Error (SVCE) entry.

Callers: SVC FLIH (IEAVESVC).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETPI creates Program Interrupt (PGM) entry.

Callers: Program FLIH (IEAVEPC).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETIO creates I/O Interrupt (IO) entry.

Callers: I/O FLIH (IEAVEIO).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETDP creates Task Dispatch (DSP) entry.

Callers: Dispatcher (IEAVEDS0).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETIS creates Initial SRB Dispatch (SRB) entry.

Callers: Dispatcher (IEAVEDS0).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETSB creates Suspended SRB Dispatch (SSRB) entry.

Callers: Dispatcher (IEAVEDS0).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETWT creates WAIT Dispatch (WAIT) entry.

Callers: Dispatcher (IEAVEDS0).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETMH creates Machine Check Interrupt (MCH) entry.

Callers: MCH-RTM Interface (IGFPMRTM).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETRE creates Restart Interrupt (RST) entry.

Callers: Restart FLIH (IEAVERES).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETAR creates Alternate CPU Recovery (ACR) entry.

Callers: Alternate CPU Recovery (IEAVTACR).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETSU creates Lock Suspension (SUSP) entry.

Callers: Suspend Lock Manager Service Routine (IEAVESLK).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETTA creates Trace Options Alteration (ALTR) entry.

Callers: System Trace Environment Alteration Routine (IEAVETEA).

Exit-Normal: Returns to the caller via BSM 0, 14.

Entry-Point IEAVETUR creates User Event (USRn) entry.

Callers: Issuer of the PTRACE macro with TYPE=USRn.

Exit-Normal: Returns to the caller via BSM 0, 14.

Exit-Error: Issues an ABEND macro to terminate the caller with a X'09E' completion code.

External-References: IEAVETTV entry-point of IEAVETSI.

IEAVETRM — System Trace Task/Address Space Termination Resource Manager

Function: IEAVETRM performs the following resource manager functions:

- For task terminations, invokes the system trace table snapshot service (SNAPTRC) to free any trace table snapshot copy headers (TTCHs) for the task.
- For address space terminations other than the trace address space, invokes SNAPTRC service to free any TTCHs for the address space.

- For a trace address space termination, cleans up the global trace resources.

Callers:

Task termination (IEAVTSKT).

Memory termination purges (IEAVTMMT).

Memory termination controller (IEAVTMTC).

Exit-Normal: Returns to the caller via BR 14.

External-References: IEAVETTV entry-point in IEAVGTSI.

IEAVETRR – System Trace Receiving/DIE/FRR Routines

Function: IEAVETRR contains three system trace services:

- Entry-point IEAVETRR is an interface to the system trace services that run as PC routines in external SLIH mode:

  System trace processor alteration routine (IEAVETPA).
  System trace processor snapshot routine (IEAVETPS).
  System trace processor verification routine (IEAVETPV).

- Entry-point IEAVETTD is a timer disabled interrupt exit (DIE) that initiates system trace structure verification.
- Entry-point IEAVETFR is a general functional recovery routine (FRR) for the system trace services.

Entry-Point IEAVETRR: Interfaces routine to the system trace services that run as PC routines in external SLIH mode.

Callers: Emergency signal SLIH (IEAVEES).

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: Issues an ABEND macro to enter the FRR IEAVETFR (entry-point in IEAVETRR) with a completion code of X'09E'

Entry-Point IEAVETTD: Timer disabled interrupt exit routine to initiate trace structure verification, which cleans up the processor trace structure for a processor that has gone off-line.

Callers: Timer SLIH (IEAVRTI0).

Exit-Normal: Returns to the caller via BR 14.

Entry-Point IEAVETFR: General FRR for system trace services.

Callers: System recovery manager (IEAVTRTS).

Exit-Normal: Returns to the caller via BR 14.

External-References: IEAVETPA, IEAVETPS, IEAVETPV, IEAVETTV entry-point in IEAVSTSI.

IEAVETRW – System Trace Formatter Refill Work Buffer Routine

Function: IEAVETRW extracts a buffer of trace data from the trace address space or a dump data set and places it in a work buffer to be formatted.

Callers:

System trace table formatter controller (IEAVETFC),

System trace formatter prime work buffer routine (IEAVETPW).

Exit-Normal: Returns to the caller.

Exit-Error: To RTM via an ABEND.

External-References: AMDPRDMP storage access routine (ACCESS, address is in the field ADPLMEMA), Entry-point IEAVETPB in module IEAVETFA.

IEAVETSC – System Trace Storage Check Recovery Routine

Function: IEAVETSC performs machine check processing for system trace. If the machine check is a storage check, IEAVETSC checks if it affects the processor trace structure, removes this storage from use, and initiates asynchronous restoration of the trace structure.

Callers: MCH/RTM Interface (IGFPMRTM).

Exit-Normal: Returns to the caller via BR 14.

External-References: IEAVETTV entry-point in IEAVETSI.

IEAVETSD – System Trace Table Snapshot Data Extraction Routine

Function: IEAVETSD provides the COPYTRC service that extracts trace table snapshot data from the trace address space and copies it to the primary address space of the COPYTRC macro issuer.

Callers: System trace COPYTRC service interface (IEAVETCT entry-point in IEAVETSI).

Exit-Error: Issues an ABEND macro to terminate the caller with a X'09E' completion code.

Exit-Error: Issues an ABEND macro to enter the FRR IEAVETFR (Entry-point in IEAVETRR) with a X'09E' completion code.

External-References: None.

IEAVETSI – System Trace Service Interfaces

Function: IEAVETSI provides the interface to the following system trace services:

1. ALTRTRC service.
2. SNAPTRC service.
3. ASIDTRC service.
4. COPYTRC service.
5. VERFYTRC service.

IEAVETSI allows these system trace service requests to complete normally, even if the trace address space has terminated or is in the process of terminating.

Entry-Point IEAVETAT: Interface to the ALTRTRC service that creates the system trace address space or modifies the system trace environment.

Callers: IEAVETCV, IEAVNP51, IEECB925, IEEVCPR, IEEVWKUP, IGFPMRTM, IGFPTERM, IGFPTSIG.

Exit-Normal: Returns to the caller via BSM 14.

Exit-Error: Issues an ABEND macro to terminate the caller with a X'09E' completion code.

Entry-Point IEAVETST: Interface to the SNAPTRC service to snapshot the system trace table or to free trace table snapshots.

Callers: IEAVAD00, IEAVAD01, IEAVETRM, IEAVTABD, IEAVTRT2, IEAVTSDC, IEAVTSDR, IEAVTSDX.

Exit-Normal: Returns to the caller via BSM 14.

Entry-Point IEAVETAF: Interface to the ASIDTRC service to create a system trace table snapshot for an ASID.

Callers: IEAVTSDM.

Exit-Normal: Returns to the caller via BSM 14.

Entry-Point IEAVETCT: Interface to the COPYTRC service to copy data from a system trace table snapshot.

Callers: IEAVETNP, IEAVETPW, IEAVETRW, IEAVTSDM.

Exit-Normal: Returns to the caller via BSM 14.

Entry-Point IEAVETTV: Interface to the VERFYTRC service to initiate verification and restoration of the system trace structure.

Callers: IEAVETAC, IEAVETFR entry-point of IEAVETRR, IEAVETMR, IEAVETRC, IEAVETRM, IEAVETSC, IEAVETSI, IEAVETSN, IEAVETTD entry-point of IEAVETRR, IEECB924.

Exit-Normal: Returns to the caller via BSM 14.

Exit-Error: Issues an ABEND macro to terminate the caller with a X'09E' completion code.

Entry-Point IEAVETVS: Interface as an SRB routine to the VERFYTRC service to verify and restore the system trace structure or function as the Resource Manager Termination Routine (RMTR) for that SRB.

Callers: IEAVETTV entry-point of IEAVETSI, IEAVEPD0.

Exit-Normal: Returns to the caller via BSM 14.

External-References: IEAVETAC, IEAVETCV, IEAVETEA, IEAVETFR entry-point of IEAVETRR, IEAVETSP, IEAVETSD, IEAVETSN, IEAVETTF, IEAVETVT.

IEAVETSN – System Trace Table Snapshot Routine

Function: IEAVETSN provides the SNAPTRC service, which creates or frees system trace table snapshots.

Callers: System Trace SNAPTRC Service Interface (IEAVETST entry-point of IEAVETSI).

Exit-Normal: Returns to the caller via PT.

Exit-Error: Issues an ABEND macro to terminate the caller with a X'09E' completion code.

External-References: IEAVETFR entry-point of IEAVETRR, IEAVETPS, IEAVETRR, IEAVETTV entry-point of IEAVETSI.

IEAVETSP – System Trace ALTRTRC SUSPEND/RESUME/PSTART Routine

Function: IEAVETSP provides the ALTRTRC SUSPEND/RESUME/PSTART service, which suspends, resumes or starts system tracing on the current processor.

Callers: System Trace ALTRTRC Service Interface (IEAVETAT entry-point of IEAVETSI).

Exit-Normal: Returns to the caller via PT.

Exit-Error: Issues an ABEND macro to terminate the caller with a X'09E' completion code.

External-References: None.

IEAVETTF – System Trace Table Snapshot Filter Routine

Function: IEAVETTF performs the ASIDTRC service, receiving control from SYSMDUMP processing when the SYSMDUMP is being taken for an unauthorized user. IEAVETTF filters an input trace table snapshot copy header (TTCH), and creates a new TTCH containing only the trace data that the input address space ID (ASID) is authorized to read.

Entry-Point IEAVETTF:

Callers: System trace ASIDTRC service interface (IEAVETAF entry-point of IEAVETSI).

Exit-Normal: Returns to the caller via a PT.

Exit-Error: ABEND X'09E'.

External-References: None.

IEAVETVP – System Trace Print Dump TRACE Verb Parser

Function: IEAVETVP is part of the system trace formatter. IEAVETVP parses the Print Dump TRACE verb for the trace formatter controller (IEAVETFC) and sets the appropriate fields in the trace formatter work area (TFWA). If selection of trace entries by a specific ASID is requested via any of the ASID selection keywords (CURRENT, ASID, or JOBNAME), IEAVETVP builds a list of the ASIDs to be selected.

Callers: IEAVETFC.

Exit-Normal: Returns to the caller.

Exit-Error: To RTM via an ABEND SVC.

External-References: IEAVETFB, AMDPRJNA.

IEAVETVT – System Trace Vector Table Module

Function: IEAVETVT is the system trace vector table (TRVT), containing addresses of the system trace service routines and constants they require. It contains no executable code.

IEAVEVAL – Validity Check

Function: This module validates an address or address range by determining if the storage key and the protect key match.

Entry from: Supervisor routines via branch.

Exit to: Caller.

IEAVEVRR – Address Space Verification Processing Routine

Function: If necessary, IEAVEVRR refreshes the PSASTOR field, rebuilds the real address first table (AFT) from the virtual AFT, and checks for invalid entries in the ASVT. If any exist, it rebuilds the ASVT and requeues the two ASVT reserve queues.

Entry from: IEAVESAR, the supervisor analysis router.

Exit to: Caller

IEAVEVS – Vector Second Level Interrupt Handler

Function: This module creates and restores the vector environment for a task issuing vector instructions.

Entry from:

For entry point IEAVEVS - Program check FLIH (IEAVEPC)

For entry point IEAVVRTM - IEAVTRTM

For entry point IEAVVFGM - IEAVEVS or IEAVVSRB

Exit to:

For entry point IEAVEVS - Program check FLIH (IEAVEPC)

For entry point IEAVVRTM - IEAVTRTM

For entry point IEAVVFGM - caller

IEAVEVT0

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Task Management.*

IEAVEWAT — WAIT Service Routine

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Task Management.*

IEAVEXS — External Call Second Level Interrupt Handler

Function: Upon an external call interruption, this module receives control from the external FLIH. It routes control to the appropriate receiving routine.

Entry from: External FLIH via BALR 2, 10.

Exit to: Caller.

IEAVFRCL — FREECELL Routine

Function: This module returns a cell to a quickcell pool. It makes the cell available for use by adding it to a queue of available cells in the pool.

Entry-Point IEAVFRCL:

> Callers: Users of the FREECELL macro.
>
> Exit-Normal: FRCEXIT0 returns to the caller.
>
> Exit-Normal: FRCERR04 indicates that the cell was already freed.
>
> Exit-Normal: FRCERR08 indicates that the address of the cell was invalid.
>
> Exit-Normal: FRCERR12 indicates that the quickcell pool was unformatted.
>
> Exit-Normal: FRCERR16 indicates that the CPID is invalid.
>
> Exit-Normal: DEQUEIT indicates that the last extent was deleted after the cell was returned.
>
> Exit-Error: No exit error conditions.

External-References: None.

IEAVFRLK — Lock Freeing Routine

Function: Depending on whether the processor holds the dispatcher or the local lock, this module clears either the global or local intersect in the SVT. It then releases all locks held by the caller. If the current task is the target of an earlier status request that could not be completed until a CML lock was released, IEAVFRLK calls the STATUS routine (entry-point IGC07904 in IEAVSETS) to complete the pending status request.

Entry from: A caller in supervisor state, zero protect key.

Exit to: Caller.

IEAVFX00 — Prefix Storage Area

Function: Provide definition of data in the Prefix Storage Area (PSA) and provide the Daton-Datoff Linkage Routine.

IEAVGTCL — GETCELL Routine

Function: This module allocates a cell from an established quickcell pool.

Entry-Point IEAVGTCL:

> Callers: Users of the GETCELL macro.
>
> Exit-Normal: GTCERR00 returns to the caller.
>
> Exit-Normal: GTCERR04 indicates that the quickcell pool was empty.
>
> Exit-Normal: GTCERR08 indicates that the quickcell pointers to the available cells have been destroyed.
>
> Exit-Normal: GTCERR12 indicates that the quickcell pool was unformatted.
>
> Exit-Normal: GTCERR16 indicates that the CPID was not defined.
>
> Exit-Error: No exit error conditions.

IEAVG700 — Subsystem Console Service Routine

Function: Acts as a service routine that allows a subsystem to acquire or alter the use of a console.

Entry Point: IEAVG700

> Callers: None identified.
>
> Exit-Normal: Return to caller.
>
> Exit-Error: Return to caller with a return code set in register 15. The SCSR completion bytes indicate which functions have been completed.

Entry Point: CLNUPVER This is a cleanup routine called by IEAVMFRR if an error occurred during input verification.

> Callers: IEAVMFRR
>
> Exit-Normal: Return to IEAVMFRR to continue termination.

Entry Point CLNUPOBT: This is a cleanup routine called by IEAVMFRR if an error occurred while obtaining a console.

Callers: IEAVMFRR

Exit-Normal: Return to IEAVMFRR to continue termination.

IEAVH600 – Communication Task Cross Memory Service Routine

Function: This module executes in the communication task address space and provides the following cross memory services to manipulate write-to-operator queue elements (WQEs) for users of the WTO/WTOR and DOM macros, a console's PFK table, and a console's Screen Area Control Blocks (SACBs).

- Locates a specific WQE in the communication task address space and copies it to the caller's WQE.
- Copies the caller's WQE(s) to the communication task address space.
- Sets the suspend indicator in the WQE for multi-line WTO processing.
- Chains a WQE to an operator request element (ORE) for WTOR processing.
- Copies a PFK table in the communication task address space and frees the old table.
- Selects/verifies an out-of-line area for a console.
- Posts any TRACK ECBs, for a specified console, that have not yet been posted.

Entry from: IEAVMWTO, IEAVVWTO, IEAVXDOM via a PC.

Exit to: Caller via PT.

Called routines: IEAVQ700.

Entry-point FRRCLNUP. At this entry-point IEAVH600 performs cleanup for the communication task functional recovery routine (IEAVMFRR). If IEAVH600 had been processing a single-line WTO or WTOR when the error occurred, FRRCLNUP frees the WQE and adjusts the WQE counts in the UCM base. If EAAVH600 had been processing a multi-line WTO, FRRCLNUP turns off the suspend bit in the WQE and returns to IEAVMFRR with the number of WQEs processed.

Entry from: RTM.

Exit to: RTM.

Called routines: None.

IEAVID00 – Contents Supervision Identify Routine

Function: This module performs two functions:

- Identifies to the system an embedded entry point.
- Identifies to the system a single copy module loaded in subpool 0.

Entry Point IGC041:

Callers: Users of the IDENTIFY SVC service.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point FRRSVC41 insures that the CDE queues are not damaged by a system error and causes a retry in IEAVID00 which allows the issuer of the IDENTIFY SVC to receive control again.

Callers: IEAVTRTS (RTM).

Exit-Normal: Returns to entry-point IGC041 to retry.

Exit-Normal: Returns via RTM to entry-point FRRSVC41 in IEAVID00 to continue with termination.

Exit-Error: No exit error conditions.

Entry-Point CDEVER Ensures that the requested element is a valid CDE.

Callers: IEAVEQV0.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External References: IEAQCDSR, IEAVEQV1, IEAVEADY, and IEAVTPVL

IEAVINIT

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVLKR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVLKRM

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVLK00 – BLDL/FETCH Interfaces

Function: This module contains the description and code for several service routines:

LINK SVC 6 at entry-point IGC006.
XCTL SVC 7 at entry-point IGC007.
LOAD SVC 8 at entry-point IGC008.
DELETE SVC 9 at entry-point IGC009.
SYNCH SVC 12 at entry-point IGC012.
LINK SVC 122 at entry-point IGC0X6.
XCTL SVC 122 at entry-point IGC0X7.
LOAD SVC 122 at entry-point IGC0X8.

In addition to the above service routines, this module includes other code whose functions are described below.

Entry-point IGC006 and IGC0X6:

- Performs a validity check of resources.
- Establishes linkage to the module.
- Searches for and determines the load module to which linkage is desired.
- Creates and updates a program request block, an extended control block (XSB), and a contents directory entry under certain conditions.
- Performs a sharing of load modules in accordance with reusability attributes.
- Invokes the test module via an SVC call in IEAVLK01 for TSO modules in test.

Entry-point IGC007 and IGC0X7:

- Performs linkage to the load module specified.
- Searches lists and libraries for the specified load module.
- Creates and updates a PRB, an XSB, and a CDE as for IGC006.
- Fetches the module.
- Reclaims the current SVRB for type 4 SVC's during the processing.

Entry-point IGC008 and IGC0X8:

- Acquires a specified load module and retains the module for use by the task issuing the load.
- Searches as in IGC006.
- Creates a twelve-byte load list element (LLE) that points to the CDE for the requested module.
- Increases the use count in the CDE and LLE when a module is loaded.
- Creates a 32-byte CDE when none exists for the requested module.
- Keeps a responsibility count in the LLE.
- Returns entry-point even if module is in use; will not queue load requests for a serially reusable module.
- Causes modules to be fetched as for IGC006.

Entry-point IGC009:

- Indicates that the usage of a module is complete.
- Locates the named module.
- Decreases the use count in the CDE and LLE and if the count in the CDE goes to zero, it forces the purging of the module.

Entry-point IGC012:

- Allows the supervisor routine to give control to a problem program routine and to regain control in supervisor mode.
- Gives task recovery the ability to schedule a synchronous exit routine.
- Allows a routine which has received control via a SYNCH request to get control in a specified key.
- Restores the caller's registers on return from a SYNCH request.
- Provides options which allow the caller to set the storage key and the program key mask for the exit that is to receive control.

Entry-point IEAQCS01:

This entry provides supervisor-assisted linkage for ATTACH requests.

Entry-point IEAQCS02: This entry starts a queued request through mainline LINK.

Entry-point IEAQCS03: This entry restarts a queued serially reusable request.

Entry-point IEAVVMSR: This entry searches the pageable LPA directory for an LPDE representing the requested module.

Entry-point IEAQCDSR: This entry searches a CDE queue for a requested name.

Entry-point IEAQCS04: This entry address is the start of a list of addresses which IEAVLK01 uses to return to IEAVLK00. There is no linkage because this is an external reference and not executable.

Entry-point CDLKBASE: This entry address resets the base address for IEAVLK00 on return from IEAVLK01. There is no linkage as this is an external reference and not executable.

Entry from:

For entry-points IGC006 and IGC0X6: SVC IH.

For entry-points IGC007 and IGC0X7: SVC IH.

For entry-points IGC008 and IGC0X8: SVC IH.

For entry-point IGC009: SVC IH.

For entry-point IGC012: SVC IH.

For entry-point IEAQCS01: ATTACH (IEAVEAT0) or memory create (IEAVEMCR).

For entry-point IEAQCS02: POST (IEAVSY50).

For entry-point IEAQCS03: POST (IEAVSY50).

For entry-point IEAVVMSR: Control program and user routines.

For entry-point IEAQCDSR: Control program and user routines.

Exit to:

For entry-points IGC006 and IGC0X6: Routine to be linked via a branch entry to EXIT prolog; wait if request was queued.

For entry-points IGC007 and IGC0X7: Routine to be linked via a branch entry to EXIT prolog; wait if the requested module was queued.

For entry-point IGC008 and IGC0X8: Caller via EXIT prolog (IEAVEEXP).

For entry-point IGC009: Caller.

For entry-point IGC012: Caller using EXIT prolog.

For entry-point IEAQCS02: Same as IGC006.

For entry-point IEAQSC03: Same as IGC006.

For entry-point IEAVVMSR: Return on register 14+0, if an LPDE is found; return on register 14+4 if no LPDE is found.

For entry-point IEAQCDSR: Return on register 14+0, if a CDE is found; return on register 14+4 if no CDE is found.

Error exit RTM via an ABEND macro.

IEAVLK01 – Program Management Search – Fetch Interface Module

Function: This module performs the following services for IEAVLK00:

- Controls the order of queue and library searches.
- Interfaces with the BLDL service routine using SVC 18.
- Interfaces with the program fetch service routine (IEEWMSEPT) using a branch.
- Provides an interface to test using SVC 61.
- Processes alias PDS entries so that the major name module can be reused, if possible, for an alias request.

Entry from: IEAVLK00 via branch.

Exit to: IEAVLK00.

IEAVLK02 – Program Management Resource Management Routines

Function: This module cleans up CDEs and any associated storage at EXIT, DELETE, ABEND and EOT.

Entry from: ABEND.

Exit to: Program management.

Entry-point CDHKEEP This entry releases the module, extent list, major CDE, and associated minors as required.

Entry from: DELETE service routine (IGC009).

Exit to: Caller via BR 14.

Entry-point GXLHKEEP This routine frees CSA storage associated with modules that have been loaded into the CSA.

Entry from: RTM2 via a branch.

Exit to: Caller via a branch.

Entry-point IEAPPGMX This routine frees the program management resources when an RB exits and determines if a task termination is in progress, and if so, frees any LLEs left on the TCBLLE queue, their associated CDEs and its associated storage.

Entry from: EXIT (IEAVEOR) via branch with LOCAL lock held.

Exit to: Caller (SVC 3) via branch.

Entry-point IEAPPGMA This routine removes partially loaded programs during ABEND processing.

Entry from: IEAVTRT1 via branch.

Exit to: Caller via BR 14.

IEAVLK03 – Program Management Recovery Routine

Function: This module attempts to prevent unwanted abends in the program manager by intercepting them and correcting errors if possible. The routine verifies CDE and LLE queues for abend situations that are not due to user error.

Entry-point CDEVER This entry-point receives as input an address and verifies that the address points to a CDE.

Entry from: IEAVEQV0.

Exit to: Caller.

Entry-point FRRPGMMG This entry-point provides program manager recovery for the basic program management functions.

Entry from: RTM.

Exit to: RTM.

Entry-point FRRPGMX This entry-point is the recovery routine for the exit resource manager (IEAPPGMX). It verifies the CDE and LLE queues and attempts a retry.

Entry from: RTM.

Exit to: RTM.

IEAVLSCO – Linkage Stack Constants Table

Function: A non-executable module that contains constants for the dimensions of the different types of linkage stacks.

IEAVLSCR

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVLSEX

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVLSGO

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVLSGR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVLSIH

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVLSIO

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVLSIR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVLSLC

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher.*

IEAVLSLO

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVLSLR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVLSPR – Processor Linkage Stack Reinitialize Routine

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVLSTA

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVMASV – Communication Task Asynchronous Service Routine

Function: The asynchronous service routine performs TPUT services for the communication task. The TPUT services issue the WAIT macro instruction.

Entry from: IEAVTPUT and IEAVMASV.

Exit to: The dispatcher via a BR 14.

IEAVMDOM – Communication Task DOM Processor

Function: The communication task DOM processing routine processes the DOM control blocks (DOMCs) created by the DOM macro instruction processing routine (IEAVXDOM). For each DOMC message ID, this routine attempts to eliminate the WQE that has the same message ID and the ORE associated with that WQE, if an ORE exists. If the WQE was retained in the action message retention buffer, it is removed. The device support processor is then called to eliminate graphic messages with the same message ID.

Entry from: IEAVMQWR.

Exit to: The communication task's wait service routine via a BR 14 instruction.

IEAVMDSV – Communication Task Device Service Routine

Function: The device service routine prepares the communication task for and issues SVC 72 (the device support processor (DSP)). The preparation includes handling pending open, close, I/O completion interrupt, and attention interrupt processing. This routine also cleans up the write queue elements (WQEs), operator reply elements (OREs), and console queue elements (CQEs) when these control blocks are no longer needed, and assures that all messages have been processed for master tracing. Additionally, it retains messages in the action message retention buffer.

Entry from: IEAVMQWR and IEAVMDSV.

Exit to: Caller via a BR 14 instruction.

IEAVMED2 – WQE/ORE Purge Routine

Function: This module purges WTO elements and ORE (operator reply elements) at task or address space termination. In particular, it:

- Frees any write wait blocks (WWBs) from the ORE and WQE chain.
- Marks any OREs to indicate that deletion is in progress.
- If this is an address space deletion, creates a DOMCB for a DOM by address space identifier (ASID).
- If deleting a job step, it creates a DOMCB for a DOM by ASID and job step TCB.
- Marks any active graphic device UCME to indicate that there is DOM work to be done.
- Stops monitor requests for this task or storage by calling the monitor control routine (IEAVMNTR).
- Posts the UCMDECB.
- Passes a task-related DOMCB to the subsystem interface.

Entry from: Address space or task termination.

Exit to: Caller.

IEAVMFIH – MSSFCALL SLIH Routine (IEAVMFIH)

Function: IEAVMFIH processes class 2401 MSSF external interrupts, which are generated by the MSSF hardware when it finishes processing an MSSFCALL DIAGNOSE instruction.

Entry from: IEAVEEXT.

Exit to: Caller.

IEAVMFRM – MSSFCALL Resource Manager

Function: IEAVMFRM cleans up any MSSFCALL resources held by a terminating task or address space. These MSSFCALL resources include the MSFAB, MSFKB and MSFCB control blocks and the MSSFCALL TP PORT serialization resource.

Entry from: IEAVTMMT during address space termination or IEAVTSKT during task termination.

Exit to: Caller.

IEAVMFRR – Communication Task Functional Recovery Routine (FRR) or ESTAE Controller

Function: The FRR/ESTAE controller receives control from recovery termination management (RTM). The controller processes both FRR and ESTAE attempts to recover from abnormal terminations in the communication task.

Entry from: RTM via FRR stack.

Exit to: RTM via a BR 14 instruction.

IEAVMNTR – Monitor Queue Manager

Function: This module maintains the monitoring status of operators' consoles and TSO terminals. This is done primarily for the MONITOR and STOPMN command processors.

Entry from: IEAVMED2, IEAVSWCH, IEE5503D, IEE7103D, IKJEFF00, or IEE4603D.

Exit to: Caller.

IEAVMODE

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Task Management.*

IEAVMQR0 – Communication Task Process Message for Inactive Terminal (Console)

Function: This routine gives the master console operator an opportunity to delete, queue to the master console, or continue queueing messages that are sent to a specific console by identification.

Entry from: IEAVMWSV.

Exit to: The dispatcher via a BR 14 instruction.

IEAVMQWR – Communication Task Wait Service Routine

Function: The wait service routine waits on all of the event control blocks (ECBs) that are posted for the communication task.

Entry from: IEAVN701.

Exit to: No exit; this is a never-ending routine.

IEAVMSF – Monitoring and System Support Facility (MSSFCALL) SVC Routine

Function: IEAVMSF provides the interface between MVS software and the hardware's monitoring and system support facility (MSSF). It determines whether the requested MSSFCALL command can be processed, and, if so, issues an MSSFCALL DIAGNOSE instruction on behalf of the requestor.

Entry from: SVC FLIH (IEAVESVC) or disabled branch entry callers at entry-points IEAVMSF1 or IEAVMSF2.

Exit to: Address in register 14.

Called routines: Validity check routine (IEAVEVAL).

IEAVMSFS – MSSFCALL SRB Routine

Function: After the MSSF hardware finishes executing an MSSFCALL DIAGNOSE instruction that IEAVMSF issued while processing an MSSFCALL SVC, it generates a class 2401 MSSF external interrupt. IEAVMFIH receives control to schedule IEAVMSFS, which posts the ECB that IEAVMSF is waiting on.

Scheduled by IEAVMFIH.

IEAVMP08 – VSM Initialization Routine

Function:

Listed by entry point.

Entry-Point IEAVMPA8:

Processes the SQA parameter and builds the VSM control blocks and page tables necessary to describe the new SQA and ESQA space.

Callers: IEAVNIPM

Exit-Normal: N/A

Exit-Error: The following wait state codes.

X'03D' - Unable to initialize page and/or segment tables.

Entry-Point IEAVNPB8:

Processes the CSA parameter and builds the VSM control blocks and page tables necessary to describe the new CSA and ECSA space.

Callers: IEAVNIPM

Exit-Normal: N/A

Exit-Error: The following wait state codes.

X'03D' - Unable to initialize page and/or segment tables.
X'0E3' - LPA or ELPA overlaps the private area.

Entry-Point IEAVNPC8:

Calls RSM to build the page tables for LPA and ELPA and marks the space GETMAIN assigned.

Callers: Contents supervision and ASM RIMs.

Exit-Normal: N/A

Exit-Error: The following wait state codes.

X'03D' - Unable to initialize page and/or segment tables.

External-References:

Routines:

- IARVB
- IARVP
- IGVAAQAT

Tables:

- IGVSPTT
- IHAAQAT
- IHASPT

IEAVMVC0 – DATOFF Macro Service Routine

Function: This module contains several generalized DATOFF routines. Access to the routines is through the DATOFF macro.

Entry from:

For entry-point IEAVMVC0: Issuer of the DATOFF macro with INDMVCL0.

For entry-point IEAVXC0: Issuer of the DATOFF macro with INDXC0 index.

For entry-point IEAVMVKY: Issuer of the DATOFF macro with INDMVCLK index.

For entry-point IEAVDFTA: Issuer of the DATOFF macro when an invalid index is detected.

Exit to: Caller.

Error exit ABEND.

IEAVMWSV – Communication Task Console Queueing Routine

Function: The console queueing routine selectively queues messages to console output queues and passes eligible messages for master tracing.

Entry from: IEAVMDSV, IEAVMQWR.

Exit to: The device service routine (IEAVMDSV), which returns control to the communication task's wait service routine (IEAVMQWR).

IEAVMWTO – Communication Task Multiple-Line WTO Service Routine

Function: The communication task multiple-line WTO (MLWTO) service routine prepares major and minor write queue elements (WQEs) associated with multiple line messages.

Entry from: IEAVM600.

Exit to: IEAVM600.

Called routines: IEAVH600, IEAVM700, IEFJSREQ, IEAVX600.

IEAVM600 – WTO/WTOR Initialization/Verification Routine

Function: Begins processing for a WTO/WTOR request. A user issues a WTO or WTOR which results in an SVC 35. A service request for a single line WTO or a WTOR will be handled by IEAVVWTO. A multiline WTO is handled by IEAVMWTO.

Entry Point: IEAVM600

Callers: SVC interrupt handler

Exit-Normal: Return to caller with the DOM id in register 1.

Exit-Error: To a FESTAE routine or an FRR (the FRR while the local lock is held). All error conditions result in a D23 abend.

External References:

- IEAVVWTO - to handle single line WTO/WTOR

- IEAVMWTO - to handle multi-line WTO
- IEECVXIT - user exit

IEAVM700 – MPF Scan Routine

Function: This module scans the MPF table built by IEECB805. The table contains message IDs and prefixes of messages to be suppressed. IEAVM700 determines whether the message ID or prefix passed as input matches any of the entries in the MPF table.

Entry from: IATCNMR, IEAVVWTO, IEAVMWTO.

Exit to: IATCNMR, IEAVVWTO, IEAVMWTO.

IEAVNPA6

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVNPCD – Reconfiguration Statis/Dynamic Partition RIM

Function:

Initialize the CPU side id array as follows:

- If the system is dynamically partitioned, place the lowest CPUid on a side in the array entries for all CPUs defned to the side.
- If the system is not dynamically partitioned, place the lowest CPUid defined to the system in the array entries for all CPUs defined to the system.

Entry-Point IEAVNPCD: See function.

Callers: IEAVNIPM - NIP Control/Service Routines.

Exit-Normal: Return to caller.

Exit-Error: Return to caller.

IEAVNPD6

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVNP00 – Reconfiguration Processor Initialization

Function: This routine initializes the IPL processor's clock and console restart text. It brings online and initializes the non-IPL processors in the hardware configuration.

Entry-Point IEAVNP00: See function.

Callers: IEAVNIPM.

Exit-Normal: Return to caller via BR 14.

External References: IEAVRTOD, IEEVCPR, and IEEVSAI.

IEAVNP06

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVNP09 – ASVT, AFT, AST Creation

Function: This module builds the ASVT and assigns the permanent address spaces. It also builds the address first tables (AFT) and address second tables (AST).

Entry from: IEAVNIPM.

Exit to: Return to caller.

IEAVNP27 – Reconfiguration Installed Resources RIM

Function: This routine obtains and initializes the installation channel path table (ICHPT) and the configuration management table (CMT).

Entry-Point IEAVNP27: See function.

Callers: IEAVNIPM.

Exit-Normal: Return to caller.

Exit-Error: Disabled wait state.

External References: IEEVORGI.

IEAVNP47 – Event Notification Facility Initialization

Function:

Initialize the Event Notification Facility (ENF) control structure.

Entry-Point IEAVNP47: Create and initialize ENF control structure.

Callers: None.

Exit-Normal: Return to IEAVNIPM.

Exit-Error: OUTPUT - Wait state 064.

External-References: IEFENFNM - Event Notification processing routine.

IEAVNP57 – Dumping Services Address Space RIM

Function: This module calls module IEAVTSAS to create the Dumping Services address space. If the address space cannot be created, message IEA768I is issued.

Entry from: IEAVNIPM.

Exit to: Caller.

Called routines: IEAVTSAS, NIPWTO.

IEAVNUCM – Nucleus Map Formatter

Function: This module formats the nucleus map that is contained in the dump data set.

Entry from: IEAVNUCM.

Exit to: Caller.

IEAVQ700 – Communications Task Queue Scanner.

Function: Scans a queue searching for a particular entry.

Entry Point IEAVC701

Callers: No specific caller.

Exit-Normal: Return to caller via BR 14

Exit-Error: IEAVMFRR via RTM

External References: IEAVMFRR - if an error occurs while scanning the queues

IEAVRTI0 – Timer Second Level Interrupt Handler

Function: This module processes interrupts caused by a synchronization check, by the CPU timer, or by the clock comparator.

Entry-point IEA0TI00 from IEAVEEXT.

Entry from: IEAVEEXT.

Exit to: IEAVEEXT.

Entry-point IEAQTE00 (TQE enqueue routine).

Pointed to by CVTQTE00.

Entry from: IEAVRT00, IEAVRT02, IEAVRTOD, IEAVRTI1, IEE6503D, IGX00013, IRARMSRV.

Exit to: Caller.

Entry-point IEAQTD00 (TQE dequeue routine).

Pointed to by CVTQTD00.

Entry from: IEAVRT00, IEAVRTOD, IEAVRTI1, IEE6503D, IRARMSRV.

Exit to: Caller.

Entry-point IEAVRCKQ (to ensure that the top TQE in the system's real time queue is being timed by at least one of the clock comparators in the system).

Pointed to by TPCCKQ.

Entry from: IEAVRTI0, IEAVRTOD, IEAVRTI1.

Exit to: Caller.

Entry-point IEAVRQCK (routine that gets control via a RPSGNL macro with the RQCHECK option).

Entry from: IEAVEXS.

Exit to: IEAVEXS.

Entry-point IEAVRJLM (routine to interface with SMF whenever an address space has exceeded its time limit).

Entry from: IEAVRTI0, IEAVEDS0.

Exit to: Caller.

Entry-point IEAVRSAE (SRB routine that gains control whenever the timer SLIH, IEAVRTI0, schedules an SRB).

Pointed to by SRBEP (initialized when the SRB is built).

Entry from: IEAVEDS0 (dispatched).

Exit to: IEAVEDS0.

IEAVRTI1 – Timer Purge and Recovery

Function: This module performs four functions: the TQE purge function called by a terminating task or memory; the hardware recovery routine called by RTM; initialization of timer control block fields and timer hardware when a processor is varied on-line; and the FRR for timer supervision which includes a TQE validation routine.

Entry-point IEAQPGTM from IEAVTSKT or IEAVTMMT.

Entry-point IEAVRSPG from IEAVEPD0.

Entry-point IEAVRCLS from IEAVTRTH.

Entry-point IEAVRCLX from IEAVEDS0.

Entry-point IEAVRNEW from IEEVCPR.

Entry-point IEAVRSPN from IEAVEDS0.

Entry-point IEAVRFRR from IEAVTRTS.

Entry-point IEAVRTVR from IEAVRTI1 (routine IEAVRFRR). or IEAVEQV0.

Exit to: Caller.

IEAVRTOD – TOD Clock Manager

Function: This module initializes TOD clocks at IPL and sets or synchronizes the TOD clock for a processor being varied on-line or for a processor whose clock has suffered a machine check.

Entry-point IEAVRINT from IEAVNP00.

Entry-point IEAVRSSC from IEAVRTOD and IEEVCPR.

Entry-point IEAVRCLA from IEAVEDS0.

Entry-point IEAVRNOT from IEEVCPR.

Entry-point IEAVRCAN from IEEVCPR.

Exit to: Caller.

IEAVRT00 – Timing Service Routines: TTIMER/STIMERM TEST/CANCEL (SVC 46) and STIMER/STIMERM SET (SVC 47)

Function: The STIMER/STIMERM SET routines establish time intervals for the requesting task. TTIMER/STIMERM TEST/CANCEL routines test or cancel previously established time intervals.

Entry-point IGC046 from IEAVESVC.

Entry-point IGC047 from IEAVESVC.

Exit to: Both of these entry-points exit to IEAVEOR via Exit Prolog (IEAVEEXP).

IEAVRT01 – TIME Service Routine (SVC 11)

Function: This module supplies the user with local time and date, or optionally, with Greenwich Mean Time and date. The current value of the Time-of-Day (TOD) clock can also be requested.

Entry-point IGC0001A from IEAVESVC.

Exit to: IEAVEOR via exit prolog (IEAVEEXP).

Entry-point IEAVRTME from IEAVEES.

Exit to: IEAVEES.

IEAVRT02 – Set DIE Routine

Function: This module allows a system program to establish a real-time interval. After the interval has passed, the program's disabled interrupt exit (DIE) routine gets control.

Entry-point IEAVRDIE from system caller.

Pointed to by TPCSDIE.

Exit to: System caller.

IEAVRT03 – SRB Time Limit Initialization

Function: This module establishes a specified time limit for a program executing in SRB mode.

Entry from: Issuer of a SRBTIMER macro.

Exit to: Caller.

Error exit Return to caller that is not in SRB mode.

IEAVRT04 – CPU Timer Service Routine

Function: This module supplies the invoker with the current value of the CPU timer. The value is returned in either timer units (TU) or in microseconds (MIC), depending on an input parameter.

Entry from: Issuer of a CPUTIMER macro.

Exit to: Caller.

IEAVSETS

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Task Management.*

IEAVSPDM – Processor Controller Damage Monitor Routine

Function: IEAVSPDM is attached by the master scheduler (IEEMB860) during initialization and waits on an ECB that will be posted when a processor machine check occurs. If the ECB is posted, IEAVSPDM notifies all listing subsystems that the processor controller is damaged via the subsystem interface. IEAVSPDM also issues a message to notify the operator that the processor controller is damaged.

IEAVSSAF – Vector Status Save Area Dump Formatter

Function: This module formats vector status save area data.

Entry from:

For entry point IEAVSSA1 - Print dump

For entry point IEAVSSA2 - IPCS or stand alone dump

Exit to: Caller

IEAVSTAA – Communications Task ESTAE Routine

Function: This routine performs ESTAE processing and retry processing for all abends that occur in communications task.

Entry point. IEAVSTAA

Purpose Recovers from errors that occur during communications task processing.

Entry from: RTM.

Exit-Normal: Returns to RTM

Entry point STAARTRY.

Purpose Cleans up and performs retry processing.

Entry from: RTM.

Exit-Normal: Exits to IEAVMQWR.

Called routines: IEA0P01, IEEMB885

IEAVSTA0

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVSTA1

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVSTA3

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVST0F

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVSVCT

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Supervisor Control.*

IEAVSWCH – Communication Task Console Switch Routine

Function: The console switch routine analyzes and switches console functions.

Entry from: IEAVMWSV, IEECVETW, IEAVVCTR, IEECVETC, IEAV1443, IEAV2540, IEE8B03D.

Exit to: A branch return to the address in register 14.

IEAVTABD – ABDUMP Processing Routine

Function: IEAVTABD processing checks options, prepares the dump data set, constructs a SNAP parameter list for the actual dump, and gives control to SNAP dump processing.

Callers: IEAVAD01.

Entry from: IEAVTRTC via BALR.

Exit to: Return to caller.

IEAVTABI – ABDUMP Initialization

Function: This module reads PARMLIB IEAABD00, IEADMP00, and IEADMR00 and sets the requested dump options in the recovery termination control table (RTCT) for later use by ABDUMP (IEAVTABD).

Entry from: IEAVNPD1 via BALR.

Exit to: IEAVNPD6.

IEAVTABX – ABDUMP Pre-dump EXIT-CSECT and Load Module

Function: No executable code. Contains load module names of installation pre-dump exit load modules.

Entry from: Loaded by IEAVTABD.

Exit to: IEAVTABD.

IEAVTACR – Alternate CPU Recovery

Function: This module uses a recovery processor to execute the recovery routines (FRRs) of the failing processor. This should free any system resources held by the failing processor and allow normal system processing to continue on the recovery processor. This is accomplished in three phases:

- Pre-processor phase status saving
- Intermediate phase resolve resource contention
- Post processing phase system cleanup, to account for the loss of a processor

Entry from:

- EMS/MFA MCH SLIH on the recovery processor.
- Lock manager and certain global spin routines when conflict is detected between recovery and failing processor work.
- Dispatcher when recovery or failing processor work is interrupted.

Exit to:

- Caller via register 2.
- Interrupted work of other processor, if this is the first failing-processor-work-exit to RTM for FRR recovery.
- If work of each processor is in the enabled condition, exit to dispatcher.

IEAVTADR – Address Conversion Routine

Function: This module converts a coded string representing a direct or indirect address into a 4-byte virtual address.

Entry from: IEAVTSLP via BALR, AHLTSLIP via BALR, IEECB905 via BALR.

Exit to: Caller.

IEAVTAS1

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTAS2 – Task Recovery Post Exit Processing IEAVTAS3 – Setup for User Retry or Continue with Termination

Function: This module performs processing for user retry or percolation. Processing consists of the following:

Retry:

- Identify the RB for retry.
- Initialize or freemain user's SDWA for retry.
- Close embedded DCBs.
- Purge paging I/O.
- Purge outstanding WTOR requests for STAE retry only.
- Purge entries in type 1 message table.
- Set pointer to EXIT (RB) or exit prolog (SVRB) in retry PSW.
- Perform SCB clean-up.
- Handle pseudo retry into IEAVTAS1 if there is recursion.

Percolation (continue with termination)

- Free user's SDWA.

Entry from: IEAVTAS1 via BALR.

Exit to: IEAVTAS1.

IEAVTB00 – SPIE/EXTRACT Service Routine

Function: The module provides two services:

- SPIE, which provides the problem program with a means of specifying an error exit routine in response to one or more program error interruptions.
- EXTRACT, which provides a means of obtaining the address of the values of a set of fields without going through the system.

  For entry IGC0001D: Performs SPIE mainline processing.

  For entry IEAVSPIE: SPIE resource manager.

  For entry IEAVSPI: Checkpoint/restart entry.

  For entry IGC00040: Performs EXTRACT mainline processing.

  For entry IGC00040+8: Performs EXTRACT mainline processing.

Entry from:

For entry IGC0001D: SVC IH.

For entry IEAVSPIE: RTM.

For entry IEAVSPI: Checkpoint/restart.

For entry IGC00040+8: Supervisor routines via branch entry.

Exit to:

For entry IGC0001D: Exit prolog.

For entry IEAVSPIE: Caller.

For entry IEAVSPI: Caller.

For entry IGC00040: Exit prolog.

For entry IGC00040+8: Caller.

IEAVTCR1 – Alternate CPU Recovery Module

Function: This module recovers failures in mainline ACR (IEAVTACR). Each mainline failure is retried once (with the exception of the IOS Machine check exit, IOSRMCH, which is given control during the post processing phase of IEAVTACR). In the event of a second mainline failure on retry, the system is terminated.

Entry from: A failure in module IEAVTACR

- Direct from branch from RTM-ACR superbit (PSAACR) pre-processor/intermediate phase.
- FRR routing post processing phase.

Exit to:

- IEAVTACR to retry mainline processing.
- System termination IEFPTERM on recursion during retry.

IEAVTDSE – DUMPSRV Address Space End of Task Exit Routine (ETXR)

Function: This module provides recovery for tasks in the DUMPSRV address space. If the IEECB926 task (dump data set processor) fails, the old task is detached and a new copy of IEECB926 is attached. If the IEAVTSEP task (SDUMP post exit processor) fails, the task is detached. If the ADYTRNS task (DUMP Analysis and Elimination transaction processor) fails, the task is detached and a new copy attached.

Entry from: RTM.

Exit to: RTM.

Called routines: IEECB926 and ADYTRNS are attached.

IEAVTDSV – DUMPSRV Address Space Job Step Task

Function: This module controls the functions in the DUMPSRV address space after the address space has completed its initialization. The IEECB926 task is attached to process SYS1.DUMP data sets and the DUMPDS operator command. The ADYTRNS task is attached to process DAE (DUMP Analysis and Elimination) transactions. The post dump exit routines (specified in IEAVTSEL) are loaded into DUMPSRV private storage. IEAVTDSV then waits for requests to process post dump exits.

Entry from: START processor when address space is created, cross memory Post from IEAVTSD1 when a dump completes.

Exit to: None. This is a never ending task which waits for post dump exit processing requests.

Exit error If this task fails, the DUMPSRV address space terminates. Cleanup for a DUMPSRV termination is handled by the SDUMP memory termination resource manager (IEAVTSDR).

Called routines: IEECB926, IEAVTSEP, and ADYTRNS are attached. IEAVTSEL is loaded (data csect).

IEAVTESP – SPIE/ESPIE Service Routine

Function: This module processes the SPIE and ESPIE SVCs, performs checkpoint/restart processing, RB, task and address space termination resource manager functions, and determines the correct PSW program mask for programs that get control via the SYNCH macro.

Entry from:

For entry-point IEAVSPIE: IEAVEOR or RTM.

For entry-point IEAVSPI: Checkpoint/Restart.

For entry-point IGC0001D: FLIH for SPIE.

For entry-point IGX00028: SVC FLIH for ESPIE. his module provides recovery for tasks in the DUMPSRV address space. If the IEECB926 task (dump data set processor) goes down, the old task is detached and a new copy of IEECB926 is attached. If the IEAVTSEP task (SDUMP post exit processor) fails, the task is detached. If the ADYTRNS task (DUMP Analysis and Elimination transaction processor) fails, the task is detached and a new copy attached.

For entry-point IEAVSPEX: IEAVEDR.

Exit to: Caller in 31-bit addressing mode.

IEAVTEST – TESTAUTH Service Routine

Function: This routine tests the authorization of any program to perform a specified function.

Entry from:

For entry IGC119: SVC IH.

For entry IEAVTEST: Supervisor routine via branch entry.

Exit to:

For entry IGC119: EXIT prolog.
For entry IEAVTEST: Caller.

IEAVTEXS

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTFDE

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTFDH

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTFDR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTFDS

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTFEH

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTFHX – HEX Format Subroutine

Function: This subroutine formats a single summary dump record in standard hexadecimal dump format with addresses on the left-hand side.

Entry from: IEAVTFSD.

Exit to: Caller.

IEAVTFHZ – Hex Dump Format Subroutine

Function: This subroutine formats a single summary dump record in standard hexadecimal format with offsets on the left-hand side and with characters on the right-hand side.

Entry from: IEAVTFSD.

Exit to: Caller.

IEAVTFMT

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTFRD – Summary Dump Read Routine

Function: This module reconstructs the records of a summary dump from a dump data set being processed by AMDPRDMP.

Entry from: IEAVTFSD.

Exit to: Caller.

IEAVTFRO

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTFSD – AMDPRDMP Service Aid Exit for SUMDUMP Control Statement

Function: This module controls the formatting of an entire summary dump by calling format subroutines to process each summary dump record.

Entry from: AMDPRDMP via the user exit interface (entry-point AMDUSRXT in AMDPRUIM).

Exit to: Caller.

IEAVTFTM – Message and Subroutine Name CSECT for IEAVTFSD

Function: This message CSECT contains all the text for the dump output headers and comments generated by IEAVTFSD, IEAVTFRD, and IEAVTFHX. It also contains a branch table of subroutines to load and call for each type of summary dump record.

Entry from: IEAVTFSD.

Exit to: Caller.

IEAVTFXM – Interface to Cross Memory Control Block and RTM Control Block Formatters

Function: This module simulates a total dump environment by using summary dump data. It invokes the Exit Service Router to request control block formatting of each IHSA, RTM2WA, XSB, and PC Link Stack.

Entry from: IEAVTFSD.

Exit to: Caller.

Called routines: The Exit Service Router, BLSQEXTI

IEAVTFXR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTGLB – SLIP Global PER Activation/Deactivation Routine

Function: This module manages the global resources on all processors necessary to activate, deactivate, or adjust PER monitoring. This module is scheduled by IEECB905, IEAVTJBN, IEAVTLCL, and IEAVTSLP.

Entry from: SRB dispatcher.

Exit to: SRB dispatcher.

IEAVTJBN – SLIP PER Selection Interface Routine

Function: This module serves as an interface between address space create and IEAVTLCL to determine if a new address space should have PER monitoring active. It also serves as the interface between scheduler routines (started task control and SWA create control) and SLIP PER routines to determine if a LOGON, START, or job select requires a change to PER monitoring in an address space.

Entry from: IEAVEMRQ via BASSM, IEESB605 via BASSM, IEFIB600 via BASSM.

Exit to: Caller.

IEAVTLCL – SLIP Local PER Activation/Deactivation Routine

Function: This module determines if PER monitoring should be on or off for the address space in which it is running and adjusts the RB and ASCB indicators accordingly. IEAVTGLB also schedules this routine to adjust the RB and ASCB indicators as required. This routine is scheduled by IEAVTGLB and IEAVTJBN.

Entry from: SRB dispatcher.

Exit to: SRB dispatcher.

Called routines: IEATRSCN, IEAVESSE, IEAVSETS.

IEAVTMMT

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTMRM – RTM Address Space Termination Resource Manager

Function: This module frees all the SDWAs obtained from SQA by RTM1.

Entry from: IEAVTMMT via BALR.

Exit to: Caller.

IEAVTMSI

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTMTC

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTMTR – Memory Termination Requestor

Function: This module is attached by the memory termination controller to invoke memory termination purges via SVC 13.

Entry from: IEAVTMTC via ATTACH.

Exit to: Caller.

IEAVTPER – Program FLIH/SLIP PER/Space Switch Interface Module

Function: This module provides the interface used to call the SLIP processor (IEAVTSLP) to handle a PER interrupt or the space switch handler (IEAVTSSH) to handle a space switch interrupt detected by the program check FLIH (IEAVEPC).

Entry from: IEAVEPC via BALR.

Exit to: Caller.

IEAVTPMT – ABEND Message Table Processor

Function: This module deletes entries from the GETMAIN/FREEMAIN ABEND information message table and, if requested, formats the message table entry and issues a WTO to print the message.

Entry-Point IEAVTPMT:

Callers: IEAVTAS3 (RTM), IEAVTMMT (RTM), IEAVTRTC (RTM), IEAVTSKT (RTM).

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External References: None.

IEAVTPVT – SLIP PVTMOD LOAD/DELETE Exit Routine

Function: This module processes a contents directory entry (CDE) whenever CDEs enter or leave the job pack area queue if the CDE matches a PVTMOD PER trap.

Entry-Point IEAVTPVL:

Callers: Contents Supervisor when a CDE is placed on the job pack queue and the extent list is valid (LINK, LOAD, IDENTIFY, XTCL, ATTACH) or when the module is brought into memory (virtual fetch).

Exit-Normal: Returns to caller.

Exit-Error: Percolates from entry point, PVTFRR.

Entry-Point IEAVTPVD:

Callers: Contents Supervisor when a CDE is removed from the job pack queue.

Exit-Normal: Returns to caller.

Exit-Error: Percolates from entry point, PVTFRR.

Entry-Point IEAVTPVR:

Callers: Virtual Fetch's resource manager at end of job step task and end of memory.

Exit-Normal: Returns to the caller.

Exit-Error: Percolates from entry point, PVTFRR.

Entry-Point PVTFRRK:

Callers: RTM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External References:

Branch enter lock manager routines
Branch enter SDUMP
Branch enter GETMAIN
Branch enter FREEMAIN
Branch enter recording facility
Branch enter RISGNL routine
Branch enter SLIP PER RISGNL routine (IEAVTSIG)
Branch enter Cross Memory POST

IEAVTRCA

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTREF – LOGREC Recording Buffer Formatter

Function: This module is a dump formatting exit that can be called from Print Dump (PRDMP) or the Interactive Problem Control Program (IPCS). IEAVTREF is invoked when the IPCS/PRDMP LOGDATA verb is executed. IEAVTREF locates the LOGREC entries that are contained in the LOGREC recording buffer and invokes the EREP program to format and print the LOGREC entries.

Entry-Point IEAVTREF: Formats and prints the LOGREC recording buffer

Callers: PRDMP or IPCS

Exit-Normal: Returns to caller.

Entry-Point REFIO: Prints the formatted output from IFCRCGIF

Callers: IFCRCGIF

Exit-Normal: Returns to caller.

External References:

- Routines: IFCRCGIF - EREP service to format LOGREC entries.
- Control blocks: ABDPL, CVT, HDR, PSA, RBCB, RCB, RCBENTRY, and SRB

IEAVTREM – Record Resource Manager

Function: This module is a resource manager that is given control at memory termination time. IEAVTREM scans the record control block (RCB) buffers for entries belonging to the address space being terminated. If an entry is found that has not been marked ready for termination, IEAVTREM marks the entry as invalid. IEAVTRET, the record task, then knows not to wait for the entry to be completed and just removes the enfry from the buffer.

Entry-Point IEAVTREM:

Callers: IEAVTMTC.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point RCDRMRCV:

Callers: RTM.

Exit-Normal: Terminates.

Exit-Error: Percolates.

External References: ESTAE service routine.

IEAVTRER – Record Request Routine

Function: This module determines whether the caller requested the recording function (via the RECORD macro) for an emergency request (by specifying TYPE = TERM) or for a non emergency request (by specifying TYPE = LOGREC or TYPE = WTO) and prepares the buffer for recording of the error asynchronously under the recording task, IEAVTRET.

Entry-Point IEAVTRER: Determines whether this is an emergency request (TYPE = TERM) or a non emergency request (TYPE = LOGREC) or (TYPE = WTO) and prepares the buffer area for the recording of errors.

Callers: Issuers of the RECORD macro

- that specify TYPE = LOGREC or TYPE = WTO (any key 0, supervisor routine using module IEAVTRGR)
- that specify TYPE = TERM (only module IGFPEMER)

Exit-Normal: Returns to the caller

Exit-Error: Returns to the caller

Entry-Point RCDRCVR: Recovers from errors encountered during IEAVTRER.

Callers: RTM

Exit-Normal: Requests retry.

Exit-Error: Requests percolation.

Entry-Point RERSRBEP: Issues the POST macro instruction for the ECB waited on by the Recording Task, IEAVTRET.

Callers: Dispatcher

Exit-Normal: Returns to the dispatcher.

Exit-Error: There are no exit error conditions.

External References:

- Routines:

  The Post service, via the POST macro
  The Schedule service, via the SCHEDULE macro
  The Setfrr service, via the SETFRR macro
  The Setlock service, via the SETLOCK macro

- Data areas: There are no data areas.
- Control blocks: ASCB, CVT, PSA, RBCB, RCB, RCBENTRY, SDWA, and SRB.

IEAVTRET – Asynchronous Recording Task

Function: This module writes recording requests.

Entry from: IEVTMSI via ATTACH, subsequently IEAVTRER schedules an SRB to POST RTCTRECB.

Exit to: None. This is a never-ending task that waits on an ECB when there is no work to be

IEAVTRET – Recording Task

Function: This module writes records to SYS1.LOGREC or to the operator which have been scheduled via the RECORD macro. If requested, IEAVTRET will free unbuffered storage and/or post the caller after the requested processing is complete. IEAVTRET will write a lost record summary to LOGREC if records were lost, and it passes control to the MCH WTO routine for an MCH operator message (decision on whether or not a message is required is made by the MCH routine.)

Entry-Point IEAVTRET: See function.

Callers: IEAVTMSI

Exit-Normal:

Exit-Error:

Entry-Point RCDTSKR: Recovers from errors encountered during IEAVTRET

Callers: RTM

Exit-Normal: Request retry

Exit-Error: Percolates

External References:

- Routines:

  ESTAE service - establishes the ESTAE environment

  GETMAIN service - obtains storage for the work buffer

  FREEMAIN service - frees the caller's buffer

  MCH WTO service (IGFPWMSG) - writes machine check messages to the operator if required

  PGSER service - releases the work buffer after processing

  WAIT service - waits for work to be done

  WTO service - writes messages to the operator

  POST service - signals the user that the request is complete

  SETLOCK service - obtains the local lock for serialization

- Control blocks: ASCB, ASVT, CVT, LRB, PSA, RBCB, RCB, RCBENTRY, SDUMP, SDWA, and SRB

IEAVTRGR – RECORD Addressing Mode Interface Routine

Function: This routine preserves the addressing mode of the issuer of RECORD and ensures that the RECORD service routine, IEAVTRER, is branch-entered in 31-bit addressing mode.

Entry from: Issuer of the RECORD macro.

Exit to: IEAVTRER.

IEAVTRF2 – RTM2WA Format Module

Function: This module formats the RTM2WA (using IEAVTRF2's formatting model CSECT IEAVTRP2) and summarizes the bit flags. IEAVTRF2 sends a return code to its caller indicating success or failure in formatting.

Entry from: IEAVTFMT via BLSQROUT and BLSQCFMT

Exit to: Caller

IEAVTRF3 – FRR and RT1W Control Block Format Module

Function: This module formats the FRR (using IEAVTRF3's formatting model CSECT IEAVTRP3), formats RT1W (using IEAVTRF3's formatting model CSECT IEAVTRP3), and summarizes RT1W's bit flags. If RT1W is valid and if the EED pointer does not equal zero, the EED is also formatted via IEAVTRF5. IEAVTRF3 sends back a return code to its caller indicating failure or success in formatting.

Entry from: IEAVTFMT via BLSQROUT and BLSQCFMT

Exit to: Caller

IEAVTRF4 – SCB/SCBX Control Block Format Module

Function: This module formats the SCB and the SCBX (using IEAVTRF4's formatting model CSECT IEAVTRP4), summarizes the bit flags in the SCB, and sends a return code to its caller indicating success or failure.

Entry from: IEAVTFMT via BLSQROUT and BLSQCFMT

Exit to: Caller

IEAVTRF5 – EED Control Block Format Module

Function: This module formats the EED (using IEAVTRF5's formatting model CSECT IEAVTRP5) and summarizes the bit flags. IEAVTRF5 sends back a return code to its caller indicating failure or success in formatting.

Entry from: IEAVTFMT via BLSQROUT and BLSQCFMT

Exit to: Caller

IEAVTRGS – IEAVSTA0 Branch Entry Addressing Mode Interface Routine

Function: This routine preserves the addressing mode of its caller and ensures that IEAVSTA0 is branch-entered in 31-bit addressing mode.

Entry from: Branch entry from another SVC routine.

Exit to: IEAVSTA0 in 31-bit addressing mode.

IEAVTRG1 – RTM1 Addressing Mode Interface Routine

Function: This routine is the addressing mode interface module between issuers of CALLRTM and module IEAVTRT1. It preserves the addressing mode of its caller and ensures that IEAVTRT1 is branch-entered in 31-bit addressing mode.

Entry from: Issuer of the CALLRTM macro.

Exit to: IEAVTRT1.

IEAVTRG2 – RTM2 Addressing Mode Interface Routine

Function: This routine is the addressing mode interface module for calls between RTM2 and modules below the 16-megabyte line.

Entry from:

IEAVTSKT via BASSM.
IEAVTMMT via BASSM.
Resource manages via BR 14.

Exit to:

IEAVTSKT via BSM.
IEAVTMMT via BSM.
IEAVTMTC via BSM.
IEAVTAS3 via BSM.
IEAVTRTC via BSM.

IEAVTRMC

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTRML – Installation Resource Manager List

Function: This CSECT can be replaced with a CSECT containing a list of modules to be called by task and address space termination. It is defined to look like a list with no entries.

IEAVTRRR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTRSA

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTRS0

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTRTC

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTRTD

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTRTE

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTRTF

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTRTM

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTRTR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTRTS

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTRTV

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTRT1

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTRT2

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR1A

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR1C

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR1F

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR1G

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR1I

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR1N

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR1R

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR1S

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR1X

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR10

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR2A

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR2C

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR2D

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR2E

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR2F

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTR2V

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager.*

IEAVTSAI – Dumping Services Address Space Initialization

Function: This module initializes the DUMPSRV address space, GETMAINs storage for the virtual storage buffer and the SMEW, and sets an Authorization Index (AX) of 1.

Entry from: IEEPRWI2.

Exit to: IEEPRWI2 to continue the full function address space (rc = 0) or to terminate the address space (rc = 4).

Called routines: IEEMB883

IEAVTSAS – DUMPSRV Address Space Create Routine

Function: IEEMB881 is invoked by this module to initiate the creation of the dumping services (DUMPSRV) address space.

Entry from: IEAVNP57 (DUMPSRV NIP RIM), IEAVTSDR (SDUMP and DUMPSRV memory termination resource manager calls IEAVTSAS to create a new DUMPSRV address space whenever the old DUMPSRV fails).

Exit to: Caller (IEAVNP57 or IEAVTSDR).

Called routines: IEEMB881, IEEMB883

IEAVTSBP

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager*

IEAVTSCB

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager*

IEAVTSDB – SVC Dump Disabled Interrupt Exit (DIE) Processor

Function: This routine protects critical SVC Dump resources through the use of disabled interrupt exits (DIEs). IEAVTSDB protects the summary dump real storage buffer (RSB), system non-dispatchability, and the suspended unit or work during suspend summary dump processing.

Entry point IEAVTSDB

Entry from: IEAVAD00, IEAVTSDT, IEAVTSDX, IEAVTSSD

Exit-Normal: Returns to the caller.

Called routines: IARXPLCK, IARXPRSB, IEAVRSTD, SETDIE

IEAVTSDC – SVC Dump Clean Up Routine

Function: This module cleans up SVC dump resources at the end of a dump.

- Sets the system dispatchable, if necessary.
- Sets tasks in this address space dispatchable.
- Issues page frees on any modules that were page-fixed.
- Frees the System Trace table, if obtained by SDUMP.
- Writes end-of-file on the dump data set
- Issues IEA984E for end-of-file on tape.
- Issues IEA911E (dump completed message).
- Issues IEA994E or IEA994A, if all dump data sets are full.
- DOM IEA994E, if dump data sets are now available.
- Calls IEAVTSD1 to start post-dump exit processing.

Secondary entry-point IEAVTSDE is called for partial cleanup of each address space of a multiple address space dump.

Secondary entry-point IEAVTSDA is called as an attention exit when the operator mounts a new dump tape in response to message IEA984E.

Entry from: IEAVAD00 and IEAVPSDT for entry IEAVTSDC. IEAVAD00 ($DESTAEX) and IEAVTSDT for entry IEAVTSDE. IEACVPST for entry IEAVTSDA.

Exit to: Caller.

Called routines: IEAVTSDP, IEAVTSDI.

IEAVTSDD – SVC Dump Data Set Initialization Routine

Function: This module parses the system parameters DUMP option and initializes the requested dump data sets and tapes. IEAVTSDD builds a queue of dump data set control blocks (SDDSQ) that is used by IEAVTSDH to select a dump data set.

Entry from: IEAVTSDI.

Exit to: Caller.

Called routines: NIP services - WTO, OPEN, MOUNT.

IEAVTSDF – SVC Dump GTF Trace Processor

Function: If TRT is a requested option in the SDUMP parameter list and GTF is active, this module invokes GTF to obtain the saved GTF trace records. It calls IEAVTSDP to write these records on the dump.

Entry from: IEAVAD00 and IEAVTSDT.

Exit to: Caller.

Called routines: IEAVTSDP.

IEAVTSDG – SVC Dump Global Storage Processor

Function: This module dumps the following system areas: SQA, CSA, global subpools requested by the caller, PSA, global storage ranges requested by the callers storage list, Nucleus, and the saved system trace table. IEAVTSDU is called to interface with global dump exits (ISGDSDMP for GRS). The system is set dispatchable after global storage has been dumped.

Entry from: IEAVAD00 and IEAVTSDT.

Exit to: Caller.

Called routines: IEAVTSD0, IEAVTSDU, IEAVTSDM, IEAVTVSM.

IEAVTSDH – SVC Dump Data Set Initialization

Function: This module chooses the dump data set for SVC dump. The SDUMP 4K buffer and the AMDPRDMP header record are written on the dump data set. If the console loop trace facility is available and active, it is dumped at this time. The range table to be used for dumping global storage is initialized with the addresses of system control blocks required in the dump.

Entry from: IEAVAD00, IEAVTSDT.

Exit to: Caller.

Called routines: IEAVTSDP, IEAVMSF1.

IEAVTSDI – SVC Dump Initialization RIM

Function: This module initializes system areas for SVC DUMP by doing the following:

- GETMAINs storage for SDUMP control blocks.
- Obtains the entry points of modules and saves the entry points for later use. Some modules are page-fixed.

Module Name	Function	Entry Point Saved
IEAVTSSD	Disabled Summary Dump	SMWKENTY and page-fixed
IEAVTSDR	Resource manager	CVTSDRM and page-fixed
IEAVTSPR	SDUMP setup routine	RTSDTSPR
IEAVTSDA	Attention exit for tape	DSVDATTN
IEAVTSSV	Suspend SUMMARY DUMP	SMWKTSSV
IEAVTSDT	Dump task	SMWKPFXB and

		page-fixed
IEAVTSDW	Summary dump writer	page-fixed
IEAVTSDO	SDUMP output routine	page-fixed
IEAVTSDH	SDUMP initialization	page-fixed
IEAVTVSM	VSM interface routine	page-fixed
ADYEXT	DAE symptom extraction routine	DSCEXT
ADYPRED	DAE predump processor	DSCPRED
IEAVTSDB	SDUMP timer die processor	RTCTSDBB

- Calls IEAVTSDD to initialize the SYS1.DUMP data sets

Entry from: IEAVNIPM.

Exit to: Caller.

Called routines: IEAVTSDD.

IEAVTSDL – SVC Dump Local Storage Processor

Function: This module adds address ranges to the SVC dump address range table for the following storage areas: LSQA, SWA, RGN (including LSQA and SWA), LPA, user storage ranges (specified with option LIST, STORAGE, or LISTA), and user requested local subpools (specified with option SUBPLST). Module IEAVTSDO is called to write the storage areas described by the address ranges in the address range table on the dump data set. This module always dumps the cross memory control blocks of the memory it is running in. (ET, LT, AT, and PCLINK stacks)

Entry from: IEAVAD00, IEAVTSDT.

Exit to: Caller.

Called routines: IEAVTSDO (output processor), IEAVTVSM (VSM interface module)

IEAVTSDM – SVC Dump System Trace Interface Routine

Function: This module interfaces with the system trace service routines. It performs the following three functions:

- Locates an RTM2 copy of the trace table.
- Compresses a trace table for unauthorized SYSMDUMPs.
- Dumps the saved trace table to the dump data set and frees the trace table when finished.

Entry from: IEAVAD00 for the locate and compress and IEAVTSDG for dumping the trace table.

Exit to: Caller.

Called routines: IEAVTSDP (entry in IEAVTSDO) to write records to the dump data set.

IEAVTSDO – SVC Dump Output Processor

Function: This module performs I/O to the dump data set.

Entry from:

IEAVTSDG via BALR to write out global storage.

IEAVTSDL via BALR to write out local storage.

Secondary entry-point IEAVTDSP:

IEAVTSDC via BALR to write out end-of-file.

IEAVTSSE via BALR to write out enabled summary dump.

IEAVTSDF via BALR to write out GTF records.

IEAVTSDH via BALR to write out the dump header record.

IEAVTSDM via BALR to write out system trace records.

IEAVTSDW via BALR to write out disabled summary dump.

Secondary entry-point IEAVTSEO:

IEEMB879 via BALR to write out Master Trace.

ISGDSDMP via BALR to write out GRS data.

Secondary entry-point IEAVTSVO:

IEAVTSDW via BALR to write out suspend summary dump only.

Exit to: Caller.

IEAVTSDR – SVC DUMP Resources Manager

Function: This module has three main functions.

1. FRR for the SRBs in a scheduled dump.
2. Memory termination resource manager for address spaces involved in an SDUMP or for the DUMPSRV address space.
3. Task termination resource manager for failing dump tasks (IEAVTSDT).

The following cleanup is performed by this module for SDUMP.

Sets the system and tasks dispatchable.

POSTs the caller of the SDUMP.

Frees up the real storage buffer (RSB).

Frees the trace table buffers acquired for the SDUMP.

Page releases the virtual storage buffer.

Resets the SDUMP caller on errors in SUSPEND summary dump processing.

The following is done for DUMPSRV memory termination.

Cleans up post exit processing resources.

Removes from the dump data set queue (SDDSQ) all data sets that were added to SDUMP with the DUMPDS command.

Restarts the DUMPSRV address space by linking to IEAVTSAS.

Entry from:

RTM as the FRR for SCHSRB in IEAVTSDX.

IEAVTMMT via BALR for all memory terminations.

IEAVTSKT via BALR for all task terminations.

Exit to: Caller.

Called routines: IARXPLCK, IARXPRSB, IEAVTSAS.

IEAVTSDS – SVC Dump Started Dump Task for DUMPSRV

Function: This modules is a never-ending task which controls the flow of SDUMP modules for dumping global storage.

Callers: IEAVTDSE and IEAVTDSV.

Exit-Normal: Returns to WAIT for the next dump request.

Exit-Error: SDSESTAE get control.

Entry-Point SDSESTAE recovers from unexpected errors suffered by mainline and cleans up it percolated to from IEAVTSD2's FRR.

Callers: RTM2.

Exit-Normal: Returns to RTM2 with retry/percolation indicator.

External-References: IEAVTSDH, IEAVTSDM, IEAVTSDO, IEAVTSDU, IEAVTSDW, IEAVTSD1, and IEAVTSEQ.

IEAVTSDT – SVC Dump Task (one in every address space, attached at address space creation time)

Function: This module performs SVC dump processing in each address space that is the target of a scheduled SVC dump request. It is awakened in an address space when the SRB routine, scheduled by IEAVTSDX, POSTs the waiting task. It calls the SVC dump service modules to dump the requested storage of the address space in which this routine is awakened.

Entry from: Attached during address space creation, it goes into a WAIT at DUMPREQ. To process a dump, IEAVTSDT is POSTed and receives control at label DUMPREQ. When processing is completed, IEAVTSDT loops back to label DUMPREQ and goes into another WAIT.

Exit to: None. This is a never ending task (as long as the address space exists).

Called routines: IEAVTSDB, IEAVTSDH, IEAVTSDW, IEAVTSDG, IEAVTSDL, IEAVTSSE, IEAVTSDF, IEAVTSDU, IEAVTSDC.

IEAVTSDU — SVC Dump Exit Interface Routine

Function: This module allows component dump exit routines to be invoked by SVC dump in order to add component-related data to the dump. A general parameter list is passed to the exit routine. The exits to be given control are specified in the dump exit list contained in the data csect IEAVTSXT. The following exits currently exist:

The Global Resource Sharing (GRS) SVC Dump exit (ISGDSDMP).

The Master Trace SVC Dump exit (IEEMB879).

Entry from:

IEAVTSDT via BALR at the end of the dump to IEEMB879.

IEAVTSDG via BALR after dumping global storage to process ISGDSDMP.

Exit to: Caller.

Called routines: IEEMB879, ISGDSDMP.

IEAVTSDW — SVC Dump Summary Dump Writer

Function: This module puts the data in the summary dump real storage frames into AMDPRDMP format and invokes module IEAVTSDP to write these constructed records on the dump data set. As the contents of each frame is written on the dump data set, the frame is returned to RSM using routine IARXPRSB.

Entry from: IEAVTSDT via BALR.

Exit to: Caller.

Called routines: IEAVTSDP, IARXPLCK, IARXPRSB.

IEAVTSDX — SVC Dump Branch Entry and Schedule Dump Routine

Function: This module schedules an SVC dump in the address space(s) specified by the caller supplied keywords, ASID, ASID list, TYPE=XMEM, TYPE=XMEME, LISTA, SUBPLST and LISTD. The address space(s), including home, in which the resident dump task is active.

Caller IAVTSGL.

Exit-Normal: Return to the caller (BSM).

Exit-Normal: Abend caller, program check.

Entry-Point IEAVTSDX cleans up after an error which occurred while attempting an SVC dump or while performing a summary dump. Retry is attempted in all cases. This FRR is always entered in home mode (PASID=SASID=HOME). Retry is never attempted for bind break abends or on retry errors. On retry cases, RTM is requested to set the cross memory mode to the time of the error. Bind break abends will cause RTM to re-enter the FRR which will then percolate. The SDWA VRA is filled with diagnostic data and the SDWA is recorded to LOGREC.

Callers: None

Exit-Normal: Return to RTM1 from schedule FRR.

External-References: IARXPRSB, IARXPLCK, IEAVTSSD, IEAVTSPR, ADYPRED, and IEAVTSDB.

IEAVTSD1 — SVC Dump Post-Dump Exit Startup Processor

Function: Invokes the post-dump exit process in the DUMPSRV address space.

- Initializes GETMAINed areas for post exit processing.
- ENQs the GETMAINed areas on the post exit processing queue.
- Cross memory post of IEAVTDSV to start post exit processing for this dump.
- DEQs from the post exit resource.

Exit from IEAVAD00 and IEAVTSDT for dump suppression path. IEAVTSDC for dump taken path.

Exit to: Caller.

Called routines: Cross memory post of IEAVTDSV.

IEAVTSD2 — SVC Dump SRB to Schedule the Copying SRB's

Function: IEAVTSD2 is a SRB scheduled by IEAVAD00 or IEAVTSDX. IEAVTSD2 posts the Started Dump Task. It sets up the invironment for the SRB's IEAVTSD3 and IEAVTSD4 to copy global data to the one-to-one buffer in DUMPSRV's dataspace.

Entry-Point IEAVTSD2:

Callers: Dispatcher.

Exit-Normal: Exit to: the dispatcher.

Entry-Point RETRYFRR records and cleans up IEAVTSD2's resources. It retries or percolates to PERC-FRR, releasing the local lock.

Callers: RTM.

Exit-Normal: Percolates to IEAVTSDS's recovery.

External-References: STATUS, IEAVTSDB, IEAVTSDG, POST, and IEAVTSMG.

IEAVTSD3 – Routine to Copy a Range of Global Storage into DUMPSRV's Scope (single) Dataspace

Function: This module is a SRB scheduled by IEAVTSD2 or by IEAVTSD3, running in DUMPSRV's address space. It is passed a range and a pointer to a page within that range or it selects a range from the global range table. IEAVTSD3 copies the data into DUMPSRV's scope (single) data space.

Entry-Point IEAVTSD3 copies a range of storage into DUMPSRV's data space.

Callers: Dispatcher.

Exit-Normal: Exit to: the disptacher.

Entry-Point IEAVTSRM is the RMTR to free the SRB storage after a PURGEDQ has occurred.

Callers: Dispatcher.

Exit-Normal: Exits to the dispatcher.

External-References: IARQDUMP.

IEAVTSD4 – Routine Scheduled to Copy a sub-range or Common Storage to DUMPSRV's Dataspace

Function: This module copies a sub-range of common storage into DUMPSRV's data space storage. IEAVTSD3 determined that this sub-range of storage was located on auxiliary storage. It loaded the sub-range into real storage and scheduled IEAVTSD4, an SRB, to do the copying.

Entry-Point IEAVTSD4 copies sub-range of global storage into data space.

Callers: Dispatcher.

Exit-Normal: Exit to: the dispatcher.

Entry-Point IEAVTS4F is the recovery routine for IEAVTSD4.

Callers: RTM.

Exit-Normal: Percolates or retries.

External-References:

- IEAVTRMT
- IEAVTSMG

IEAVTSD5 – SDUMP Set Caller Non-Dispatchable Routine

Function: This module sets the SDUMP caller's tasks non-dispatchable, resets the caller, and schedules IEAVTSD2 to get the dump started. This prevents the caller's storage from being freed at task termination before the dump gets to process it.

Entry-Point IEAVTSD5 sets the caller non-dispatchable, reset the caller, and schedules IEAVTSD2.

Callers: IEAVTSDX.

Exit-Normal: Returns to the dispatcher.

Exit-Error: Unexpected unretryable error leading to percolation from FRR routine.

Entry-Point SD5FRR serves as FRR for IEAVTSD5. They record and retry.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: Abnormal termination of FRR routine.

External-References:

- IEAVESRT
- IEAVTSDB
- STATUS

IEAVTSEL – SVC Dump List of Post Dump Exits

Function: This is a non-executable load module that contains a list of the SVC dump post exit processors in the order in which they are to be executed. IEAVTSEL is loaded and read by IEAVTDSV when the DUMPSRV address space is created. See IEAVTSEP for use of post exits.

IEAVTSEP – SVC Dump Post Exit Processing Routine

Function: This module is ATTACHed as a subtask in the DUMPSRV address space after the completion of every SVC Dump or SYSMDUMP. Its function is to build a parameter list, containing a pointer to a copy of the dump header record, and call the dump exits which were specified in the post dump exit list (IEAVTSEL). After all the exits in the list have been called, storage is freed and the task is terminated.

Entry from: Attached by IEAVTDSV at the end of each dump.

Exit to: Exit prolog.

Called routines: Whatever exits are specified in IEAVTSEL.

IEAVTSFR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager*

IEAVTSGL — SVC Dump Glue Routine

Function: The purpose of this module is to support branch entry to SDUMP in either 24 or 31 bit addressing mode. If the caller is in 24 bit mode, the high order byte of register 1 and register 13 are cleared to prevent addressing exceptions in IEAVTSDX when it runs in 31 bit addressing mode. The callers registers are saved before passing control to IEAVTSDX.

Entry from: IEAVAD00 on a scheduled dump request on SVC 51. Issuer of the SDUMP macro with BRANCH=YES.

Exit to: IEAVTSDX. (Note: IEAVTSDX does not return to IEAVTSGL.)

Called routines: IEAVTSDX.

IEAVTSFR — SETFRR

Function: The SETFRR macro calls this module to update the FRR stacks. Depending on the request, IEAVTSFR adds or deletes a current stack entry, purges all the entries from the current stack, flushes all the entries from the normal stack, or flushes all the entries from the normal stack and then adds an entry to this stack.

Entry from: SETFRR macro.

Exit to: Caller.

IEAVTSIG — SLIP PER RISGNL Routine

Function: This module adjusts the PSW PER indicators in the I/O, external, and SVC new PSWs, and loads the PER control registers (9-11) with appropriate values when PER monitoring is being activated or deactivated.

Entry from: IEAVTGLB via BALR.

Exit to: Caller.

IEAVTSIN — FRR Stack Initialization

Function: IEAVTSIN is an executable macro included by IEAVNIPO and IEEVCPR to perform FRR stack initialization for a given processor.

Entry from: This inline expansion is included by IEAVNIP0 (during system initialization) and by IEEVCPO (during vary processor processing).

Exit to: Caller.

IEAVTSKT

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Recovery Termination Manager*

IEAVTSLB — SLIP Action Processor — Part 2

Function: This module routes control to the SLIP trap match routines ((IEAVTSL1 or IEAVTSL2) depending on the parameters found in the SCVA)) and processes the MATCHLIM and PRCNTLIM keywords.

Entry from: IEAVTSLP.

Exit to: IEAVTSLE.

Called routines: IEAVTSL1, IEAVTSL2.

IEAVTSLC — SLIP/CMSET Intercept Interface Routine

Function: When SLIP/PER is active, this module intercepts CMSET SET and RESET macros and calls IEAVTSSH to determine if PER needs to be adjusted.

Entry from:

For entry-point IEAVTSLC: Any issuer of the CMSET SET macro when there is an active SLIP trap for that macro.

For entry-point IEAVTSL8: Any issuer of the CMSET RESET, CHKAUTH=NO macro when there is an active SLIP trap for that macro.

For entry-point IEAVTSL9: Any issuer of the CMSET RESET, CHKAUTH=YES macro when there is an active SLIP trap for that macro.

Exit to: Caller.

IEAVTSLE – SLIP Action Processor – Part 3

Function: This module initiates message processing requested by IEAVTSLP when a trap has been automatically disabled. IEAVTSLE also removes the recovery environment established by IEAVTSLP, and returns to the caller of IEAVTSLP.

Entry from: IEAVTSLB, IEAVTSLR.

Exit to: The caller of IEAVTSLP via BSM.

IEAVTSLP – SLIP Action Processor – Part 1

Function: This module receives control when SLIP is active from either an RTM routine as part of normal processing, or from IEAVTPER after a PER interruption. IEAVTSLP establishes error recovery to match the environment of the caller, serializes the SCE chain, and passes control to IEAVTSLB to continue SLIP action processing.

Entry from: IEAVTRTS, IEAVTRT2, IEAVTRTM, or IEAVTPER via BASSM.

Exit to: IEAVTSLB.

IEAVTSLR – SLIP Processor Recovery

Function: This module provides the FRR recovery for the SLIP processor modules (IEAVTSLP, IEAVTSLB, IEAVTSLE, IEAVTSL1, IEAVTSL2, IEAVTSLS) and the address conversion routine (IEAVTADR) when it is called by SLIP. If a recoverable error has occurred, this module attempts a retry at an appropriate point in the SLIP processor module. If the error is not recoverable, IEAVTSLR requests percolation.

Entry from: IEAVTRTS (FRR router).

Exit to: IEAVTRTS.

IEAVTSLS – SLIP Processor Service Routine

Function: This module is called to: determine the system mode, issue a DUMP, or schedule IEAVTGLB to turn off PER monitoring.

Entry from: IEAVTSLB, IEAVTSL1, IEAVTSL2.

Exit to: Caller.

IEAVTSL1 – SLIP Trap Matching Routine – Part 1

Function: This module determines whether the event qualifier keywords specified on the SLIP command match the current system conditions. IEAVTSL1 processes the ADDRESS, ASID, COMP, ERRTYP, JOBNAME, JSPGM, LPAMOD, MODE, RANGE, and ASIDSA event qualifier keywords.

Exit to: Caller.

IEAVTSL2 – SLIP Trap Matching Routine – Part 2

Function: This module: determines whether the event qualifier keywords specified on the SLIP command match the current system conditions (IEAVTSL2 processes the DATA and PVTMOD event qualifier keywords). When a match is found for all event qualifier keywords, IEAVTSL2 is called to perform the action specified in this trap.

Entry from: IEAVTSLB.

Exit to: Caller.

IEAVTSPR – SVC Dump Parameter Routine

Function: This module is called to perform four different functions.

- Initializes SDUMP storage, copies the callers parameter list and associated lists into SDUMP storage, and merges the requested dumping options with the CHNGDUMP options.
- Validates the callers DCB, if the SDUMP DCB option was specified.
- Parses the callers DCB list and separates the global and local subpools for dumping at different times.
- Adds the ASIDs specified in the ASID, ASIDLST, and LISTS options to the RTCT ASID table.

Entry from: IEAVAD00, IEAVTSDX.

Exit to: Caller.

IEAVTSRB – SVC Dump SRB Routine

Function: IEAVTSRB is the SRB scheduled by IEAVTSDS. The SRB will set the address space environment so that a dump can be taken by setting the tasks of the address space non-dispatchable. The SRB will then activate the resident dump task to dump storage in this address space.

Entry-Point IEAVTSRB is the SRB scheduled by IEAVTSDS. The purpose is to initiate the dumping process. The SRB will set the address space environment so that a dump can be taken by setting the tasks of the address space non-dispatchable. The SRB will then activate the resident dump task.

Callers: None

Entry-Point SRBFRR1 is a recovery from a failure in the SVC dump SRB Routine.

Callers: None

Exit-Normal: Return to RTM1.

External References:

- IEAVPT00
- IEAVTSDB

IEAVTSR1

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis. Recovery Termination Manager*

IEAVTSSD – SVC DUMP Disabled Summary Dump Processor

Function: Performs suspend and disabled summary dump processing. For suspend summary dumps, captures volatile system data in the real storage buffer and calls IEAVTSSV to complete the summary by capturing volatile pageable data in the virtual storage buffer. For disabled summary dumps, captures volatile, page-in system data in a real storage buffer. This data represents the state of the system when the SVC dump was invoked.

Callers: IEAVTSDX.

Exit-Normal: Return to caller - module IEAVTSDX.

Entry-Point IEAVTSD0 moves the contents of a real storage buffer frame, which is being stolen back by RSM, to another frame.

Callers: RSM.

Exit-Normal: Return to caller.

Exit-Error: Return to caller.

Entry-Point IEAVTSSS allows CSECTs outside of IEAVTSSD to use the subroutine logic instead of IEAVTSSD. A function code is passed in the SMWK to tell this entry point what subroutine of IEAVTSSD is to be called.

Callers: IEAVTSSM and IEAVTSSV.

Exit-Normal: Return to caller.

Exit-Error: None.

External-References:

- SETFRR
- SETLOCK
- IARXPRSB
- IARXPLCK
- IEAVTSSM
- IEAVTSSV

IEAVTSSE – SVC Dump Enabled Summary Dump Processor

Function: For non-branch entries to SVC DUMP, this module writes a summary dump on the dump data set. This summary dump includes the following storage areas:

- Storage areas specified with the SUMLIST or SUMLSTA keywords on the SDUMP macro.
- For each TCB in the TCB chain, the following is dumped:

  The RTM2 work area, if it exists.

  4K of storage before and after the addresses in the registers and the PSW at the time the error is stored in the RTM2 work area.

  Any PCLINK stack elements pointed to by the XSB (pointed to by RBXSB).

Entry from: IEAVTSDT, IEAVAD00

Exit to: Caller.

Called routines: IEAVTSDP.

IEAVTSSH – SLIP Space Switch Handler

Function: When a unit of work changes the address space in which it is executing, this module adjusts the PER bit in the PSW according to the criteria established on the SLIP command.

Entry from:

For entry-point IEAVTSSH: IEAVTPER when a space switch interrupt occurs.

For entry-point IEAVTSS1: IEAVTSLC (CMSET SET and RESET).

Exit to:

For entry-point IEAVTSSH: IEAVTPER using a BR 14.

For entry-point IEAVTSS1: IEAVTSLC using a BSM.

IEAVTSSM — SVC Dump Summary Dump Data Move Routine

Function: For disabled and suspend summary dumps, this module moves volatile system data to the real storage buffer or the virtual storage buffer located in the DUMPSRV address space.

Entry from: IEAVTSSD, IEAVTSSV.

Exit to: Caller.

IEAVTSSV — SVC Dump Suspend Summary Dump Processor

Function: Performs suspend summary dump processing. This modules moves data important to a recovery routine to a virtual storage buffer located in the dumping services address space (DUMPSRV).

Callers: IEAVTSSD.

Exit-Normal: Returns to IEAVTSSD.

Exit-Error: None.

External-References:

- IEAVTSSM
- IEAVTSSS
- IEAVELK

IEAVTSSX — Space Switch Extension

Function: This module adjusts the PER bit in the PSW for RTM1. IEAVTSSX performs the same function as IEAVTSSH, but is tailored to meet the needs of RTM1.

Entry from: IEAVTRTS (RTM1) using a BALR.

Exit to: IEAVTRTS using a BR 14.

IEAVTTKN

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Task Management*

IEAVTSXT — SVC Dump Component Dump Exit List

Function: This is a non-executable module that contains a list of component related dump exits and flags that indicate when they are to be given control. The list currently contains IEEMB879 (Master Trace Exit) and ISGDSDMP (GRS Exit).

Entry from: This module is loaded and read by IEAVTSDI.

IEAVTSYM — Display Symptom Dump

Function: This module displays the registers, PSW, ABEND code, and ERRDRID from the RTM2WA pointed to by TCBRTWA. It also finds the active module pointed to by the PSW, if applicable. The offset into the module is calculated and 6 bytes prior to the PSW, and the data where the PSW points, is dumped.

Entry from: IEAVTABD via BALR.

Exit-Normal: To caller.

Exit-Error: To caller.

IEAVTVSM — SVC Dump VSM Interface Routine

Function: This module does all the interfacing with VSM to obtain storage ranges to be dumped for the following options: RGN, LSQA, SWA, CSA, SQA, and SUBPLST.

Entry from: IEAVTSDG, IEAVTSDL.

Exit to: Caller.

IEAVVCRA — Communication Task Console Attention Processor

Function: The console attention processor receives control as a result of someone pressing the console's attention key. Pressing the attention key causes an attention interrupt. As a result of the attention interruption, this routine posts the communication task.

Entry from: IEACVPST.

Exit to: The POST status routine of the IOS handler via a BR 14 instruction.

IEAVVCRX – External Interrupt Key Processor

Function: This module receives control from the external first level interrupt handler to post the communication task.

Entry from: IEAVEEXT.

Exit to: The external first level interrupt handler by branching to the address in register 2.

IEAVVCTR – SVC 72 Router

Function: IEAVVCTR determines the appropriate device support processor (DSP) to process the communications task service.

Entry from: Issuer of SVC 72.

Exit to:

- IEAV1443 for a 1403 printer or 3211 printer supported as a console.
- IEAV2540 for a 2540 card reader punch, 2501 card reader, 3505 card reader, or 3525 card punch supported as a console.
- IEEC2740 for a 2740 communications terminal supported as a console.
- IEECVETW for a 3284/3286 printer supported as a console.
- IEECVET1 for a display device supported as a console.

IEAVVJST

Function: The diagnosis information for this module is in *MVS/ESA Component Service: Supervisor Control and Dispatcher*

IEAVVMCH – Vector Service Routine

Function: This module saves or invalidates vector environment or validates vector registers.

Entry from: Machine check handler or alternate CPU recovery

Exit to: Caller

IEAVVRP2 – Reply Processor - Stage 2

Function: This module checks whether the text of a user's buffer area is valid. If the buffer is valid, IEAVVRP2 moves the reply text into the buffer.

Entry from: IEECB811 (via an SRB).

Exit to: Dispatcher.

IEAVVSR – Vector SLIH Recovery Routine

Function: This module cleans up system resources obtained by the vector SLIH.

Entry from: RTM

Exit to: IEAVEVS2 in the vector SLIH for retry

IEAVVSRB – Create Vector Environment SRB Routine

Function: This module creates a task's vector environment or cleans up the vector SRB routine's system resources.

Entry from:

For entry point IEAVVSRB - SRB dispatcher via LPSW

For entry point IEAVVSR2 - RTM

Exit to:

For entry point IEAVVSRB - IEAVETCL

For entry point IEAVVSR2 - dispatcher at entry point DS0SRBRN for retry

IEAVVWTO – Single Line WTO and WTOR Service Routine

Function: This module performs the processing for a single line WTO or WTOR. It receives control from module IEAVM600 and builds a dummy WQE and, for a WTOR, an ORE. Processing for a write-to-Programmer (routing code 11 specified) is handled by IGC0203E (IEEJB840).

Entry from: IEAVM600.

Exit to: IEAVM600

Error exit: When a WTO message has a length of zero bytes, the service routine branches to the SVC first level interrupt handler at the address in register 14. For all other errors, IEAVVWTO gives control to RTM via an ABEND macro. RTM terminates the issuer of the WTO or WTOR macro with completion code X'D23'

Called routines: IGC0203E, IEAVH600, IEAVM700, IEFJSREQ, IEAVX600

IEAVWSAT – Global Work Save Area Vector Table and Global, Local,and CPU Templates

Function: Contains the GLOBAL work save area vector table, the template for this table and the templates for the COPY and local tables (mapped by the IHAWSAVT macro).

Callers: This modules is not executable.

Exit to: N/A

IEAVXAAC

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXABE

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXALA

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXALD

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXALE

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXALR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXALS

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXAPM

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXART

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXDOM – DOM Service Routine

Function: Services the Delete Operator Message (DOM or SVC 87) service request. Builds a DOMCB and posts the communications task to process the DOMCB and delete the message.

Entry Point IEAVC701

Callers: SVC FLIH

Exit-Normal: Return to caller

Exit-Error: IEAVMFRR, via the RTM

External-References:

- XMPOST - to notify the comm task to process a DOMCB
- IEAVH600 - WQE service routine

IEAVXD00

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXD10 – ARCHECK Service Router

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXD11 – Access Register Check Service for TCB

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXD12 – Access Register Check Service for SRB

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXD13 – Access Register Check Service for a Dump Record

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXD14 – Access Register Check Service for an Access List

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXD15 – ARCHECK Translate and Output Processing

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXD16

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXECO

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXECR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXEDE

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXEDI

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXEPM – Program Call Nucleus Entry-Point Search Module

Function: When called at entry-point IEAVXEPM, this module searches the nucleus entry-point table (IEAVXNET) for a specified entry-point, and returns the entry-point address to the caller. IEAVXEPM also receives control at entry-point IEAVXABE when a program call (PC) instruction references a default entry table entry. It issues an X'0D6' ABEND.

Entry from:

> For entry-point IEAVXEPM: IEAVXMAS during PC/AUTH address space initialization, IEAVXECR when creating an entry table.
>
> For entry-point IEAVXABE: Branched to from a PC instruction.

Exit to:

> For entry-point IEAVXEPM: Caller.
>
> For entry-point IEAVXABE: RTM via an ABEND macro.

Error exit RTM via an ABEND macro.

IEAVXLFR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXLRE

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXMAS

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: System Initialization*

IEAVXMCM

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXMIN – Memory Initialization Routine for PC/AUTH Services

Function: During address space initialization, IEAVEMIN gives IEAVXMIN control in SRB mode. IEAVXMIN connects the address space being initialized to the system authorization table (SAT) and system linkage table (SLT). It also sets the address space's authorization index (AX) to zero and indicates that the address space can be accessed in cross memory mode.

Entry from:

For entry-point IEAVXMIN: IEAVEMIN.
For entry-point XMINFRR: RTM.

Exit to: Caller.

IEAVXNEP – Nucleus Entry-Point Table Module

Function: This module contains a table (IEAVXNET) that includes:

- The names of routines in the nucleus that can be entered using a program call (PC) instruction.
- A VCON for each routine's entry-point address.

Reference from IEAVXMAS (described in *Component Diagnosis: System Initialization*) and IEAVXECR.

Exit to: Not applicable (This module does not contain executable code.)

IEAVXPAM

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXPCR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXREX

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXRFE

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXSEM

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: System Initialization*

IEAVXSET

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVXSFM – System Function Table Module

Function: This module contains the system function table (SFT), which contains pre-allocated PC numbers for system-wide program call services. This module contains no executable code.

IEAVXSTK

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: PC/AUTH Service Routines*

IEAVX600 – WTO User Exit Interface Routine

Function: This routine makes modifications to the WTO queue element (WQE) associated with the WTO message.

Entry point IEAVX600

Entry from: IEAVMWTO, IEAVVWTO

Exit-Normal: Returns to the caller.

Entry point XITABEND

Purpose XITABEND is an ESTAE recovery routine that provides diagnostic information for WTO user exit abends.

Entry from: RTM.

Exit-Normal: Returns to RTM.

Called routines: IEAVMXIT, any specified WTO user exit.

IECDAFMT – Data Management Control Block Formatter Mainline

Function: This module locates the data management control blocks associated with the TCB passed in the parameter list and determines if they are to be formatted. If they are, it calls the appropriate control block formatter for each data management control block.

Entry from: IEAVADIN or AMDPRUIM.

Exit to: Caller.

IECDAFT1 – Data Management Control Block Format Module

Function: This module formats either the DEB, DCB, or IOB passed in the parameter list depending on which entry is called. The DCB formatter determines the access method type and formats the DCB accordingly. It also returns this information to the caller for use in locating the IOBs. The IOB formatter also uses the access method information to format the IOB (ICB,LCB). All entry-points send a return code to the caller indicating success or failure in formatting. In addition, the DCB formatter sets a return code indicating failure if the access method does not have IOBs to be formatted.

Entry from: IECDAFMT at entry points IECDADCB, IECDADEB, IECDAIOB, and IECDAFT1.

Exit to: Caller.

IECDPRFX – SIO Exit for 3380 Devices with 3880 Speed Matching Buffer

Function: This module performs the following functions:

1. It builds a channel program prefix, if necessary.
2. It invokes a routine to start the channel program

Callers: IOSVSSCH

Exit-Normal: Returns to caller.

External-References: Start Channel Program Service Routine (located via the IOCSCPE field in the IOCOM).

IECDPRF2 – SIO Exit for 3350P Devices

Function: This module performs the following functions:

1. It selects an exposure to use in starting I/O.
2. It builds a channel program prefix, if necessary.
3. It invokes a routine to start the channel program.

Callers: IOSVSSCH.

Exit-Normal: Returns to the caller.

External-References: IOSVSMGR, Start Channel Program Service Routine (located via the IOCSCPE field in the IOCOM).

IECIOFMT – Telecommunications Device Descriptor Table

Function: This module, called as a TCB exit from AMDPRDMP and by the IO verb under SNAP/ABEND, locates the IOS control blocks that are to be formatted and calls the appropriate formatter.

Callers: IEAVAD01 and AMDPRDMP.

Exit-Normal: Return to caller.

Exit-Error: Return to caller.

External-References: ACCESS, IECIOEXD, IECIOUCB, and PRINT.

IECIOFT1 – IOS Control Block Format Module

Function: This module formats the control block whose address is passed by the calling routine.

Callers: None

Exit-Normal: Return to caller.

Exit-Error: Return to caller.

External-References:

- IPCS service routines (access, format, print)
- IOSVFMTY

IECLMSGC – TP Message Exit

Function: IECLMSGC fills in the sections of the message for teleprocessing devices and issues the message with a WTO macro.

Callers: IGE0025C.

Exit-Normal: Returns to the caller.

External-References: None.

IECLMSGD – DASD Message Exit

Function: IECLMSGD fills in the sections of the message for the direct-access devices and issues the message with a WTO macro.

Callers: IGE0025C.

Exit-Normal: Returns to the caller.

External-References: None.

IECLMSGG – Graphics Message Exit

Function: IECLMSGG fills in the sections of the message for graphics devices and issues the message with a WTO macro.

Callers: IGE0025C.

Exit-Normal: Returns to the caller.

External-References: None.

IECLMSGM – MSC Message Exit

Function: IECLMSGM fills in the sections of the message for the mass storage controller devices and issues the message with a WTO macro.

Callers: IGE0025C.

Exit-Normal: Returns to the caller.

External-References: None.

IECLMSGT – TAPE Message Exit

Function: IECLMSGT fills in the sections of the message for tape devices and issues the message with a WTO macro.

Callers: IGE0025C.

Exit-Normal: Returns to the caller.

External-References: None.

IECLMSGU – Unit Record Message Exit

Function: IECLMSGU fills in the sections of the message for unit record devices and issues the message with a WTO macro.

Callers: IGE0025C.

Exit-Normal: Returns to the caller.

External-References: None.

IECTCATN – Channel-To-Channel Device Attention Routine

Function: This module processes attention status on a channel-to-channel device.

Callers: IECVPST.

Exit-Normal: Returns to the Post Status routine.

Entry-Point IECTCQSC: Resets the quiesce level and restarts pending I/O requests.

Callers: Users of the service routine.

Exit-Normal: Returns to the address in register 14.

External-References: A routine which has its address in UCBCTCAL is called to notify the owner of the channel-to-channel device that an attention status has been raised.

IECVDDT2 – Telecommunications Device Descriptor Table

Function: This module provides the device dependent descriptor table for the following devices and those functionally equivalent:

Telecommunications

1030
1050
1050X
115A
2265
2740
2740X
2741P
2741C
3791L
7770
83B3
BSC1
BSC2
BSC3
TWX
WTTA

Callers: None.

External-References: None.

IECVDDT4 – Display Devices Device Descriptor Table

Function: This module provides the device descriptor table for the following devices and those functionally equivalent:

- 2250
- 3270
- 3277
- 3278
- 3279
- 3284

- 3286

Callers: None.

External-References: None.

IECVDDT5 – Unit Record Device Descriptor Table

Function: This module provides the device descriptor table for the following devices and those functionally equivalent:

- 1403
- 2501
- 2540P
- 2540R
- 3505
- 3525
- CTC

Callers: None.

External-References: None.

IECVDDT7 – Control Unit Device Descriptor Table

Function: This module provides the device descriptor table for the following devices and those functionally equivalent:

- 3704
- 3705

Callers: None.

External-References: None.

IECVDPTH – IOS Dynamic Pathing Module

Function: This module establishes or removes the path groups for devices supporting the dynamic pathing feature.

- During master scheduler initialization, this module establishes path groups for all on-line devices having the dynamic pathing feature.
- When a device is varied on-line, this module establishes a path group containing all physical and logical paths.
- When a device is varied off-line, this module disbands the path group.
- When a path is varied on-line, this module adds the path to the path group for the device.
- When a path is varied off-line, this module removes the path from the path group.
- When a channel path is being recovered, this module re-establishes the path in the path group if the logical path mask is on for that channel path.
- When a device is being boxed, this module disbands the path group.

Callers:

In the TCB mode: IECVIOSI, IECVIOPM, IEE3103D, IEFAB488, IEFAB429.

In SRB mode: IOSVVARY, IOS channel recovery modules, any module boxing a device.

Exit-Normal: Returns to the caller.

Entry-Point DPTHDPTH: Retries the switch to SRB mode after an error that caused exit to the ESTAE routine.

Callers: RTM.

Entry-Point DPTCBEX: Retries after an error that caused exit to the ESTAE routine if exit processing was not complete.

Callers: RTM.

Entry-Point DPTHTCB: Performs special initialization for TCB-mode entry before continuing with dynamic pathing function.

Callers: Dispatcher via SRB scheduled by IECVDPTH.

Exit-Normal: Processing continues at label DPTHSRB in IECVDPTH.

Entry-Point DPTHFUNC: Retries the dynamic pathing function after an error that caused exit to the FRR routine.

Callers: RTM.

Entry-Point ISSRTRY: Retries the dynamic pathing function after an error that caused exit to the FRR routine.

Callers: RTM.

Entry-Point DPTHRTRN: Updates the UCB according to the results of the Modify Subchannel and the dynamic pathing function requested.

Callers: Dispatcher via a SRB scheduled by IOSVMSCQ.

Exit-Normal: Returns to the caller.

Entry-point DPTHEXIT Retries after an error that caused exit to the FRR routine if module exit processing could not complete.

Callers: RTM.

Entry-point DPTHRTR1 Processes purge requests for the SRB scheduled by IECVDPTH.

Callers: PURGEDEQ.

Exit-Normal: Returns to the caller.

Exit-Error: Returns to the caller.

Entry-point DPTHRTR2 Processes purge requests for the MSCH SRB.

Callers: PURGEDEQ.

Exit-Normal: Returns to the caller.

Exit-Error: Returns to the caller.

Entry-Point DPTHESTA: Recovers from an unexpected error that caused an exit to RTM, while executing in TCB mode. Attempts to retry.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: Returns to RTM.

Entry-Point DPTHFRR: Recovers from an unexpected error that caused an exit to RTM, while executing in SRB mode. On first entry, attempts retry. If entered a second time, percolates the error.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: Returns to RTM.

External-References: IOSRSRBR, IOSVDPDR, IOSVMSCQ.

IECVERPL – ERP Load and XCTL Routine

Function: This module passes control to one of several possible modules:

- A non-direct access ERP.
- The ERP message writer.
- Dynamic device reconfiguration.
- An ERP service routine.
- Another ERP load module.

Callers: IEAVEEE0.

Exit-Normal: To one of the following:

- A non-direct access ERP.
- The ERP message writer (IGE0025C).
- The dynamic device reconfiguration routine (IGE0660A).
- IOS post status routine (IECVPST) via SVC 15, then SVC 3.

Exit-Error: To RTM.

Entry-Point IECXTLER: Passes control to an ERP service routine or to another ERP load module.

Callers: Non-direct access ERPs and ERP service modules.

Exit-Normal: To one of the following:

- A non-direct access ERP.
- An ERP service routine.
- IOS post status (IECVPST) via SVC 15, then SVC 3.

Exit-Error: To RTM.

Entry-Point ERPLESTA: Recovers from an unexpected error that caused exit to RTM.

Callers: RTM (for ESTAE requests from this module).

Exit-Normal: To RTM.

Exit-Error: To RTM.

External-References: IEAQCDSR, IEAVVMSR.

IECVEXCP – EXCP Processor for SVC 0 (EXCP) and SVC 114 (EXCPVR) Requests

Function: This module processes EXCP and EXCPVR I/O requests. This module handles the initiation of a caller's request to IOS, handles the I/O interruption from IOS, and passes the results back to the caller through its appendages.

Entry-Point IGC000: Process EXCP (SVC0) requests.

Callers: Issuer of SVC 0.

Exit-Normal: Return to SVC type 1 exit.

Exit-Error: To RTM.

External-References: IARPSIV, IEAV0PT02, IEASMFEX, IECVQCNT, IECVRCHN, IECVEXSM, IECVTCCW, IECVEXFR, IFGDEBVR, and IOSVHCRS.

Entry from: Issuers of SVC 0, SVC 114, and SVC 92; IOS Post Status (IECVPST); DASD ERP (IECVDERP); EXCP Purge Routine (IECVEXPR); EXCP Functional Recovery Routine (IECVEXFR).

Exit to: Caller.

IECVEXFR – EXCP Functional Recovery Routine (FRR)

Function: To recover from an unexpected error that caused exit to RTM. It also performs abend processing as a result of EXCP issuing an SVC 13 (D).

Entry-Point IECVEXFR: RTM linkage to FRR.

Callers: RTM.

Exit-Normal: Return to caller.

External-References: IECVSMFR, IECVTCFR, IECVXTRM, and IECVX025.

IECVEXPR – EXCP Processor's Purge and Restore Routines

Function: IECVEXPR controls and manages the EXCP processor purge and restore functions. Four entry points are provided for callers to interface with the EXCP processor for purge and restore functions.

Entry-Point IECVXPUR+0: Accepts IOS purge function from module IGC0001F.

Callers: IOS purge module IGC0001F.

Exit-Normal: Return to caller.

Entry-Point IECVXPUR+4: Receives control from the IOS SVC 33 function module IGC0003C.

Callers: IOS I/O halt module IGC0003C.

Exit-Normal: Return to caller.

Entry-Point IECVXRES: Receives control from the IOS SVC 17 function module IGC0001G.

Callers: IOS I/O restore SVC module IGC0001G.

Exit-Normal: Return to caller.

Entry-Point IECVRCHN: Builds an EXCP to restore IOB chain.

Callers: Entry point IECVXPUR or IECVEXCP.

Exit-Normal: Return to caller.

Entry-Point IECVEXCL: Dequeues IOBs from the IOB restore chain.

Callers: RTM module IEAVTAS3.

Exit-Normal: Return to caller.

External-References: IEA0PT02, IECVSMLF, IECVSMLG, IECXTRM, and IECVX025.

IECVEXSM – EXCP Storage Manager

Function: IECVEXSM controls and manages the two control block sizes (RQE and large blocks) that the EXCP component requires in order to provide I/O functions.

Entry-Point IECVSMMG: Processing to acquire an RQE block.

Callers: IECVEXCP.

Exit-Normal: Return to caller.

Entry-Point IECVSMMF: Processing to free an RQE block.

Callers: IECVEXCP.

Exit-Normal: Return to caller.

Entry-Point IECVSMEG: Processing to acquire large blocks.

Callers: IECVEXCP.

Exit-Normal: Return to caller.

Entry-Point SMEFRCHN: Processing to free large blocks.

Callers: IECVEXCP.

Exit-Normal: Return to caller.

Entry-Point IECVSMPF: Normal processing to purge/free large and RQE blocks in the caller's address space.

Callers: IOS PURGA.

Exit-Normal: Return to caller.

Entry-Point SMGFREVR: Verifies that the storage manager's pool pages are chained correctly.

Callers: IEAVEQV2.

Exit-Normal: Return to caller.

Entry-Point IECVSMEL: Called by the ENF facility on event code 4 to determine if a compress of medium and large blocks would be beneficial.

Callers: ENF routine IEFENFNM.

Exit-Normal: Return to caller.

Entry-Point IECVSCOM: Compresses both medium and large blocks.

Callers: Dispatcher when SRB is scheduled.

Exit-Normal: Return to caller.

External-References: IEAVVEQ1.

IECVFDEV – IOS Force Device Interface Routine

Function: This routine is invoked in task mode to force a device offline.

Callers: IEE3103D, IEE4603D.

Exit-Normal: Returns to the caller.

Entry-Point FDEVSRB: Calls IOSRFDEV in SRB mode.

Callers: Dispatcher.

Exit-Normal: Returns to the dispatcher.

External-References: IOSRFDEV.

IECVGENA – Miscellaneous IOS Out of Line Macro Services

Function: This module performs the out of line processing for the following macros: IOSGEN, IOSINFO, IOSINTRP, IOSLOOK. IECVGENA contains a branch vector table to the various routines contained in this module. The vector table callers are as follows:

```
IECVGENA+0  = Reserved.
IECVGENA+4  = Reserved.
IECVGENA+8  = IOSINTRP.
IECVGENA+12 = IOSLOOK.
IECVGENA+16 = IOSGEN UCBFLG.
IECVGENA+20 = IOSGEN WTOID.
IECVGENA+24 = IOSGEN SETVOIDC.
IECVGENA+28 = IOSINFO.
```

Entry-Point IECVGENA + 8: This entry-point is used by IOSINTRP to simulate an unsolicited device end interrupt if the device is not ready (UCBNRY) or no paths are available (UCBNOPTH). This routine also redrives the subchannel.

Callers: IOSINTRP macro.

Exit-Normal: To the I/O FLIH via a LPSW. The I/O FLIH will return to IOSINTRP macro processing.

Entry-Point IECVGENA + 12: This entry-point is used by the IOSLOOK macro to change to 31-bit addressing mode and branch to IOSVLOOK.

Callers: IOSLOOK macro.

Exit-Normal: Branches to IOSVLOOK. IOSVLOOK will return to the IOSLOOK macro processing.

Entry-Point IECVGENA + 16: This entry-point is used by the IOSGEN UCBFLG macro to update the UCB.

Callers: IOSGEN UCBFLG macro.

Exit-Normal: Returns to the caller.

Entry-Point IECVGENA + 20: This entry point is used by the IOSGEN WTOID macro. It updates the UCBWTOID field with the requested message id and deletes any messages whose id is currently in the UCBWTOID field of the UCB.

Callers: IOSGEN WTOID macro.

Exit-Normal: Returns to the caller.

Entry-Point IECVGENA + 24: This entry point is used by the IOSGEN SETVOIDC macro. It updates the vector of I/O drivers (VOID).

Callers: IOSGEN SETVOIDC macro.

Exit-Normal: Returns to the caller.

Entry-Point IECVGENA + 28: This entry point is used by the IOSINFO macro. It obtains a subchannel number for a requested UCB.

Callers: IOSINFO macro.

Exit-Normal: Returns to the caller.

External-References:

IOSVSMGR - Invoked from entry-point IECVGENA + 16 to free large blocks and IOQs.

IOSVIRBA - Invoked from entry-point IECVGENA + 8 to process simulated interrupts.

IOSVLOOK - Invoked from entry-point IECVGENA + 12 to find the UCB address.

IECVIOPM – I/O and Path Mask Update Routine

Function: IECVIOPM is called to verify the logical and physical paths to a device and to record each path's status by updating the UCB logical path mask.

Callers: IEEVCHPR, IEEVDEV, and IEEVPTHR.

Exit-Normal: Returns to the caller.

External-References: IEA0PT01, IECVMAP, IECVQCNT, IOSVDPTH, IOSVMSCQ, DIEXIT.

IECVIOSI – IOS Dynamic Pathing Initialization Module

Function: This module initializes dynamic pathing for devices that support this feature. This module also establishes the IOS storage manager as an ENF listener.

Callers: IEEMB860.

Exit-Normal: Returns to the caller.

Entry-Point IOSIDPTH: Retries from the ESTAE routine IOSIRECV if IECVIOSI has not yet called IECVDPTH.

Callers: RTM.

Entry-Point IOSISMGR: Retries from the ESTAE routine IOSIRECV if IECVIOSI has not yet called IOSVSMEI.

Callers: RTM.

Entry-Point IOSIEXIT: Retries from the ESTAE routine if IECVIOSI has already called IECVDPTH and IOSVSMEI.

Callers: RTM.

Entry-Point IOSIRECV (ESTAE routine): Recovers from an unexpected error that caused exit to RTM. Attempts to retry the first two times it is entered. Retries the call to IECVDPTH if the error occurred before IECVIOSI called IECVDPTH. Otherwise, it retries the call to IOSVSMEI if the error occurred before IECVIOSI called IOSVSMEI. If both calls have been made, it retries at the exit code, label IOSIEXIT. If entered a third time, percolates the error.

Callers: RTM (for ESTAE requests from this module).

Exit-Normal: To RTM.

Exit-Error: To RTM.

External-References: IECVDPTH, IOSVSMEI.

IECVOID – IOS VOID Table

Function: This module contains the VOID tables for I/O requestors that require special handling routines associated with their I/O requests. These dependent routines include purge (both halt and quiesce), restore routine, extent check routine, and notification routine.

Callers: None.

Exit-Normal: Not applicable.

Exit-Error: Not applicable.

External-References: Various driver routines.

IECVMAP – Device Path Map Build Routine

Function: This module determines the availability of all installed paths to a device. It builds a table containing the physical and logical availability of all paths and their on-line and off-line status.

Callers: All users of IOS that desire this function.

Exit-Normal: Returns to the caller via BASR.

External-References: None.

IECVOPTB – Channel Command Word (CCW) Operation Table for 3704/3705 Communication Device

Function: This module provides the CCW translation operation table that indicates how a device command code is to be translated for 3704/3705 communication device.

IECVOPTC – Channel Command Word (CCW) Operation Table for Teleprocessing Class Device

Function: This module provides the CCW translation operation table that indicates how a device command code is to be translated for teleprocessing class device.

IECVOPTD – Channel Command Word (CCW) Operation Table for DASD Class Devices

Function: This module provides the CCW translation operation table that indicates how a device command code is to be translated for DASD class devices.

IECVOPTE – Channel Command Word (CCW) Operation Table for 3211 printer device

Function: This module provides the CCW translation operation table that indicates how a device command code is to be translated for 3211 printer device.

IECVOPTG – Channel Command Word (CCW) Operation Table for Graphic Class Devices

Function: This module provides the CCW translation operation table that indicates how a device command code is to be translated for graphic class devices.

IECVOPTH – Channel Command Word (CCW) Operation Table for 3890 MICR Device

Function: This module provides the CCW translation operation table that indicates how a device command code is to be translated for 3890 MICR device.

IECVOPTI – Channel Command Word (CCW) Operation Table for 3886 OCR Device

Function: This module provides the CCW translation operation table that indicates how a device command code is to be translated for 3886 OCR device.

IECVOPTJ – Channel Command Word (CCW) Operation Table for 3895 Printer

Function: This module provides the CCW translation operation table that indicates how a device command code is to be translated for 3895 printer.

IECVOPTK – Channel Command Word (CCW) Operation Table for 1287/1288 Optical Reader

Function: This module provides the CCW translation operation table that indicates how a device command code is to be translated for 1287/1288 optical reader.

IECVOPTL – Channel Command Word (CCW) Operation Table for 3851 MSS Controller

Function: This module provides the CCW translation operation table that indicates how a device command code is to be translated for 3851 MSS controller.

IECVOPTM – Channel Command Word (CCW) Operation Table for 3540 Diskette

Function: This module provides the CCW translation operation table that indicates how a device command code is to be translated for 3540 diskette.

IECVOPTN – Channel Command Word (CCW) Operation Table for 3838 VPSS

Function: This module provides the CCW translation operation table that indicates how a device command code is to be translated for 3838 VPSS.

IECVOPTT – Channel Command Word (CCW) Operation Table for Tape Class Devices

Function: This module provides the CCW translation operation table that indicates how a device command code is to be translated for tape class devices.

IECVOPTU – Channel Command Word (CCW) Operation Table for Unit Record Class Devices

Function: This module provides the CCW translation operation table that indicates how a device command code is to be translated for unit record class devices.

IECVPST – IOS Post-Status

Function: The post status routine does enabled processing for termination of I/O requests. It interfaces with user-specified exits (normal, abnormal and intermediate status) and attention routines to handle unsolicited device end interruptions. Post status also interfaces with device-dependent error recovery procedures (ERPs) and handles their returns via SVC 15.

Callers: IOS modules, unconditional reserve processor.

Exit-Normal: BSM to driver terminating routine (for driver-generate IOSBs); BSM to dispatcher for IOS-generated IOSBs.

Exit-Error: Same as normal exit. See also entry-point PSTFRRTN.

Entry-Point IGC015: Retries channel programs for ERPs, or calls the normal or abnormal end exit when the ERP retries are completed.

Callers: ERPs, DDR, OBR, IGE0025C.

Exit-Normal: BSM to dispatcher.

Exit-Error: BSM to dispatcher.

Entry-Point PSTFRRTN: Recovers from an unexpected error that caused exit to RTM.

Callers: RTM.

Exit-Normal: Returns to RTM specifying retry and the address of a routine in IECVPST.

Entry-Point PSTWFRR: Recovers from an unexpected error in the restartable wait routine that caused exit to RTM.

Callers: RTM.

Exit-Normal: Returns to RTM specifying retry and the address of a routine in IECVPST.

External-References: Normal end exit, Abnormal end exit, Intermediate status exit, Attention exit, Resident ERPs, Non-resident ERPs, IOSVURDT, IEEVLDWT, IOSVSMGR, IEAPTRV.

IECVPURG – Resident Purge Routine

Function: The resident purge routine performs three functions:

1. Decreases the count of partially-processed I/O requests.
2. Frees the error recovery procedures (ERP) work area addressed in the SIRB.
3. If the nonresident purge routine (IGC0001F) issued a PURGEDQ macro, puts the SRB/IOSB on the queue of SRB/IOSBs chained from the IOS purge interface block (IPIB).

Entry-Point IECVQCNT: Decreases the count of partially-processed I/O requests.

Callers: Drivers' termination procedures or the nonresident purge routine (IGC0001F).

Exit-Normal: Returns to the caller via branch on register 14.

Entry-Point IECVQCNT+4: Frees the ERP work area addressed in the SIRB.

Callers: Dispatcher.

Exit-Normal: Returns to the caller via SVC 3.

Entry-Point IECVQCNT+A: Puts a SRB/IOSB on the queue of SRB/IOSBs chained from the IOS purge interface block (IPIB).

Callers: System's SPL purge module.

Exit-Normal: Returns to the caller via branch on register 14.

Exit-Error: Issues X'C0D' abend if PURGEDQ was not issued by IGC0001F.

External-References: IEA0PT01, IOSVSMGR.

IECVURVL – Unconditional Reserve Validation Routine

Function: This module is called by IOS modules to determine if unconditional reserve processing is to be performed. This module is called by the following IOS modules:

- IOSRSLH - Channel errors.
- IOSVIRBD - End-of-sense (EOS) conditions.
- IOSVIRBN - Deferred condition code 3 conditions.
- IOSRMIHR - From the MIH clear DIE routine to handle simulated interrupts and start pending conditions.

Callers: IOSRSLH, IOSVIRBN, IOSRMHIR, and IOSVIRBD.

Exit-Normal: Returns to the caller.

Entry-Point URVLFRR: To recover from an unexpected error that caused an exit to RTM. On first entry, attempts retry. If entered a second time, percolates the error.

Callers: RTM.

Exit-Normal: To RTM.

Exit-Error: To RTM.

External-References: IOSFQVIR, IEAVDVAL, IOSVSMGR.

IECVSTIO – STARTIO Compatibility Module

Function: Provides a compatibility interface to the start subchannel queue routine for users who code the SRB parameter on the STARTIO macro.

Callers: STARTIO macro.

Exit-Normal: BSM 0,15 to IOSVSSCQ.

Note: IOSVSSCQ will return to the caller of this module.

External-References: IOSVSSCQ.

IECVTCCW – CCW Translator Module

Function: This module performs four options involved with the translation of a caller's virtual channel program to a real channel program. These options include:

- Translating the caller's virtual channel program to a real channel program and fixing the program's data areas
- Unfixing the program's data areas
- Translating a virtual real channel address to its corresponding virtual channel program address
- Translating a virtual channel program address to its real channel program address

Another option allows IECVTCCW to request from the caller an additional large block of storage when more storage is required to translate the caller's virtual channel program to a real channel program.

Entry from: IECVEXCP and others who require virtual channel program translation.

Exit to: Caller.

IECVXDAS – DASD SIO Exit

Function: This module performs the following functions:

1. It initiates the volume verification process if necessary.
2. If volume verification is not necessary, it builds a channel program prefix if requested.
3. It invokes IOSVSSCH to start the channel program.

Callers: IOSVSSCH.

Exit-Normal: Returns to the caller.

External-References: IOSVSSCH.

IECVXDAT – DASD Trap Exit

Function: This routine processes completion of a reserve or release command for DASD devices.

Callers: IOSVIRBA.

Exit-Normal: Returns to the caller.

External-References: None.

IECVXDAU – DASD Unsolicited Interruption Exit

Function: This routine notifies VSAM if an unsolicited device end interruption (a possible volume change) occurs.

Callers: IOSVIRBD.

Exit-Normal: Returns to the caller.

External-References: None.

IECVXDRS – 2305 SIO Exit

Function: This module performs the following functions:

1. It locates a usable device address. There are eight possible device addresses for a 2305. The lowest device address is used only for reserve or release operations unless it is specifically requested by the driver. The other seven device addresses are used only for data transfer.
2. It builds a stand-alone reserve or release CCW, if needed.
3. It builds the channel program prefix, if needed.
4. It invokes a routine to start the channel program.

Callers: IOSVSSCH.

Exit-Normal: Returns to the caller.

External-References: IOSVSSCH, IECVSMGR.

IECVXDRT – 2305 Trap Exit

Function: This routine processes the completion of a release or reserve command for 2305 devices.

Callers: IOSVIRBA.

Exit-Normal: Returns to the caller.

External-References: None.

IECVXGRT – Graphics Trap Exit

Function: This routine receives control when channel end occurs for a graphics device. It signals IOS not to schedule the post status routine. This prevents posting the driver with normal completion because a unit check might be presented later when device end occurs.

Callers: IOSVIRBA.

Exit-Normal: Returns to the caller.

External-References: None.

IECVXMGN – 3851/3838 Sense Exit

Function: This routine sets a quiesce level, a 1-byte numerical indicator that reflects the level of error and whether or not I/O requests can still be processed. The quiesce level ensures that no I/O is done to the device between the sense operation and the ERP retry.

Callers: IOSVSSCH.

Exit-Normal: Returns to the caller.

External-References: None.

IECVXPRE – 3211/3800 End-Of-Sense Exit

Function: This routine indicates to IOS whether or not IOS should intercept the next I/O request. If a request is intercepted, IOS does not process it but instead invokes the appropriate ERP.

Callers: IOSVIRBD.

Exit-Normal: Returns to the caller.

External-References: None.

IECVXTAT – Tape Trap Exit

Function: This routine updates the I/O request count in the UCB tape extension.

Callers: IOSVIRBA.

Exit-Normal: Returns to the caller.

External-References: None.

IECVXTPT – TP Trap Exit

Function: This routine receives control when a solicited interruption with no unit check occurs for a TP device. This routine simply returns to IOSVIRBA with a return code of zero because no trap exit processing is required for the interruption.

Callers: IOSVIRBA.

Exit-Normal: Returns to the caller.

External-References: None.

IECVXT3S – 3400 Tape SIO Exit

Function: This routine does the following functions:

1. It initiates the volume verification process if necessary by returning to the caller with a return code of eight.
2. It builds a channel program prefix if necessary.
3. It invokes a routine to start the channel program.

Callers: IOSVSSCH.

Exit-Normal: Returns to the caller.

External-References: IOSVSSCH.

IECVXURS – Unit Record SIO Exit

Function: This routine builds an immediate command if requested to do so by the ERP. Whether or not it builds an immediate command, it invokes the start channel program service routine to start the channel program.

Callers: IOSVSSCH.

Exit-Normal: Returns to the caller.

External-References: IOSVSSCH.

IECVXURT – Unit Record Trap Exit

Function: This routine receives control when a solicited interruption with no unit check occurs for a unit record device. This routine simply returns to IOSVIRBA with a return code of zero because no trap exit processing is required for the interruption.

Callers: IOSVIRBA.

Exit-Normal: Returns to the caller.

External-References: None.

IECVXVRS – 3330V SIO Exit

Function: This module performs the following functions:

1. Determines if a request that has been held should be released or deferred.
2. Builds a channel program prefix if necessary.
3. Starts the channel program.

Callers: IOSVSSCH.

Exit-Normal: Returns to the caller.

External-References: IOSVSSCH, IOSVIOQC.

IECVXVRT – 3330V Trap Exit

Function: This routine detects and processes cylinder faults on 3330V devices. It also processes the completion of a reserve or release command.

Callers: IOSVIRBA.

Exit-Normal: Returns to the caller.

External-References: None.

IECVXVRU – 3330V Unsolicited Interruption Exit

Function: This routine notifies VSAM if an unsolicited device end interruption (a possible volume change) has occurred. It also processes attention interrupts.

Callers: IOSVIRBD.

Exit-Normal: Returns to the caller.

External-References: None.

IEDAY3 – LOGON Synchronization Module

Function: This routine synchronizes the LOGON process for TIOC by preventing memory creation from proceeding until the TIOC LOGON routine has fully initialized the ASCB.

Entry from: IEEVWAIT.

Exit to: IEEVEMCR.

IEEAB400 – WTP Buffer Routine

Function: This module puts WTP messages into buffers and, if the messages are also WTO messages, issues the WTO macro. When the buffer becomes full, this module calls IEEAB401 to issue the WTPs.

Entry from: IEFAB4FD, IEFAB4DD, IEFAB4E4.

Exit to: Caller.

Called routines: IEEAB401.

IEEAB401 – WTP PUT Routine

Function: This module issues the PUT macro to write any WTP messages present in the buffer built by IEEAB400.

Entry from: IEEAB400, IEFAB4A0, IEFAB4DD, IEFAB421, IEFAB4E4, IEFAB4E8, IEFBB401, IEFBB410.

Exit to: Caller.

Called routines: None.

IEECB800 – DISPLAY/TRACK Command Common Processor

Function: This module builds the display for a DISPLAY A or TRACK A command. The display consists of system status information about active tasks, jobs, and time sharing terminals. TRACK command requests appear on a graphics (screen) device. DISPLAY command requests appear on either a graphics or a paper device.

Entry from: IEEVWAIT via ATTACH.

Exit to: End of task.

Called routines: IEECB801.

IEECB801 – Issuer of WTO and TPUT for DISPLAY and TRACK Commands

Function: This module issues the multi-line WTO (MLWTO) or iterative TPUTs for the DISPLAY and TRACK commands or issues a single-line WTO or TPUT.

Entry from: IEECB800, IEECB804, IEECB808, IEECB815.

Exit to: Caller.

IEECB804 – DISPLAY R Command Processor

Function: This module displays on an operator's console, the text for the following types of messages:

- Messages awaiting replies (WTORs).
- Immediate action messages.
- Eventual action messages.

Addresses of devices that need to be made ready or need operator intervention are also displayed.

Entry from: Attached by IEEVWAIT.

Exit to: Caller.

IEECB805 – SET MPF Processor

Function: This module builds a sorted table of message IDs and message prefixes for the specification of suppression, action message relation, and associated WTO us er exits. It initializes the table with entries from the MPFLSTxx SYS1.PARMLIB member, where xx is the operand specified on the SET MPF command.

Entry from: IEEMB811.

Exit to: IEEMB811.

Called routines: IEECB812, IEEMB878, IEEMB882, IEE0503D, IEFJSIMW.

IEECB806 – TRACE Command Router and Operands (MT and STATUS) Processor and Display TRACE Command Processor

Function: This module syntax checks and processes the following TRACE command operands:

- MT
- MT,
- MT,OFF

Link to IEECB925 to process the following TRACE command operands:

- ST
- ST,

Link to IEECB954 to process the following TRACE command operands:

- CT
- CT,

To process the TRACE command operand status. Link to IEECB924 to obtain system TRACE status prior to issuing the TRACE status message IEE839I. To process the display TRACE command:

- Link to IEECB924 to obtain system TRACE status. Build display TRACE message IEE843I through display message service in IEECB808. Link to IEECB955 to append component TRACE status information to message IEE843I. Return to IEECB806 to issue message IEE843I.

Entry from: IEEVWAIT via ATTACH.

Exit to: End of task.

IEECB807 – DISPLAY MPF Processor

Function: This module displays the entries in the current MPF and color specification tables built by IEECB805.

Entry from: IEECB808.

Exit to: IEECB808.

Called routines: IEECB808, IEE0503D.

IEECB808 – Command Service Routine

Function: This module routes control to command processors and provides standard command facilities.

Entry point IEECB808: See FUNCTION.

Exit Normal: Return to caller.

External-References: None.

IEECB809 – VARY Device Service Routine

Function: This module handles all ASSIGN/UNASSIGN and dynamic pathing processing for the MVS VARY Device command.

Entry from: IEECB862, IEE3130D.

Exit to: IEECB862, IEE3130D.

Called routines: IECVDPTH, IEEMB814, IEFAUSRV.

IEECB811 – Reply Syntax Checker

Function: This module validates the command syntax and ID and schedules an SRB for stage 2 processing.

Entry from: IEECB808.

Exit to: IEECB808.

IEECB812 – CLRA Table Entry Processor

Function: This module processes color entries and changes the CLRA (color/highlighting attribute).

Entry from: IEECB805.

Exit to: Caller.

Called routines: IEEMB882.

| ### IEECB814 – Display Consoles Message Builder

| **Function:** This routine builds the display consoles
| response, message IEE249I, information.

| Entry-Point IEECB814: See FUNCTION.

Callers: None.

Exit-Normal: Return to IEEXEDNA via BR 14.

External-References: None.

| ### IEECB815 – Routing Code Keyword Value Syntax
| Processor for Commands

| **Function:** This routine checks the syntax of the
| routing code keyword value and returns the parsed
| keyword value to the caller.

| Entry-Point IEECB815: Processes the routing
| code keyword value.

| Callers: IEE4403D, IEE4703D and IEECB804.

| Exit-Normal: Return to caller.

Exit-Error: Return to caller.

External-References: IEE0503D and IEECB801.

IEECB816 — Set PFK Processor

Function: This module builds a sorted table containing PFK tables and the PFK definitions within each PFK table. This table is initialized with entries from the PFKTABxx SYS1.PARMLIB member, where xx is the operand specified on the SET PFK command.

Entry-Point IEEMB816

Callers: IEEMB811 via link.

Exit-Normal: Return to caller via BR 14

Exit-Error: Return to caller via BR 14 with a return code of 20 in reg 15

External-References:

Routines:

- IEEMB887 - Generalized Parser
- IEEMB878 - Parmlib Read Service Routine
- IEAVH600 - Comm Task Cross Memory Service Routine

IEECB817 — Parse Definitions to syntax check PFKTABxx member of SYS1.PARMLIB

Function: These parse definitions are accessed by the parser module (IEEMB887) to validity check the syntax of the PFKTAB statement in the PFKTABxx member of SYS1.PARMLIB. (This is a non-executable module).

Entry from: N/A

Exit to: N/A

Called routines: None

IEECB818 — Message Loss Detection

Function: This module detects situations where broadcast messages are not being received by any active console, and not all routing codes are assigned to at least one console receiving informational messages.

Entry from: IEE4603D, IEE4903D, IEE7703D.

Exit to: Caller.

Called routines: SVC 34.

IEECB819 — General WTO User Exit Delete Routine

Function: This module deletes the general WTO user exit (IEAVMXIT) and frees the storage used for the GENX table.

Entry from: IEE7903D.

Exit to: Caller.

IEECB860 — Command ESTAE Creation/Exit Routine

Function: This module creates an ESTAE environment for the module that invokes it. RTM invokes the module, it provides a dump, messages, and other appropriate processing.

Entry from: Caller.

Exit to: RTM or caller.

IEECB862 — VARY Device ESTAE Exit Routine

Function: This module is the ESTAE exit routine for VARY Online/Offline/Console processing. It updates the SDWA and takes an SVC dump. Then, if any device was assigned or had dynamic pathing done, but is not disbanded. IEECB862 then returns two RTM to percolate to IEECB860.

Entry from: RTM.

Exit to: RTM.

Called routines: IEECB809.

IEECB866 — Console Dump

Function: This module causes a dump of virtual storage to a SYS1.DUMP data set. - An SVC dump is taken with the options specified on the response to the DUMP command.

Entry from: IEEVWAIT.

Exit to: System via SVC 3.

IEECB867 — SVD Dump Reason Code Text Processor

Function: To use the reason code passed by the caller to locate the corresponding test and copy that text into the callers buffer. The set of reason codes handled by this module are the reason codes returned by SVC Dump when a requested dump is not taken (SVC Dump return code of 8).

Entry-Point IEECB867 To use the reason code passed by the caller to locate the corresponding test and copy that text into the callers buffer. The set of reason codes handled by this module are the reason codes returned by SVC Dump when a requested dump is not taken (SVC Dump return code of 8).

Callers: None

Exit-Normal: Returns to caller.

External-References:

- IEECB868
- RCTEXT - table in IEECB868 containing the reason code text.

IEECB900 – VARY CN Syntax Processor

Function: This module syntax checks the command and passes control to the authority processor for valid commands.

Entry from: IEEVWAIT.

Exit to: End of task.

Called routines: IEECB901.

IEECB901 – VARY CN Console Processor

Function: This module modifies a console authority value for eligible consoles.

Entry from: IEECB900.

Exit to: Caller.

IEECB904 – VARY Range Processor

Function: This module creates a VARY command from a VARY device address-range command.

Entry from: IEE3603D, IEE3103D, IEE3303D, IEE4203D, IEE4603D.

Exit to: Caller.

IEECB905 – SLIP Command Processor

Function: This module sets, modifies, and deletes SLIP traps. It also schedules IEAVTGLB to activate or deactivate PER monitoring.

Entry from: IEEVWAIT via ATTACH.

Exit to: SVC 3 via a BR 14 instruction.

IEECB906 – SLIP ESTAE Routine

Function: This module provides ESTAE recovery for the SLIP command (IEECB905 and IEECB909), the DISPLAY SLIP command (IEECB907) and the SLIP message module (IEECB908). It cleans up resources and takes an SVC dump. It does not retry.

Entry from: ABEND STAE interface routine.

Exit to: ABEND STAE interface routine.

IEECB907 – Display SLIP Processor

Function: This module displays SLIP traps either in summary or in detail in response to the DISPLAY SLIP operator command. If the SLIP command processor requests it, IEECB907 also provides detailed trap information.

Entry from:

- IEEVWAIT via ATTACH for DISPLAY SLIP.
- IEECB905 via BALR to show options.

Exit to: End of task if entered from IEEVWAIT, IEECB905 if entered from IEECB905.

Called routines:

- IEECB860 via BALR and ESTAE exit.
- IEECB906 via ESTAE exit.
- IEECB908 via BALR.
- IEE0503D via BALR.

IEECB908 – SLIP Message Module

Function: This module contains SLIP messages. It formats and writes out the message requested and if the message demands a response, waits for a reply. It also posts IEECB914 to converse with a TSO SLIP user.

Entry from: IEECB905 and IEECB907.

Exit to: Caller.

IEECB909 – SLIP Command Interpreter

Function: This module scans and interprets all SLIP command input text.

Entry from: IEECB905 via BALR.

Exit to: Caller.

IEECB910 — DISPLAY DUMP Processor

Function: This module produces the IEA852I, IEA853I, IEA854I, and IEA857I message output in response to the operator command DISPLAY DUMP.

Entry from: IEECB808 via LINK.

Exit to: Caller.

Called routines:

IEECB912 to process the OPTIONS request.

IEECB808 message service routine

IEE0503D for syntax error messages.

IEECB911 — DISPLAY DUMP Message CSECT

Function: This CSECT contains the message text that IEECB910 and IEECB912 use to construct the lines for output messages.

Used by IEECB910 and IEECB912.

IEECB912 — DISPLAY DUMP OPTION Keyword Processor

Function: This module builds the operator message IEEB857I in response to the operator message: DISPLAY DUMP,OPTIONS.

Entry from: IEECB910.

Exit to: Caller.

Called routines: IEECB808.

IEECB913 — SMF SETSMF Command Router

Function: This module starts processing for the SETSMF command. It waits for any SET or DISPLAY command in process to complete, then it posts IFASMF to invoke IFASSMF to execute the SETSMF command.

Entry from: IEEVWAIT.

Exit to: Caller.

Called routines: IEEMB824, IEEMB860, IFASMF via POST.

IEECB914 — SLIP TSO Communication Routine

Function: This module issues a TPUT and TGET to the SLIP TSO user.

Entry from: IKJEFF00 via ATTACH.

Exit to: SVC 3 via BR14.

IEECB915 — SLIP Processor POST Routine

Function: When it is scheduled by the SLIP processor (IEAVTSLP), this module posts the SLIP command processor (IEECB905). The posting is done when the SLIP processor requests message processing.

Entry from: SRB dispatcher.

Exit to: Caller.

IEECB916 — SMF DISPLAY SMF Command Router

Function: This module starts processing for the DISPLAY SMF command. It checks the validity of the command, then waits for any SET or DISPLAY commands in process to complete. Posts IFAFMS to invoke IFADSMF to execute the DISPLAY command.

Entry from: IEEVWAIT

Exit to: Caller.

Called routines: IEEMB824, IFASMF via POST.

IEECB923 — DUMPDS Command Processor

Function: This module is given control to process a DUMPDS operator command. The command is parsed for errors and a control block (DSPA) is built to represent the DUMPDS request. IEECB926 is POSTed to process this DSPA and IEECB923 issues a WAIT. When IEECB926 is done, it POSTs IEECB923 who then issues message IEE855I which displays the results of the DUMPDS request to the operator.

Note: The DUMPDS command is used to ADD, DELETE, and CLEAR SYS1.DUMP data sets.

Entry from: IEECB808 via LINK.

Exit to: Caller.

Called routines: IEECB808, IEE0503D, IEECB926.

IEECB924 — System Trace Status Processing Routine

Function: IEECB924 updates the system trace parameter list (TRST) with the current trace status and the tracing options in effect. The caller uses the updated TRST to display the system trace status and options at the operator console.

Callers: IEECB806 (TRACE Command Processor).

Exit-Normal: Returns to the caller.

Exit-Error: ABEND X'09E'.

External-References: None.

IEECB925 – System Trace Operator Command Processing Routine

Function: IEECB925 controls the parsing and validation of the TRACE ST operator command to alter the system trace status and options, and issues the ALTRTRC macro to change the system trace status and options.

Callers: IEECB806 (TRACE command processor).

Exit-Normal: Returns to the caller.

Exit-Error: ABEND X'09E'.

External-References: IEE0503D.

IEECB926 – SYS1.DUMP and DUMPDS Processor

Function: This module provides the following two functions:

1. When the DUMPSRV address space is created, IEAVTDSV attaches IEECB926 as a subtask. This module then does a dynamic allocation on all the DASD SYS1.DUMP data sets to prevent anyone from scratching the data set while SDUMP is using it.
2. This module then WAITs for IEECB923 to post it for DUMPDS processing. IEECB926 implements the requests made on the DUMPDS command to ADD, DELETE, and CLEAR SYS1.DUMP data sets. When all the requested data sets have been processed, IEECB923 is POSTed and this module goes back to WAIT.

Entry from:

IEAVTDSV via ATTACH during DUMPSRV address space creation.

IEAVTDSE via ATTACH after a IEECB926 task termination.

IEECB923 via POST for DUMPDS processing.

Exit to: None. This is a never ending task. When it's work is complete it issues a WAIT macro to wait for further DUMPDS requests.

IEECB927 – CONFIG Command Processor

Function: This module establishes resources, determines the type of request, checks authorization, and passes control to process the CONFIG request.

Entry From IEECB808.

Exit To Caller.

Called routines: IEEVSCAN, IEEVRSCN, IEEVRDPM, IEEVCONF, IEEVMESS.

IEECB929 – SET SLIP = xx Processor

Function: This routine calls IEECB905 to process commands from PARMLIB member IEASLPxx.

Entry Point IEECB929: See Function.

Exit To: Caller.

Called routines: IEECB905.

IEECB930 – SHDR SCE/ID Queue Verification

Function: This routine repairs SCE PER and non-PER queues as well as ID queue if any errors are found.

Entry Point IEECB930: See Function.

Exit To: Caller.

Called routines: None.

IEECVCTE – Dummy WTO Exit Routine

Function: This routine simply returns to the caller.

Entry from: Any caller.

Exit to: Address in register 14.

Called routines: None.

IEECVETA – CONTROL Command Syntax Checker (DIDOCS)

Function: Checks the syntax of the CONTROL command.

Entry from: IEECVET1, IEECVET4.

Exit to:

- IEECVETK if roll mode or RTME (time interval specification) changes.

- IEECVETD to issue warning message.
- IEECVETU (3270 I/O module) to rewrite the entry area.
- IEECVETD to issue an error message.

IEECVETC – Asynchronous Error Processing Module (DIDOCS)

Function: This module initializes the DCM for a reopen. It processes asynchronous errors and prepares for a console switch after a permanent synchronous error.

Entry from:

- IEECVET1 for asynchronous error processing.
- IEECVFTG for asynchronous error processing.
- IEECVET4 for asynchronous error processing.

Exit to:

- IEECVETE to issue error message.
- IEECVFTG to assist in error processing.
- IEECVETK to set timer interval.
- IEECVETU (3270 I/O module) to write the screen image.
- IEAVSWCH to perform console switch.
- IEECVET4 to simulate pressing the CLEAR key if the device supports the erase-write-alternate (EWA) command.

IEECVETD – Message Module (DIDOCS)

Function: For full capability consoles, places error or informational messages in the warning line or the instruction line; for message stream or status display consoles, places messages in the warning line.

Entry from: Any DIDOCS module detecting a message-output condition.

Exit to:

- IEECVETU (3270 I/O module) to write the screen.
- IEECVET1 if hold mode is in effect.
- IEECVET3 if roll mode is in effect and an unviewable message is needed.

IEECVETE – Message Module (DIDOCS)

Function: Places messages into the instruction or warning line and places asynchronous error messages into the entry area.

Entry from:

- IEECVETC to issue asynchronous error message.
- IEECVET4 to issue error message.
- IEECVETK to issue error message.
- IEECVETF to issue error message.
- IEECVET6 to issue error message.
- IEECVET8 to issue error message.

Exit to: IEECVETU (3270 I/O module) to write messages to the screen.

IEECVETF – Light-Pen and Cursor Detect Analyzer (DIDOCS)

Function: Determines from the location of a light-pen or cursor detect what function is being requested.

Entry from: IEECVET1.

Exit to:

- IEECVET4 if detect occurred in out-of-line display.
- IEECVETU (3270 I/O module) to process ENTER.
- IEECVFTA to process PFK line detect.
- IEECVET8 to process delete request.
- IEECVET9 to process verification of a delete operation.
- IEECVETD to issue error message.
- IEECVETE to issue error message.
- IEECVET1 to route ENTER for 3277 device.

IEECVETG – Open/Close Module (DIDOCS)

Function: Performs open and close processing for cathode ray tube devices used as consoles.

Entry from:

- IEECVET1 if open pending.
- IEECVETK after it removes the console from roll mode.

Exit to:

- IEAVSWCH if device was opened for console switch.
- IEECVET1 after open to perform other communications task operations.
- IEECVFTG to complete close of a console.
- IEECVETK to remove console from roll mode before close.

IEECVETI – Timer Exit Routine for Roll Mode

Function: This module gains control automatically whenever the STIMER elapses. It then posts an ECB in a console's UCME so that the timer-interpreter can analyze the situation.

Entry-Point IEECVETI: To post a console's ECB when the timer elapses.

Callers: None

Exit-Normal: To caller, via BR 14.

Exit-Error: None

External-References:

- Routines:
  - Post, branch entered

IEECVETJ – Roll Mode Processor (DIDOCS)

Function: Rolls messages off the console screen based on a timer interval and the number of lines to be rolled.

Entry from: IEECVET1.

Exit to:

- IEECVETK to display the number of messages waiting to be displayed.
- IEECVET2 to handle timer supervision if no roll occurred and to display messages if space exists in the message area.

IEECVETK – Timer Interpreter (DIDOCS)

Function: Resolves the different time intervals at which roll is to occur for different CRT consoles.

Entry from:

- IEECVET1 if timer expires.
- IEECVETC to reset time interval after an asynchronous error.
- IEECVETG to remove a console from roll mode before closing the console.
- IEECVETA if roll mode or RTME (time interval) changes occur.

Exit to:

- IEECVET2 to display message after roll.
- IEECVETE to display error message.
- IEECVETD to display warning message.
- IEECVETG to close the device.
- IEECVETU to write the screen image on the console screen.
- IEECVET1 if no roll is to occur.

IEECVETQ – 2250 Device I/O Module - Part 2 (DIDOCS)

Function: Performs I/O operations to 2250 devices used as operator consoles.

Entry from: IEECVETP.

Exit to: IEECVET1 after I/O is initiated.

IEECVETU – 3270 Device I/O Module - Part 1 (DIDOCS)

Function: Performs I/O operations to 3270 display devices used as operator consoles.

Entry from: Any DIDOCS module that modifies the SIB.

Exit to:

- IEECVET1 after I/O is initiated.
- IEECVETF if light-pen or cursor detect occurs outside entry area.
- IEECVFTG for a status switch.
- IEECVETK for roll mode.
- IEECVETV to complete I/O operations.
- IEECVETZ to complete I/O operations.

IEECVETV – 3270 Device I/O Module - Part 2 (DIDOCS)

Function: Performs I/O operations to 3270 display devices used as operator consoles.

Entry from: IEECVETU.

Exit to: IEECVET1 after I/O has been initiated.

IEECVETW – 3284/3286 Console Device Support Processor

Function: Performs the following functions for 3284/3286 printers used as consoles:

- Opens 3284 and 3286 as consoles.
- Initiates a write operation to write messages to the console.
- After a write operation completes successfully, updates the console queue.
- When the 3284 and 3286 printers are to be removed from console status, closes the consoles.

Entry from: IEAVVCTR.

Exit to:

- IEAVSWCH if an I/O error occurs or the console cannot be opened.
- Issuer of SVC 72, otherwise.

IEECVETZ – 3270 Extended Data Stream I/O Routine

Function: This routine is a device dependent routine that performs the requested I/O in the proper screen format. This routine is a copy of IEECVETV with modifications for the use of the extended data stream orders for extended color and highlighting.

Entry from: IEECVETU.

Exit to: Returns to IEECVET1.

Called routines: None.

IEECVET1 – DIDOCS Router Module

Function: Passes control to DIDOCS modules on the basis of requested function.

Entry from: IEAVVCTR, IEECVETQ, IEECVETV, IEECVFTR, and IEECVFTB.

Exit to:

- IEECVETG for console open and close.
- IEECVETC for asynchronous error processing.
- IEECVET4 for command processing.
- IEECVET2 for message output.
- IEECVET7 for message deletion.
- IEECVET9 for message deletion.
- IEECVFTP for erasing a status display or stopping a dynamic display.
- IEECVETJ for roll.
- IEECVETK for timer processing.
- IEECVFTM for out-of-line message output.
- IEECVETF for light-pen or cursor detect.
- IEECVETU (3270 I/O module) to perform I/O.
- IEECVFTA for PFK attention.
- IEECVFTL for in-line message output.
- IEECVFTT for display area blanking.
- IEECVETD to issue error messages.
- IEECVET6 for routed CONTROL E commands.
- IEECVET8 for routed CONTROL E/CONTROL E,SEG commands.
- IEECVETA for routed CONTROL S commands.
- IEECVFTB for routed CONTROL D,PFK/CONTROL E,PFK/CONTROL N commands.
- IEECVFTN for routed CONTROL D,F/CONTROL D,U/CONTROL D,H commands.
- IEAVVCTR to return to the caller.
- IEECVFTU for TDCM and PFK initialization.

IEECVET2 – In-line Message Output Module (DIDOCS)

Function: Controls the output of in-line messages.

Entry from:

- IEECVFT2 to determine exit from message output modules after single-line message output.
- IEECVET1 to display a message.
- IEECVFTL to determine exit from message output modules after multiple-line message output.
- IEECVETJ to display a message after roll.
- IEECVETK to display a message after roll.

- IEECVET9 to display a message after automatic deletion occurs.

Exit to:

- IEECVETU (3270 I/O module) to write messages to the screen.
- IEECVET1 if there were no messages to display.
- IEECVETD to issue a warning message.
- IEECVFT2 to move messages to the DCM.
- IEECVET7 to delete INTERVENTION REQUIRED messages.
- IEECVET9 for automatic deletion of messages.

IEECVET3 – Message Output Module (DIDOCS)

Function: Issues STIMER for roll mode after roll has been initialized.

Entry from: IEECVET2 to issue STIMER for roll mode.

Exit to:

- IEECVET1 if no I/O is to be performed after STIMER.
- IEECVETD if &q1.unviewable message&eq1. warning line is requested.
- IEECVETU (3270 I/O module) to write the screen image on the console screen.

IEECVET4 – Command Analyzer (DIDOCS)

Function: Analyzes command input and routes the command to other DIDOCS modules or to the system for processing.

Entry from:

- IEECVET1 to process command input.
- IEECVETF to process a cancel key or light-pen detect on a cancel function.
- IEECVFTA to issue the command that PFK processing placed in the entry area.
- IEECVETC to simulate pressing the CLEAR key.

Exit to:

- IEECVETC to complete asynchronous error processing.
- IEECVET8 to initialize message deletion.
- IEECVET9 to process delete verification.
- IEECVET1 after command processing is complete.
- IEECVETU (3270 I/O module) to rewrite the screen.
- IEECVETD for repeat last command processing.
- The appropriate CONTROL command processing modules (IEECVET6, IEECVET8, IEECVET9, IEECVETA, IEECVFTB, IEECVFTN, or IEECVFTP).

IEECVET6 – Message Deletion Module (DIDOCS)

Function: Deletes messages requested by the K E,F or K E,nn(,nn) command.

Entry from: IEECVET1, IEECVET4.

Exit to:

- IEECVET9 to remove messages.
- IEECVETD for message verification.
- IEECVETE to issue error messages.

IEECVET7 – Message Deletion Module (DIDOCS)

Function: Deletes INTERVENTION REQUIRED messages and messages requested by DOM.

Entry from: IEECVET1. IEECVET2. IEECVFT2.

Exit to: IEECVETU (3270 I/O module) to write the screen. IEECVET9 to attempt automatic deletion. IEECVET1 on NOP entry.

IEECVET8 – Message Deletion Module (DIDOCS)

Function: Deletes messages requested by the K E,SEG command, a cursor detect, or a light-pen detect.

Entry from: IEECVETF to process a light-pen or cursor detect. IEECVET4 to process the K E,SEG command. IEECVET1 for routed K E/K E,SEG commands.

Exit to: IEECVET9 to remove messages from the screen. IEECVETD to issue a verification message. IEECVETE to issue NO DELETABLE MESSAGE message.

IEECVET9 – Message Deletion Module (DIDOCS)

Function: Processes automatic deletion when the screen is full, numbers messages in response to the K D,N command, and deletes messages in response to a K command.

Entry from: IEECVET4 to process the K D,N command. IEECVET6 to delete messages if no verification is necessary. IEECVET7 to perform automatic deletion. IEECVET2 to perform automatic deletion if screen is full.

Exit to: IEECVET2 after automatic deletion. IEECVETD to issue error messages. IEECVETU (3270 I/O module) to rewrite the screen.

IEECVFTA – PFK-Entered Command Processor (DIDOCS)

Function: Processes automatic command entry that results from a depressed PFK or a light-pen detect on the PFK line.

Entry from: IEECVET4 for depressed PFK. IEECVETF for light-pen detect on the PFK line.

Exit to: IEECVET4 to process an entered command (nonconversational mode). IEECVETU (3270 I/O module) to write the the command to the entry area of the screen (conversational mode). IEECVFTD to issue error messages.

IEECVFTB – PFK Definition Processor (K D,PFK/K E,PFK/K N,PFK) (DIDOCS)

Function: Processes a PFK definition or redefinition in response to the K N,PFK command; displays or erases the PFK line in response to the K E,PFK or K D,PFK command.

Entry from: IEECVET1, IEECVET4.

Exit to: IEECVETU (3270 I/O module) to write or or erase the PFK line. IEECVFT1 when a PFK is defined. IEECVETD to write error messages. IEECVFTD to write error messages.

IEECVFTD – Message Module (DIDOCS)

Function: Prepares for the writing of informational or error messages.

Entry from: IEECVFTA and IEECVFTB.

Exit to: IEECVETU (3270 I/O module) to write messages to the screen.

IEECVFTG – DIDOCS Cleanup Module

Function: Finishes functions that involve a system interface.

- If device is being closed, flags messages no longer needed.
- If device is changing status to message stream or full capability, flags out-of-line messages on the console queue.
- If device is changing status to status display, flags messages on console queue.
- If device is recovering from asynchronous error, flags out-of-line messages on the console queue.
- If this routine flags any out-of-line messages except for an asynchronous error, it stops a dynamic display, frees SACBs obtained with a GETMAIN, and stores the area definitions in the UCME pageable extension for that console.

Entry from: IEECVETG for open or close of a device. IEECVETC for an asynchronous error. IEECVETU (3270 I/O module)

Exit to: IEECVET1 after close. IEECVETC after all processing other than close.

IEECVFTL – In-line Multiple-line Message Processor (DIDOCS)

Function: Controls the output of in-line, multiple-line messages (MLWTO).

Entry from: IEECVET1, IEECVFT2.

Exit to: IEECVFT2 to process next WQE. IEECVET2 if roll mode is in effect and the screen is full. IEECVETU (3270 I/O module) to write the screen. IEECVET1 if IEECVFTL encounters the end of the MLWTO chain but not the end of the message. IEECVETD to issue error messages.

IEECVFTM – Out-of-line Multiple-line Message Processor (DIDOCS)

Function: Controls the output of out-of-line, multiple-line messages (MLWTO).

Entry from: IEECVET1.

Exit to: IEECVFTO for display lines that overlay messages. IEECVFTQ for display lines that do not overlay messages. IEECVET1 if no more out-of-line messages are on the queue. IEECVFTP for blanking of messages below a display.

IEECVFTN – Status Display Processor (DIDOCS)

Function: Rewrites the display control line in response to a K D,F (framing), K D,H (holding), or K D,U (updating) command.

Entry from: IEECVET4 if console is a full-capability console. IEECVET1 if console is a status-display console.

Exit to: IEECVETU (3270 I/O module) to write the screen.

IEECVFTO – Status Display Processor (DIDOCS)

Function: Controls the output of a status display that overlays messages.

Entry from: IEECVFTM.

Exit to: IEECVETU (3270 I/O module) to write three lines of the display area.

IEECVFTP – Status Display Processor (DIDOCS)

Function: Controls the erasing of out-of-line displays.

Entry from: IEECVET1, IEECVET4.

Exit to: IEECVETU (3270 I/O module) to write the console screen. IEECVFTT to blank the screen below the erased display. IEECVET1 for a NOP entry.

IEECVFTQ – Status Display Processor (DIDOCS)

Function: Controls the output of a status display to an out-of-line area that does not overlay messages.

Entry from: IEECVFTM.

Exit to: IEECVETU (3270 I/O module) to write display lines to the screen. IEECVFTM if no I/O is done.

IEECVFTS – Color/Highlighting Attribute Table

Function: Non-executable table of color/highlighting attributes for consoles, used when no message processing facility (MPF) member of SYS1.PARMLIB is in effect.

IEECVFTT – Status Display Processor (DIDOCS)

Function: Overlays with blanks any in-line message that is below an out-of-line display or between two out-of-line displays.

Entry from: IEECVET1, IEECVFTM.

Exit to: IEECVETU (3270 I/O module) to write the screen.

IEECVFTU – Console Storage Initialization

Function: This module obtains and initializes the virtual storage containing the TDCM and the PFK definitions when a console is being opened.

Entry from: IEECVET1.

Exit to: IEECVET1.

IEECVFTV – TDCM Swap Routine

Function: This routine obtains the correct DIDOCS pageable DCM (TDCM) to use for the console and swaps it with the TDCM used during the open.

Entry from: IEECVET1.

Exit from Returns to the caller.

Called routines: IEECVFTU.

IEECVFT2 – Single-line Message Processor (DIDOCS)

Function: Displays and marks both action and deletable messages; locates and routes non-displayed multiple-line messages (MLWTOs).

Entry from: IEECVET2 to move messages to the DCM. IEECVFTL to process the next WQE.

Exit to: IEECVET2 to continue output queue scan. IEECVFTL to process in-line, multiple-line messages.

IEEC2740 – 2740 Console Device Support Processor

Function: Performs the following functions for a 2740 communications terminal used as a console.

- Opens the 2740 communications terminal as a console.
- Uses BTAM to read a command from the console.

- After a read operation completes successfully, passes the command to command processing (SVC 34).
- Uses BTAM to write messages to the console.
- When the 2740 communications terminal is to be removed from console status, closes the console.

Entry from: IEAVVCTR.

Exit to: IEAVSWCH if an I/O error occurs or the console cannot be opened. Otherwise IEAVVCTR.

IEEDISPD – Display Domain Processor

Function: Writes a console display of entries in the Domain Descriptor Table.

Entry from: IEEVWAIT.

Exit to: End of Task.

IEEDMCHP – Display M CHP Processing

Function: IEEDMCHP interprets the internal format of the DISPLAY M command and processes all channel path requests.

Entry-Point IEEDMCHP: See FUNCTION.

Callers: IEEMPDM.

Exit-Normal: Return to caller via register 14.

Exit-Error: If the ESTAE SVC returns a non-zero return code a X'0B7'abend is issued.

Entry Point CHPESTAE: ESTAE recovery routine.

Callers: RTM.

Exit Normal: Return to RTM via R14.

External-References:

- IEAVMSF, IEEDMMSG, IEECB808, and IOCSTSQE.

IEEDMCPU – Display M CPU Processing

Function: IEEDMCPU interprets the information table and displays the state of requested CPU's and vectors.

Entry-Point IEEDMCPU: See FUNCTION.

Callers: IEEMPDM.

Exit-Normal: Return to caller via register 14.

Exit-Error: If the ESTAE SVC returns a non-zero return code a X'0B7'abend is issued.

Entry Point CHPESTAE: ESTAE recovery routine.

Callers: RTM.

Exit Normal: Return to RTM via R14.

External-References:

- IEAVMSF, IEEDMMSG, and IEECB808.

IEEDMDEV – Display M = DEV Processing

Function: IEEDMDEV processes all D M = DEV requests. It interprets the information table and determines the type of display requested.

Entry-Point IEEDMCPU: See FUNCTION.

Callers: IEEMPDM.

Exit-Normal: Return to caller via register 14.

Exit-Error: If the ESTAE SVC returns a non-zero return code a X'0B7'abend is issued.

Entry-Point DEVESTAE: ESTAE recovery routine for IEEDMDEV.

Callers: RTM.

Exit-Normal: Return to caller to continue termination.

External-References:

- IECVGENA IEEDMMSG, IOSVSTSQ and IEECB808.

IEEDMMSG – Display M Message Formatting Routine

Function: IEEDMMSG formats an output buffer from a message definition and the parameters that were passed on the cell to this module.

Entry-Point IEEDMMSG: See FUNCTION.

Callers: IEEMPDM, IEENPDEV, IEEDMCPU, IEEDMCHP, IEEDMDEV, and IEEDMSTR.

Exit-Normal: Return to address provided in register 14.

Exit-Error: Abend, if requested message can not be formatted.

Entry-Point MSGESTAE: Provides information about the abend.

Callers: RTM.

Exit-Normal: Return to caller to continue termination.

| External-References: IEEMPDMM.

IEEDMSID – Display Side Information

Function: IEEDMMSID builds the IEE521I CONFIG multi-line write to operator or add the Display M side status display to an existing IEE174I MLWTO.

Entry-Point IEEDMSID: See FUNCTION.

Callers: IEEMPDM and IEEVRSCN.

Exit-Normal: Return to caller.

Exit-Error: Abend X'0B7' with reason code in register 15.

External-References:

- CMSGRSV, CM0503D, IEEDMMSG, IEEVOLSI, and IEEVORGI.

IEEDMSTR – Display Matrix Storage Processor

Function: IEEDMMSTR processes command requests to display storage where the command is DISPLAY M and the operands specified are one or more of the following:

HSA
HIGH
STOR
ESTOR

Entry-Point IEEDMSTR: Process Display Matrix storage related requests.

Callers: IEEMPDM.

Exit-Normal: Return to caller.

Exit-Error: Abend.

Entry-Point STRESTAE: Entry point ESTAE routine.

Callers: RTM.

Exit-Normal: Return to caller.

Exit-Error: None.

External-References:

- IEEDMMSG, IEECB808, IARXRCF, and IARERCF.

IEEJB840 – See IGC0203E

IEEMB803 – System Log Initialization/Writer Module

Function: This module initializes the system log, opens and closes the log data set, and writes the log buffers to the log data set.

Entry from: IEEVWAIT.

Exit to: None - always available through waiting on ECB.

IEEMB804 – Write-To-Log (WTL) Processor – SVC 36

Function: This module processes all valid WTL requests.

Entry from: IEE1603D.

Exit to: IEE1603D.

IEEMB805 – System Log Resource Initialization

Function: This module acquires and initializes log resources used through log activation.

Entry from: IEEMB803.

Exit to: IEEMB803.

IEEMB806 – System Log ESTAE Processor

Function: This module sets up two ESTAE environments for the system log, and it processes system log task ABEND situations. (The second ESTAE handles abnormal terminations of the first ESTAE).

Entry from: IEEMB803 (ESTAE/ABEND interface).

Exit to: ESTAE/ABEND interface.

IEEMB807 – System Log Message Module

Function: This module issues messages for the system log task.

Entry from: IEEMB803, IEEMB806.

Exit to: Caller.

IEEMB808 – Master Trace Data Entry Routine

Function: This module adds entries to the master trace table.

Entry from: IEETRACE macro processing.

Exit to: Caller.

IEEMB809 – Master Trace Create/Deactivate Routine

Function: This module initializes a new master trace table or deactivates an existing master trace table.

Entry from: IEETRACE macro processing.

Exit to: Caller.

IEEMB810 – Performance Group Reset Module

Function: This module passes performance group reset parameters to the system resources manager (SRM).

Entry from: IEEVWAIT.

Exit to: End of task.

IEEMB811 – SET Keyword Scanner

Function: This module scans for various SET keywords and passes control to the appropriate keyword processor.

Entry from: IEEVWAIT.

Exit to: End of task.

IEEMB812 – SET IPS, Installation Control Specification, OPT Processor

Function: This module invokes a list processor to process the data in the appropriate parmlib member (IEAIPSxx, IEAICSxx, or IEAOPTxx). It also notifies the system resources manager of the changes.

Entry from: IEEMB811.

Exit to: IEEMB811.

IEEMB813 – UNLOAD Command Syntax Scanner

Function: This module scans the information in the UNLOAD command and validates the syntax.

Entry from: IEEVWAIT.

Exit to: End of task.

IEEMB814 – Message Module for Parmlib Process

Function: This module issues messages for IEEMB811 and IEEMB878.

Entry from: IEEMB811 and IEEMB878.

Exit to: Caller.

IEEMB815 – CHNGDUMP Command Processor

Function: This module implements the CHNGDUMP operator command. IEEMB815 scans the operand portion of the command and updates the RTCT dump options lists.

Entry from: IEE0403D via BALR.

Exit to: IEE0003D.

IEEMB816 – Master Trace Functional Recovery Routine

Function: This module performs error recovery for master trace routines IEEMB808 and IEEMB809.

Entry from: RTM via FRR stack.

Exit to: Caller of master trace.

IEEMB817 – Master Trace Table Formatter

Function: This module formats the Master Trace Table entries in a first-in-first-out order. Information formatted includes a tag, immediate data, and the message data.

Entry from: AMDPRUIM via ATTACH.

Exit to: AMDPRUIM.

Called routines: Print Dump Storage Access Service Routine and Print Dump Print Service Routine.

IEEMB819 – General WTO User Exit Load/Delete Routine

Function: This module processes K M commands to activate and deactivate the general WTO user exit, IEAVMXIT.

Entry-Point IEEMB819:

Callers: IEE790D

Exit-Normal: To dispatcher

Exit-Error: None

External-References:

Routines:

- IEE0503D

IEEMB820 – SMF Initialization Router Module

Function: This module starts the SMF address space initialization process during master scheduler initialization. IEEMB820 sets up the address space attributes and parameters, specifying IFASTART as the SMF address space create routine, then links to IEEMB881. After the address space is created, IEEMB820 links to IEFSMFIE to start interval recording for the master scheduler address space.

Entry from: IEEMB860.

Exit to: Caller.

Called routines: IEEMB824, IEFSMFIE and IEEMB881 via link.

IEEMB821 – SMF Input Merge Module

Function: This module processes the SYS1.PARMLIB member containing SMF options. It parses the values, prompts the operator for changes, and sets the resulting values in the SMCA.

Entry from: IFASSMF, IFASTART, IFATSMF.

Exit to: Caller.

Called routines: IEEMB822, IEEMB824, IEEMB832, IEEMB833, IEEMB878.

IEEMB822 – SMF Data Set Initialization

Function: This module prepares SMF data sets for recording by:

- Dynamically allocating the data sets specified in the data set parameter.
- Pre-formatting the data sets, if necessary.
- Creating SMF buffers (BQES).
- Calls IEEMB829 to open the data sets for recording.

Entry from: IFASMF at IPL time; IFATSMF SET SMF time; IEEMB821.

Exit to: Caller.

Called routines: IEEMB824, IEEMB829.

IEEMB823 – SMF Initialization Record Processor

Function: At IPL time, this module issues the initialization records (types 0, 8, 22). It issues SVC 78 to produce a fifth record (type 19). Additionally, it prompts the operator for the IPL reason. At SET SMF and SETSMF time, this module produces a type 90 record containing the new SMF options. It also sets up accounting for early address spaces.

Entry from: IFASTART at IPL time; IFATSMF at SET SMF time to build record type 90; IFASSMF at SETSMF time to build record type 90.

Exit to: Caller.

Called routines: IEEMB824, IFAINIT, IEEMB830 via SMFEWTM.

IEEMB824 – SMF Message Processor Module

Function: This module builds and issues messages according to the parameters passed to it by the caller. Messages issued are WTOs, WTORs, multi-line WTOs, and PUTs to a message data set.

Entry from: IEECB913, IEECB916, IEEMB820, IEEMB821, IEEMB822, IEEMB823, IEEMB825, IEEMB827, IEEMB829, IEEMB831, IEEMB832, IEEMB833, IEEMB835, IFASMFDP, IFADSMF, IFASMF, IFASSMF, IFATSMF, IFASTART.

Exit to: Caller.

Called routines: IEEMB826.

IEEMB825 – SMF Control Task Recovery (ESTAE)

Function: If the SMF control task (IFASMF) ABENDs, this module performs recovery. Its functions include dump processing, message processing, retry or SMF recording clean-up, and global resource clean-up.

Entry from: RTM.

Exit to: Caller.

Called routines: IEEMB824, IEEMB827 the global clean-up subroutine, IEEMBFGR).

IEEMB826 – SMF Message Processor Language Parts Table

Function: This module contains translatable language parts that IEEMB824 uses to build SMF messages.

Entry from: Not applicable (non-executable code).

Exit to: Not applicable (non-executable code).

IEEMB827 – SMF Initialization Recovery (ESTAE)

Function: This module manages recovery from failures that occur during SMF address space initialization It sends messages to the operator, creates an SDUMP, and cleans up global resources. After initialization is complete, IEEMB825, the recovery routine for the SMF control task uses the global resource clean up subroutine. IEEMB825 branch-enters IEEMB827 at the secondary entry point (IEEMFGR).

Entry from: RTM.

Exit to: Caller.

Called routines: IEEMB824, branch to TQEDEQUE.

IEEMB828 – SMF Message Processor Table

Function: This dummy CSECT is part of the SMF message processor, IEEMB824. It contains a series of indexes used by IEEMB824 to locate messages.

Entry from: Not applicable (non-executable code).

Exit to: Not applicable (non-executable code).

IEEMB829 – SMF Utility Task

Function: This module completes SMF data set initialization, then handles SMF functions that require TCB mode:

- Completing the SMF data set switch.
- Ensuring that SMF data sets were dumped by the operator as requested.
- Processing SMF data sets for the SET command.
- Handling I/O errors on SMF data sets.
- Issuing exit disabled message (IEE952I).
- Issuing no more buffer space message (IEE979W).

Entry from: IFASMF, IFASSMF, IFATSMF, IEEMB822.

Exit to: Caller.

Called routines: IEEMB824.

IEEMB830 – SMF Writer SVC

Function: This module completes the header section of all SMF non-user records; it invokes either the IEFU83 or IEFU84 exit. If a record is not suppressed by an exit, IEEMB830 calls IFAPCWTR to move the record to the buffer in the SMF address space.

Entry-point IEEMB830 Issuer of the SMFEWTM, BRANCH=YES.

Entry-point IGC0008C SVC FLIH.

Exit to: Caller.

Called routines: IFAPCWTR.

IEEMB831 – SMF Syntax Analyzer Module

Function: This module is used by IEEMB832 to analyze input text and return a number corresponding to the data description. It also processes syntax and invalid option errors.

Entry from: IEEMB832.

Exit to: Caller.

Called routines: IEEMB824 via BALR.

IEEMB832 – SMF Queue Options Module

Function: This module parses the input text. If no errors are found, it adds the option to the SMF option chain.

Entry from: IEEMB821 or IFASMFDP.

Exit to: Caller.

Called routines: IEEMB824, IEEMB831.

IEEMB833 – SMF List Options Module

Function: This module lists the SMF options currently in effect. The list is directed to the console in the case of IPL or SET or to SYSPRINT in the case of the dump program (IFASMFDP).

Entry from: IEEMB821, IFASMFDP, or IFADSMF.

Exit to: Caller.

Called routines: IEEMB824.

IEEMB834 – SMF Writer SRB

Function: This module takes the VSAM control intervals from the buffers and writes them to the SMF data set.

Entry from: Dispatcher (IEAVEDS0).

Exit to: Caller.

Called routines: POST (IEAOPT01).

IEEMB835 – SMF SET Command Router

Function: This module starts processing for the SET SMF command. IEEMB835 either creates an SMF address space if none is present, or waits for any SET or DISPLAY command in process to complete. It then posts IFASMF to invoke IFATSMF to execute the SET SMF command.

Entry from: IEEMB811.

Exit to: Caller.

Called routines: IEEMB824, IEEMB881 via LINK, IFASMF via POST.

IEEMB836 – Interval Data Collection Module

Function: This module is scheduled when the SMF interval timer for a job expires. Running in SRB mode, it builds and writes interval records (types 30 and 32).

Entry from: Dispatcher (IEAVEDS0).

Exit to: Caller.

Called routines: IEFTB727, IEFTB728.

IEEMB837 – SMF Option Tables

Function: This module consists of the following CSECTS:

- SMFOPTAB - A table of valid keywords.
- SMFDEFLT - A table of SMF writer default options.
- DMPDEFLT - A table of default options for the SMF dump program (IFASMFDP).

Entry from: Loaded by IFASTART; referenced in IEEMB821 and IFASMFDP.

Exit to: Not applicable (non-executable code).

IEEMB838 – SMF Macro Module

Function: This module performs the processing invoked by the SMF macros.

Entry from: SMF macros: SMFEWTM, SMFRTEST, SMFDETAL, SMFINTVL, and SMFEXIT.

Exit to: Caller.

IEEMB839 – SMF Timer Program

Function: This module is branch entered by the caller to enqueue or dequeue timer elements. The module handles four types of processing: MAXDORM, STATUS, INTERVAL for each ASID, and dump detector.

Entry from: IFASTART, IFATSMF, IFAINIT, IFAEASIL, IFAEASI.

Exit to: Caller.

IEEMB842 – SMF MAXDORM and STATUS Processor

Function: This module performs STATUS and MAXDORM processing. For STATUS, it builds and writes the type 23 statistics record. For MAXDORM, if the current record has been in the SMF buffer longer than the interval specified by the MAXDORM keyword in the SMFPRMxx parmlib member, IEEMB842 schedules the SMF writer SRB to force the data to be written to the SMF data set.

Entry from: Scheduled by IEEMB839 at the expiration of the MAXDORM and STATUS time intervals.

Exit to: Caller.

IEEMB846 – TSO Command Table for Accounting

Function: This module includes all commands counted in the SMF type 32 record. This module contains no executable code.

IEEMB847 – SMF Message Language Part Table

Function: This dummy CSECT is contained in the SMF message processor language parts table, This module contains no executable code.

IEEMB848 – SMF Dynamic DD Data Collector Module

Function: This module saves EXCP counts (1) for data sets dynamically unallocated and (2) for all data sets at dynamic deconcatenation/concatenation. The counts are then available for the SMF type 30 record.

Entry from: IEFDB4F9.

Exit to: Caller.

IEEMB860 – Master Region Initialization Routine

Function: See *Component Diagnosis: Master Scheduler* for a module description of IEEMB860.

IEEMB876 – SET Command Keyword Table

Function: This module contains the SET command keywords processed by IEEMB811.

Entry from: IEEMB811.

Exit to: IEEMB811.

IEEMB877 – Parmlib Variable Messages

Function: This module contains the messages for parmlib processing.

Entry from: IEEMB814.

Exit to: IEEMB814.

IEEMB878 – SYS1.PARMLIB Read Routine

Function: This module reads records from a specified member of SYS1.PARMLIB.

Entry from: IEAVN600, IEECB805, IEEMPDEV, IEFJSIN2, IEEMB888.

Exit to: Caller.

IEEMB879 – Message Trace Dump Exit Routine

Function: This module is invoked by SVC dump and uses SVC dump services to dump the master trace table and its related control blocks. It schedules an SRB (IEEMB880) to run in ASID 1 and moves the master trace data to a common area buffer. It invokes the SVC dump exit output service (IEAVTSEO entry in IEAVTSDO) to write the master trace data on the dump data set.

Entry from: IEAVTSDU via BALR.

Exit to: Caller.

IEEMB880 – Master Trace Dump Exit Data Move Routine

Function: This module runs as an SRB in the address space containing the master trace table data. It is scheduled by IEEMB879 for every page of master trace data to be dumped. It moves one page (4K) of master trace data to a common buffer and then posts the ECB (via IEAVSY50) that IEEMB879 is waiting on.

Entry from: Dispatcher (IEAVEDS0) as an SRB scheduled by IEEMB879.

Exit to: IEAVEDS0.

IEEMB881 – System Address Space Create Routine

Function: This module creates a system address space either before or after master scheduler initialization is complete. The address space can be created with a limited-function start or a full-function start. If it is created with a full-function start, IEEMB881 passes the START parameters (input to IEEMB881) to the START command syntax check routine (IEEVSTAR).

Entry from: IEEVIPL, a resource initialization module (RIM), or any other module in supervisor mode.

Exit to: Caller.

Called routines: IEAVEMCR, IEAVEMRQ, IEEVWAIT.

Entry-point EAESTAE This routine receives control if an ABEND occurs during system address space create processing. It fills in the SDWA (if one exists), takes a dump, and passes control to RTM, which attempts a retry at entry-point EARETRY.

Entry from: RTM.

Exit to: RTM.

Entry-point EARETRY This entry-point sets the return and reason codes, releases storage for the CSCB, terminates the address space (if one was created), and terminates the ESTAE.

Entry from: RTM.

Exit to: Caller of IEEMB881.

IEEMB882 – Positional Parser Routine

Function: This module syntax checks a record using a rules list associated with the record. The rules list specifies the number of parameters in the record along with the minimum and maximum length of each parameter. It also indicates if the parameter is optional and if it can be enclosed in apostrophes.

Entry from: IEECB805, IEEJSIN2, IEECB812.

Exit to: Caller.

IEEMB883 – System Address Space Initialization WAIT/POST Routine

Function: This module waits for an event (indicated by the event code passed as input) to complete. If event code 1 is specified, IEEMB883 waits for master scheduler initialization to complete.

Entry from: IATINXM, IEAVTSAI, IEAVXMAS, IEEPRWI2, IEFAB4I1, ISGNASIM.

Exit to: Caller.

Entry-point WPESTAE This entry-point receives control if an ABEND occurs during IEEMB883. It fills in the SDWA (if one exists) and takes a dump.

Entry from: RTM.

Exit to: RTM.

Entry-point WPRETRY This entry-point receives control from RTM following the execution of WPESTAE. It sets return and reason codes.

Entry from: RTM.

Exit to: Caller of IEEMB883.

IEEMB884 – Scheduled Commands Table

Function: This module contains a table that contains the verb code, module name of the command processor, and the target execution address space for each scheduled command.

Entry from: None

IEEMB887 – Generalized Parser

Function: This module checks the syntax of input records based on the parse descriptions passed to it.

Entry from: Caller.

Exit to: Caller.

Called routines: ROUT and I/O exit routines.

IEEMPDEV – Display Deviations Module

Function: This module compares the actual configuration of the system with the configuration described in the information table. It displays differences, if any, to the operator.

Entry Point IEEMPDEV: See Function.

Callers: IEEMPDM and IEECB927.

Exit Normal: Function successful, status display provided.

Exit Error: ABEND if the ESTAE SVC returns a nonzero return code.

Entry Point MPDEVSTA: ESTAE recovery routine for IEEMPDEV.

Callers: RTM.

Exit Normal: Return to RTM.

External References: IEEVRDPM, IEEVSCAN, IEEDMMSG, IEECB808, IECVMAP, IOSGEN and IOCSTSQE.

IEEMPDM – DISPLAY Matrix Command Processor

Function: This module displays, to the operator, the following information: status of processor, channel paths, I/O engines, storage elements and device paths; high storage address; storage status (on/off line).

Entry Point IEEMPDM: Process DISPLAY MATRIX command.

Callers: IEECB808.

Exit Normal: Return to the caller.

Exit Error: A XX'0B7' ABEND occurs if the ESTAE SVC returns a nonzero return code.

Entry Point ESTARTN: Entry point of ESTAE error routine. This entry will receive control if there is a program or hardware error. All held resources will be freed at this point. There is no retry, and all failures are percolated to the caller.

Callers: RTM.

Exit Normal: Return to RTM.

External References: IEAVMSF, IEEMPDEV, IEEDMCHP, IEEDMDEV, IEEDMMSG, IEEDMSTR, and IEEVSCAN.

IEEMPS03 – QUIESCE Command Processor

Function: This module initiates processing to put the system into a restartable wait state with wait state code "ccc."

Entry from: IEEVWAIT.

Exit to: Caller.

IEEMPVST – CONFIG Storage Processor

Function: This routine either makes a contiguous range of 4K blocks of storage available on-line or takes a range of blocks off-line.

Entry from: IEEVWAIT via ATTACH, IEEVSTOR.

Exit to: Caller.

IEERDUMP – Reconfiguration Dump Routine

Function: This module takes an SDUMP of reconfiguration data areas.

Entry-Point IEERDUMP: See FUNCTION.

Callers: IEERESTA.

Exit-Normal: Return to caller.

Exit-Error: None.

External-References:

- None.

IEEPRTN2 – STC Free Region Routine

Function: This module determines if there is a CSCB in existence at the end of started task control processing. If there is, IEEPRTN2 frees the CSCB.

Entry from: IEESB605, IEEPRWI2, IEEVMNT1, IEEVSTAR, or IKJEFLA via XCTL.

Exit to: Region control task via SVC 3.

IEEPRWI2 – STC Get Region Routine

Function: This module gets a new region for started task control processing. If a system component address space is being created, this routine invokes the address space initialization routine. It also determines whether a START, MOUNT, or LOGON command is being processed, and based on that decision invokes the appropriate command processor.

Entry from: IEAVAR00 by means of ATTACH.

Exit to: IEEVSTAR, IEEVMNT1, IKJEFLA, or IEEPRTN2.

Called routines: IEEMB883 via BALR, a system address space initialization routine via LINK, IFAESIL via LINK.

IEERECON – DAT OFF Restart FLIH

Function: To load the control registers, floating point registers, clock comparator, and CPU timer of the CPU on which it is running.

Entry from: When operator hits RESTART after the system has been stopped via a Quiesce command.

Exit to: The RESTFLIH routine in IEESTPRS via a LPSW instruction.

IEERESTA – Reconfiguration Recovery Routine

Function: This routine performs ESTAE and FRR processing for reconfiguration modules, logs errors, takes dumps, frees resources, and returns to retry point.

Entry-Point IEERESTA: Performs ESTAE recovery processing.

Callers: RTM.

Exit-Normal: Return to caller.

Entry-Point IEERETRY: Sets up the reconfiguration retry environment after the ESTAE routine has returned to RTM.

Callers: RTM.

Exit-Normal: Branch to mainline retry point via BR 15.

Entry-Point IEERFRR: Performs FRR processing.

Callers: RTM.

Exit-Normal: Return to RTM requesting percolation.

External-References:

- IEERDUMP, IEERSYMR, and IEEVMESS.

IEERSYMR – Reconfiguration SYMREC Routine

Function: This routine writes to LOGREC, a symptom record (SYMREC) that describes a failure or other unusual event in reconfiguration.

Entry-Point IEERSYMR: See FUNCTION.

Callers: Reconfiguration modules.

Exit-Normal: Return to caller.

Exit-Error: None.

External-References: None.

IEESAFF – CPU/VF Affinity Status Routine

Function: This routine determines the CPU/VF job affinity status.

Entry-Point IEESAFF: See FUNCTION.

Callers: IEEVCPLF and IEEVVFLF.

Exit-Normal: Return to caller.

Exit-Error: None.

External-References:

- IEEUMVCL.

IEESB601 – SWA and TIOT Initialization for Private Catalog Processing

Function: This module creates an SWA control block structure for use by the dynamic allocation routines. These routines are invoked to process private catalogs for START, MOUNT, and LOGON commands.

Entry from: IEESB601 via BALR.

Exit to: IEESB601 via branch.

IEESB605 – Job Scheduling Subroutine

Function: This module sets up the environment necessary to initiate a START, MOUNT, or LOGON command task. It builds the SSOB, SSIB, local job scheduling parameter lists, and an SWA structure for the task; then it invokes the initiator. When initiator processing is done, IEESB605 deletes the SWA structure it created and frees the CSCB for the task.

Entry from: IEEVJCL via XCTL for a START or MOUNT command, or IKJEFLB for LOGON commands.

Exit to: IEFSD060, an initiator routine via LINK, to initiate the command task; IEEPRTN2 via XCTL, for end processing of a START or MOUNT command; or IEEPRTN2 via XCTL, for end processing of LOGON commands.

IEESB606 – STC Get JSCB Routine

Function: This routine builds a JSCB during storage initialization for a START, MOUNT, or LOGON command.

Entry from: IEAVEMIN via CALL.

Exit to: Caller.

IEESB665 – STC Recovery Exit Routine

Function: This module schedules a retry of STC when one of the following conditions has occurred:

- A program check.
- A machine check.
- An ABEND.
- Depression of the RESTART key on a console.

If percolation from a lower level recovery routine has occurred, IEESB665 continues percolation.

Entry from: Recovery termination management via LINK.

Exit to: Caller via branch.

IEESB670 – Job Scheduling Subroutine Recovery Exit Routine

Function: This module schedules a retry of IEESB605 when one of the following conditions has occurred:

- A program check.
- A machine check.
- An ABEND.
- Depression of the RESTART key on a console.

If percolation from a lower level recovery routine has occurred, IEESB665 continues percolation.

Entry from: Recovery termination management via LINK.

Exit to: Caller via branch.

IEESCPVF — Obtain CPU and VF Physical Status

Function: This routine obtains the CPU and VF physical status.

Entry-Point IEESCPVF: See FUNCTION.

Callers: IEEVCPR.

Exit-Normal: Return to caller.

Exit-Error: None.

External-References: IEEUMVCL, IEEUSRVC, and IEERSYMR.

IEESTPRS — Stop/Restart Subroutine

Function: This module stops the operating system in a manner that allows a restart interruption to bring the system back into operation.

Entry from: IEEVLDWT and any routine in supervisor state that wants to put the system into a restartable state.

Exit to: Caller.

IEESTSTR — Processor Stop Routine

Function: This module puts a specified CPU into a wait state with the wait state code specified by the caller.

Entry from: IEESTPRS and IEEVDCCR.

Exit to: Caller.

IEEVALST — Validate Page Routine

Function: This module sets to zero the storage and storage key of a 4K section of real storage.

Entry from: IEEMPVST, IEEVSTGL.

Exit to: Caller.

IEEVCHPR — CONFIG CHP Reconfiguration Processor

Function: This module processes requests for channel path (CHP) reconfiguration.

Entry from: IEEVCONF, IEEVIOSR.

Exit to: Caller.

Called routines: IECVIOPM, IOSVARY, IEEVIOSR, IEEVDEV, IGC0407B, IOSVFCHP, IEEVMESS, IEFENFFX, IEFAB49C, IEAVMSF.

IEEVCONF — CONFIG Command Reconfiguration Processor

Function: This module interprets the parameter table built by IEEVSCAN and calls the appropriate reconfiguration module.

Entry to: IEECB927.

Exit to: Caller.

Called routines: IEEVCHPR, IEEVIOSR, IEEVSTEL, IEEVSTGP, IEEVCPR, IEEVMESS, IEAVMSF.

IEEVCPH — CPU/VF Physical Reconfiguration Routine

Function: This routine configures the target CPU/VF physically online/offline.

Entry-Point IEEVCPH: See FUNCTION.

Callers: IEEVCPR.

Exit-Normal: Return to caller.

Exit-Error: None.

External-References:

- IEEUMVCL, IEEUSRVC and IEEVMESS.

IEEVCPLF — Configure CPU Logically offline

Function: This routine configures the target CPU logically offline.

Entry-Point IEESVPLF: See FUNCTION.

Callers: IEEVCPR.

Exit-Normal: Return to caller.

Exit-Error: None.

External-References: IEEVTOD, IEESAFF, IEEUMVCL, IEEUSMF, IEEVCPRA, IEEVMESS and IEEVCVSR.

IEEVCPLN — Configure CPU Logically Online

Function: This routine configures the target CPU logically online.

Entry-Point IEEVCPLN: See FUNCTION.

Callers: IEEVCPR.

Exit-Normal: Return to caller.

Exit-Error: None.

External-References:

- IEEVTOD, IEEUMVCL, IEEVCPRA, IEERSYMR and IEEVCVSR.

IEEVCPR — CPU and VF Reconfiguration Processor

Function: This routine configures a CPU and/or a VF online or offline.

Entry-Point IEEVCPR: See FUNCTION.

Callers: IEEVNP00 and IEEVCONF.

Exit-Normal: Return to caller.

Exit-Error: Issue ABEND to give control to ESTAECPR, IEEVCPR's ESTAE routine.

Entry Point ESTAECPR: Provide ESTAE recovery for module IEEVCPR.

Callers: RTM.

Entry Point FRRCPR: Provide FRR recovery for the STARTUP routine when the dispatcher lock or restart resource is held.

Exit Normal: Return to RTM.

External-References: AHLVCOFF, AHLVCON, IEAVEBBR, IEAVRCAN, IEAVRNOT, IEAVRSSC,IEAVTR1N, IOSVCPRM, IOSVSHUP, IEEVMESS, IGFPBUCR, and SVC 122.

IEEVCPRA — Obtain or Release CPU-related Area Routine

Function: This routine obtains or releases CPU-related areas.

Entry-Point IEEVCPRA: See FUNCTION.

Callers: IEEVCPLN and IEEVCPLF.

Exit-Normal: Return to caller.

Exit-Error: None.

External-References: IEAVEBBR, IEAVINIT, IEAVTRIN, IEEUMVCL, IEERSYMR, IEEVMESS, IGFPBUCR, and IOSVCPRM.

IEEVCVSR — CPU/VF Physical Reconfiguration Routine

Function: This routine invokes IEEVQUIT or IEEVWKUP on the target CPU and monitors their execution.

Entry-Point IEEVCVSR: See FUNCTION.

Callers: IEEVCPLN, IEEVCPLF and IEEVVFLF.

Exit-Normal: Return to caller.

Exit-Error: None.

External-References:

- IEEUMVCL, IEEVQUIT and IEEVWKUP.

IEEVC801 — PFK Definitions for Display Consoles

Function: Non-executable table of default PFK definitions for all display consoles. It is used when console initialization is unable to load PFK definitions from SYS1.DCMLIB.

Entry from: N/A.

Exit to: N/A.

IEEVDCCR — Disabled Console Communication Routine

Function: This module enables disabled recovery routines to communicate with the system operator through the master console (or its alternate) during system recovery processing.

Entry from: IEESTPRS, IEEVEXSN, IGFPXMFA.

Exit to: Caller.

IEEVDEV — Device Subroutine

Function: This routine obtains information regarding the condition of available paths to a device and provides that information to the caller.

Entry from: Allocation, DDR, VAP, CONFIG CPU, IEEVPTHR, IEEVCHPR.

Exit to: Caller.

IEEVEXSN — Excessive Spin Notification Routine

Function: This routine is called by various system routines to communicate with the operator when an excessively long spin is detected. The operator is asked if the spin should continue or if an ACR condition for the other processor should be simulated.

Entry from: IEAVELK, IEAVERI, IEEVLDWT, IEAVESGP, and IEAVINV.

Exit to: Caller.

IEEVG800 – System Termination Message Channel Programs

Function: Non-executable table of channel programs. It is used by the system termination routine when writing a message to the master console.

IEEVIOSD – CONFIG CHP FORCE I/O SIDE Reconfiguration Routine

Function: This module processes the I/O SIDE for CONFIG CHP FORCE request.

Entry to: IEEVCHPF.

Exit to: Caller.

Called routines: IEAVMSF.

IEEVIOSR – I/O SIDE Reconfiguration Processor

Function: This module processes requests for the I/O SIDE reconfiguration.

Entry to: IEEVCONF, IEEVCHPR.

Exit to: Caller.

Called routines: IEEVCHPR, IEAVMSF, IEEVMESS.

IEEVIPL — Master Scheduler Base Initialization

Function: Initializes the master scheduler, invokes the subsystem-interface initialization function (IEFSJINT), and other initialization routines, specifically IEAVN701, IEFSD060, IEFAB4I0, IKJEFXSR, and IEFQBINT.

Callers: IEEMB860.

Exit IEFSD060.

Error Entry At entry pont IPSTAR from failure detection or from ABEND.

Error Exit To a wait state.

IEEVJCL – Job Control Language Build Routine

Function: This module builds internal JCL text for a command task that represents a START, MOUNT, or LOGON command.

Entry from: IEEVSTAR via branch for a START command; IEEVMNT1 via branch for a MOUNT command; IKJEFLA via branch for a LOGON command.

Exit to: IEESB605 via XCTL.

IEEVLDWT – Load Wait State Processor

Function: This module serves as the interface for loading a wait state. For restartable wait states, it interfaces with the stop/restart subroutine (IEESTPRS). For non-restartable wait states, it calls IGFPTERM.

Entry from: IEEMPS03, IECVIRST, and IECVRSTI.

Exit to: Caller.

IEEVMESS – Reconfiguration Message Module

Function: IEEVMESS performs message processing for all the reconfiguration modules. It is called to:

- Obtain storage for and initialize a message buffer.
- Place a specified message in a given message buffer.
- Write out the messages in a specified message buffer.
- Issue a specified message as a WTO or WTOR.
- Free message buffer storage.

Called by: Reconfiguration modules.

Exit to: Caller.

Error exit: RTM via an ABEND macro.

Called routines: IEEVTEXT.

IEEVMNT1 – MOUNT Command Syntax Check Routine

Function: This module checks a MOUNT command and its parameters for correct syntax. It also uses this input to build a START descriptor table (SDT) that contains internal JCL.

Entry from: IEEPRWI2 via XCTL.

Exit to: IEEVJCL via branch for normal processing; IEEPRTN2 via XCTL for error processing.

IEEVMNT2 – MOUNT Command Processor

Function: This module checks the unit control block (UCB) associated with the device to be mounted. If the device is a tape drive marked reserved, IEEVMNT2 issues an error message. Otherwise, it also checks the MOUNT command input buffer (CIB) for the use attribute specified for the device.

Entry from: IEFSD263, an initiator routine, via ATTACH.

Exit to: IEFSD263 via branch.

IEEVOLSI – Obtain Logical Status Information

Function: Obtains the logical status of CPs, VFs, CHPs, I/O SIDES, and returns the information to the caller in bit map arrays.

Entry from: IEEDMSID.

Exit to: Callers.

IEEVORGI – Obtain Resource Group Information

Function: Calculates and returns the size of the storage block required to hold the side information. Obtains and returns the side information consisting of the components of the specified side.

Entry from:

- IEAVNP27 during system initialization.
- IEEDMSID after system initialization.

Exit Normal: Callers:.

Exit Error: Abend X'0B7' with reason code.

External References: SVC 122 extended SVC routing code 6.

IEEVPTH – VARY Path Command Processor

Function: This routine interfaces with IEEVPTHR to make individual paths to devices either available or unavailable for I/O processing.

Entry from: IEEVWAIT.

Exit to: Caller.

Called routines: IEEVPTHR.

IEEVPTHR – VARY Path Reconfiguration Routine

Function: This module makes a path to a device either available or unavailable for I/O processing. If the last path to an unallocated but available device is made unavailable, the device is also made unavailable. If a path to a device that is unavailable for hierarchical reasons is made available, the device itself is also made available if it has an operational path.

Entry from: IEEVPTH.

Exit to: IEEVPTH.

IEEVRDPM – SYS1.PARMLIB Member Read Routine

Function: This module reads a member from SYS1.PARMLIB and passes it to IEEVSCAN for syntax scanning.

Entry From IEEMPDEV, IEECB927.

Exit To Caller.

Called Routines IEEMB878.

IEEVRSCN – CONFIG Read Screen Module

Function: This module displays the system configuration on the operator's console and reads the reply.

Entry From IEECB927.

Exit To Returns to the caller.

Called Routines Communications Task Message Service, IEAVMSF.

IEEVSCAN – Syntax Scan Routine

Function: The Syntax Scan Routine for the CONFIG command and the DISPLAY M = CONFIG option of the DISPLAY M command scans control cards and builds a parameter table.

Entry From IEECB927, IEEVRSCN, IEEVRDPM.

Exit To Caller.

IEEVSEND – Operator SEND Command Main Control

Function: This module provides a console or TSO-terminal operator with a means of communicating with other TSO and/or console operators.

Entry from: IEEVWAIT.

Exit to: End of task.

IEEVSMSG – STC Message Module

Function: This module contains the text of messages issued by the started task control routines.

Entry from: IEEVMNT2 or IEESB605 via CALL.

Exit to: Caller via branch.

IEEVSND2 – SEND Command Mail Handler

Function: This module saves the mail for specified users in the mail section of the SYS1.BRODCAST data set.

Entry from: IEEVSND9.

Exit to: IEEVSND9.

IEEVSND3 – SEND Command List Handler

Function: This module lists all notices from or recovers a specific notice from, the SYS1.BRODCAST data set for the sending operator.

Entry from: IEEVSND9.

Exit to: IEEVSND9.

IEEVSND4 – SEND Command Cleanup Routine

Function: This module issues informational, warning, and error messages to the operator.

Entry from: IEEVSEND.

Exit to: IEEVSEND.

IEEVSND5 – Operator SEND Command I/O Routine

Function: This module performs I/O operations for the SEND command.

Entry from: IEEVSND2, IEEVSND3, IEEVSND8, IEEVSND9.

Exit to: Caller.

IEEVSND6 – SEND Command Now Processor

Function: This module issues the SEND command text to the specified recipients.

Entry from: IEEVSEND.

Exit to: IEEVSEND.

IEEVSND8 – SEND Command Notice Handler

Function: This module adds notices to or deletes notices from the SYS1.BRODCAST data set.

Entry from: IEEVSND9.

Exit to: IEEVSND9.

IEEVSND9 – SYS1.BRODCAST Data Set Access Controller

Function: This routine coordinates the operator usage of the SYS1.BRODCAST data set.

Entry from: IEEVSEND.

Exit to: IEEVSEND.

IEEVSTAR – START Command Syntax Check Routine

Function: This module checks a START command and its parameters for correct syntax. It uses this input to build a START descriptor table (SDT) that contains the internal JCL for the task to be started.

Entry from: IEEPRWI2 via XCTL.

Exit to: IEEVJCL via branch, for normal processing; IEEPRTN2 via XCTL, for error processing.

IEEVSTEE – Storage Element Reconfiguration ESTAE

Function: This module performs recovery from a problem during storage element reconfiguration.

Called by: RTM.

Exit to: Caller.

| Called Routines: IEEVMESS.

IEEVSTEL – Storage Element Reconfiguration Processor

Function: IEEVSTEL varies a storage element online or offline.

Called by: IEEVSTOR.

Exit to: Caller.

Error exit: RTM via an ABEND macro.

Called routines: IEEVSTGL, IEEVSTFA, IEEVSTEE, and IEEVMESS.

IEEVSTFA – Logical Storage Element Offline Reconfiguration

Function: This routine varies a storage element logically offline by varying offline original storage subincrements and/or alternate storage subincrements selected to be varied offline.

Entry-Point IEEVSTFA: See module function.

Callers: IEEVSTEL.

Exit-Normal: Return to caller.

Exit-Error: Issues an abend with a completion code of X'0B7'.

External-References: IARXRCF and IEEVSTGL.

IEEVSTGL – Logical Storage Reconfiguration

Function: IEEVSTGL logically varies a set of 4K frames of real or extended storage online or offline.

Called by: IEEVSTGP, IEEVSTEL, IEEVSTEE, IEEVSTPE, and IEEVSTFA.

Exit to: Caller.

Error exit: RTM via an ABEND macro.

External References: IARXRCF, IARERCF, IEEVALST, and IEEVMESS.

IEEVSTGP – Storage Reconfiguration Routine

Function: IEEVSTGP varies either a range or an amount of real storage online or offline.

Called by: IEEVCONF.

Exit to: Caller.

Error exit: RTM via an ABEND macro.

Called routines: IARXRCF, IEEVMESS, IEEVSTPE, and IEEVSTGL.

IEEVSTOP – Vary CPU Stop Routine

Function: IEEVSTOP stops the processor on which it is running. In order to stop the processor, IEEVSTOP ensures that the CSD indicates that the processor is off-line, resets the processor's prefix register to zero, and issues a SIGP STOP instruction to the processor.

Entry from: Machine check handler, timer, vary CPU offline.

Exit to: None.

IEEVSTOR – VARY STOR Command Processor

Function: IEEVSTOR controls VARY STOR command processing. It:

- Determines if the command is valid and has been issued from an authorized source.
- Routes control to the proper routine to perform the reconfiguration.
- Issues the appropriate reconfiguration message(s) after the storage has been varied.

Called by: The master scheduler (IEEVWAIT) via an ATTACH macro after the operator issues a VARY STOR command.

Exit to: Caller.

Error exit: RTM via an ABEND macro.

Called routines: IEEMPVST, IEEVMESS, IEEVSTEL, and IEEVSTGP.

IEEVSTPE – Storage Reconfiguration ESTAE Routine

Function: IEEVSTPE recovers from a problem during storage reconfiguration.

Called by: RTM.

Exit to: Caller.

Called routines: IARXRCF, IEEVMESS, and IEEVSTGL.

IEEVSTPE – ESTAE Routine for IEEVSTGP

Function: IEEVSTPE performs recovery processing for IEEVSTGP. Unless diagnostic information has already been produced, IEEVSTPE writes a record in SYS1.LOGREC and issues an SDUMP. It also issues a message telling the operator what happened before the error occurred. In some cases, it backs out of processing done prior to the error.

Called by: RTM.

Exit to: Caller.

Called routines: IEAVRCF3, IEEVMESS, and IEEVSTGL.

IEEVSTRE – ESTAE Routine for IEEVSTOR

Function: IEEVSTRE performs recovery processing for IEEVSTOR. It records the diagnostic information in SYS1.LOGREC, issues an SVC dump, and issues message IEE713I.

Called by: RTM.

Exit to: Caller.

Called routines: IEEVMESS.

IEEVVFLF – Configure VF Logically Offline

Function: IEEVVFLF configures the target VF logically offline.

Entry-Point IEEVVFLF: See module function.

Callers: IEEVCPR.

Exit-Normal: BR 14 to the caller.

Exit-Error: None.

External-References: IEESAFF, IEEUMVCL, IEEVMESS and IEEVCVSR.

IEEVVFLN – Configure VF Logically Online

Function: IEEVVFLN configures the target VF logically online.

Entry-Point IEEVVFLN: See module function.

Callers: IEEVCPR.

Exit-Normal: BR 14 to the caller.

Exit-Error: None.

External-References: IEEUMVCL.

IEEVWAIT – Master Scheduler Wait

Function: In the Master Scheduler Address Space

At system initialization time: IEEVWAIT terminates system trace (if necessary), and attaches the system log task. After system initialization, IEEVWAIT's main function is to scan the CSCB chain and process the pending commands that execute in the master scheduler address space. If invoked for a START command, IEEVWAIT posts ECB CHASWT to notify the caller when master scheduler initialization is complete.

Entry from: IEEMB860 at system initialization time; IEE0803D via POST.

Exit to: No normal exit. This is a permanent task.

Error entry: At entry-point STAE0000 from failure or from ABEND.

Error exit: To a wait state if restart fails.

In the Communications Task Address Space: During system initialization, IEEVWAIT performs no major function in the communications task address space. After system initialization, IEEVWAIT's only function in the communications task address space is to scan the CSCB chain and process the pending commands that execute in this address space.

Entry from: IEAVN701 during system initialization, IEE0803D via POST.

Exit to: No normal exit. This is a permanent task.

Error entry: At entry-point STAE0000 from failure or from ABEND.

Error exit: To the dispatcher.

IEEVWKUP – CONFIG CPU/VF Wakeup and Quit Routine

Function: IEEVWKUP is the module that actually marks the target CPU logically online. It performs two functions.

- Wakeup processing for a CONFIG CPU,ONLINE command.
- Quit processing for a CONFIG CPU/VF OFFLINE command.

Entry-Point IEEVWKUP: Prepare a CPU to be brought online for presentation to the dispatcher.

Callers: None.

Entry-Point IEEVQUIT: To stop a CPU/VF from being taken offline.

Callers: None.

External-References:

- A work area is passed to entry points IEEVWKUP and IWWVQUIT that contains a flag byte for communication with the invoker.

IEEXEDNA – Display Consoles Operand Processor

Function: This module gathers console information for the Display Consoles Command response (message IEE249I), and stores it in the DCPL and consent table (mapped by IHACTM) for use by IEECB814.

Entry from: IEECB813.

Exit to: Caller

Called routines: IEECB814.

IEE0003D – ESTAE Environment Creation Routine

Function: This routine creates an ESTAE environment routine to protect the command scheduler from ABEND situations.

Entry from: Issuer of SVC 34.

Exit to: IEE0303D.

IEE0303D – Control Block Chain Manipulator; QEDIT and MGCR Macro Function Processor

Function: This module manipulates chains. It also communicates ABTERM requests to the ABTERM routines, and it is the QEDIT processor.

Entry from: IEE0003D.

Exit to: IEE0403D.

IEE0403D – Command Scheduler Router

Function: This module scans, identifies, and verifies commands. It gives control to the proper command processor.

Entry from: IEE5403D.

Exit to: IEE0503D, command processors.

IEE0503D – Command Processing Message Assembly

Function: This module assembles and edits messages for command processors. It issues WTO or TPUT macros to write the messages.

Entry from: Command processors.

Exit to: The address passed by the caller.

IEE0603D – Immediate Routine for SET Command

Function: This module inspects the operands of the SET command and routes control to the appropriate processing module.

Entry from: IEE0403D.

Exit to: IEE6503D, IEE6603D, IEE0803D, IEE0503D.

IEE0703D – MODIFY and STOP Command Processor

Function: This module performs processing for the STOP and MODIFY commands.

Entry from: IEE0403D.

Exit to: IEE0403D.

IEE0803D – CSCB and ASCB Creation

Function: For task-creating commands, this module builds a CSCB. For address space-creating commands, this module interfaces with address space-requesting routines.

Entry from: Variable.

Exit to: IEE0003D, to post IEEVWAIT.

IEE10110 – Display C, K Processor

Function: This module writes a status display showing the CONTROL (K) command operands using a multiple-line WTO macro.

Entry from: IEE00110 via CALL.

Exit to: IEE00110.

IEE1403D – HALT, SWITCH, and TRACE Command Syntax Checker

Function: This module preprocesses the HALT, SWITCH, and TRACE commands by checking the command syntax.

Entry from: IEE0403D.

Exit to: IEE0803D, IED1303D, ISTCFF3D.

IEE1603D – LOG and WRITELOG Command Processor

Function: This module processes the commands LOG and WRITELOG (with the options CLOSE, START, or CLASS).

Entry from: IEE0403D.

Exit to: IEE0403D.

IEE20110 – Display Unit Processor – Unit Status, One

Function: This module analyzes the operands of a D U command, gets a work area, and finds the first UCB for the first unit to be displayed.

Entry from: IEE00110.

Exit to: IEE23110, IEE22110.

IEE21110 – Display Unit Processor – Unit Status, Two

Function: This module loads the device name table the first time it is needed and creates the text of each line to be displayed.

Entry from: IEE23110.

Exit to: IEE23110.

IEE22110 – Display Unit Processor – Unit Status, Three

Function: This module issues error messages, frees work area storage, and deletes the device name table.

Entry from: IEE20110, IEE23110, IEE24110.

Exit to: End of task.

IEE2303D – SMF VARY Record Handler

Function: This module writes an SMF VARY ONLINE record.

Entry from: IEE3303D.

Exit to: IEE4203D, IEE4403D.

IEE23110 – Display Unit Processor – Unit Status, Four

Function: This module determines the next unit address to be displayed and issues the WTO for the lines of the display.

Entry from: IEE20110.

Exit to: IEE21110, IEE22110.

Called routines: IEE21110, IEE24110.

IEE24110 – D U,,ALLOC Command Processor

Function: This module gathers and displays the information requested on the D U,,ALLOC command. The information displayed consists of the jobname and ASID of the job(s), started task(s), or TSO user(s) to which a unit is allocated.

Entry from: IEE23110.

Exit to: IEE22110, IEE23110.

Called routines: IEECB801.

IEE3103D – VARY ONLINE/OFFLINE Processor

Function: This module varies non-console units to either an on-line or an off-line status.

Entry from: IEE4203D, IEE4603D.

Exit to: End of task, IEECB904.

Called routines: IEFENFFX.

IEE3203D – VARY Keyword Scan

Function: This module scans first the primary keywords then the secondary keywords of the VARY command. It passes control according to the keyword found.

Entry from: IEE0403D.

Exit to: Variable.

IEE3303D – VARY ONLINE/OFFLINE/CONSOLE Syntax Scan

Function: This module checks the unit field syntax, command authority, and presence of SMF in order to route control to the appropriate processor.

Entry from: IEE3603D.

Exit to: IEE4403D, IEE4203D, IEE2303D, IEECB904.

IEE3503D – DISPLAY/TRACK Router

Function: This module routes control to a processor module based on the command operand. It processes the D T command itself.

Entry from: IEE0403D.

Exit to: Variable.

IEE3603D – VARY Command Pre-processor

Function: This module presets the environment for the VARY ONLINE/OFFLINE/CONSOLE commands. It also keeps off-line all devices without available paths.

Entry from: IEEVWAIT, IEECB904.

Exit to: IEE3303D.

IEE3703D – CANCEL/FORCE Command Handler

Function: This module cancels or forces any background and foreground (TSO) jobs that are currently executing.

Entry from: IEE0403D.

Exit to: IEE0003D.

IEE40110 – Display PFK Processor

Function: This module displays the status of the following:

PFK definitions for a console
PFK definitions in a PFK table
PFK tables available for use.

Entry from: IEECB808.

Exit to: Caller.

Called routines: IEECB808, IEE0503D.

IEE4203D – UCME Scanner for VARY Command

Function: This module does syntax validation for VARY ONLINE/OFFLINE/CONSOLE commands.

Entry from: IEE4403D, IEE3303D, IEE2303D.

Exit to: IEE4903D, IEE3103D, IEE4603D, IEECB904.

IEE4303D – Processor for VARY Master Console Command

Function: This module processes the VARY MSTCON.

Entry from: IEE3203D.

Exit to: IEE0403D.

IEE4403D – VARY Keyword Scanner

Function: This module scans the input buffer and validates keyword values.

Entry from: IEE3303D, IEE2303D.

Exit to: IEE4203D.

Called Routines IEECB815.

IEE4603D – VARY ONLINE/OFFLINE for Console Devices

Function: This module processes the VARY command keywords ONLINE and OFFLINE.

Entry from: IEE4203D.

Exit to: IEE3103D, IEECB904, SVC 3 (IGC003).

IEE4703D – VARY HARDCPY Operand Processor

Function: This module begins the processing of a VARY HARDCPY Command.

Entry from: IEE3203D.

Exit to: IEE5703D, IEE7203D.

Called Routines IEECB815.

IEE4903D – VARY Console Operand Processor

Function: This module processes the CONSOLE Operand for the VARY command.

Entry from: IEE4203D.

Exit to: End of task.

Called Routines IEE0503D.

IEE5103D – ESTAE Exit Routine

Function: This module is the SVC 34 ESTAE exit routine. It provides a storage dump in response to an ABEND situation, and it checks the validity of entries in the CSCB and CIB chains.

Entry from: ABEND STAE interface routine.

Exit to: Termination retry address in parameter list.

IEE5403D – Command Translation Routine

Function: This module translates lower case letters, initializes the XSA, finds the beginning of a command verb and writes the command to hard copy if required.

Entry from: IEE0403D.

Exit to: IEE0403D.

IEE5503D – Stop Track and Stop Monitor Processor

Function: This module processes the operands for both the STOPTR and STOPMN commands.

Entry from: IEE7503D.

Exit to: IEE6703D, IEE0003D, IEE7103D, IEE0503D.

IEE5603D – MSGRT and CONTROL Command Message Module

Function: This module issues error messages for CONTROL and MSGRT commands and for L=cca operands.

Entry from: Variable.

Exit to: IEE5903D, IEE0503D, or system.

IEE5703D – VARY HARDCPY, OFF P and VARY HARDCPY Message Processor

Function: This module removes the hardcopy function from a system when requested or issues a message as a result of a vary hardcpy command.

Entry from: IEE4703D, IEE7203D.

Exit to: IEE0403D.

IEE5903D – Error Message Module 2 for CONTROL and MSGRT Commands

Function: This module handles the second parts of messages for the appropriate commands and also for the L=cca operand errors.

Entry from: IEE5603D.

Exit to: IEE0403D.

IEE6303D – MSGRT Command Handler I

Function: This module builds the route control table (RCT) for each console. This table provides default routing values.

Entry from: IEE0403D.

Exit to: IEE0503D, IEE6403D.

IEE6403D – MSGRT Command Handler II

Function: This module formats a MSGRT command on the basis of values in the route control table.

Entry from: IEE6303D.

Exit to: IEE0403D, IEE5603D.

IEE6503D – Set Local Time

Function: This module changes or resets the local time and date. It also updates the real time timer queue element (TQE) queue.

Entry from: IEE0603D.

Exit to: IEE0603D.

IEE6603D – Set Time-of-Day Clock Routine

Function: This module sets a TOD clock with a time and date value that has been specified by an operator.

Entry from: IEE0603D.

Exit to: Caller.

IEE6703D – Processor for CONTROL and STOPTR Commands

Function: This module handles the syntax checking of the K command and schedules the STOPTR command and the K command.

Entry from: IEE7503D.

Exit to: Variable.

IEE6803D – K A,nn,nn Command Handler

Function: This module handles the processing of the K A command, which defines CRT screen areas.

Entry from: IEE6703D.

Exit to: IEE0403D, IEE5603D.

IEE6903D – K A, K T, K V,REF, and K M,REF Command Handler

Function: This module handles K T, K A, K A,NONE, K M, K M,REF, and K M,REF commands.

Entry from: IEE6703D.

Exit to: IEE0403D, IEE5603D.

IEE70110 – Halt-End-of-Day and Switch-SMF Processor

Function: This module processes HALT EOD and SWITCH SMF commands.

Entry from: Attached by IEEVWAIT.

Exit to: Caller.

Called routines: IEE90110 via BALR, IEECB860 (ESTAE), and IEEMB803 log task via POST.

IEE7103D — MONITOR Command Processor

Function: This module processes the MONITOR command operands.

Entry from: IEE7503D.

Exit to: IEE0403D, IEE0503D.

IEE7203D — VARY HARDCPY Processor II

Function: This module changes the hard copy configuration in a multiple console support (MCS) environment.

Entry from: IEE4703D.

Exit to: IEE4103D.

Called routines: IEE5703D.

IEE7503D — Routing and List Operand Processor

Function: This module processes the routing operands for DISPLAY, TRACK, CONTROL, and STOPTR commands. It also syntax checks the secondary operands (LIST, ALL, and name) for the DISPLAY and TRACK commands.

Entry from: IEE3503D, IEE0403D.

Exit to: IEE0803D, IEE5503D, IEE6703D, IEE5603D, IEE0503D, IEE9403D.,

IEE7603D — List Operand Processor for Subsystem Commands

Function: This module processes the "L=cca" operands on subsystem commands.

Callers: JES2 routine.

Exit to: JES2 routine.

IEE7703D — CONTROL V Command Handler

Function: This module processes CONTROL V commands.

Entry from: IEE6703D.

Exit to: IEE0403D, IEE5603D.

IEE7803D — CONTROL C,D Command Handler

Function: This module processes the CONTROL C,D command for cancelling an inline MLWTO.

Entry from: IEE6703D.

Exit to: IEE0403D, IEE5603D.

IEE7903D — CONTROL M Command Handler

Function: This module processes CONTROL M commands for activating or deactivating the action message retention facility, changing the buffer limits, and activating or deactivating the general WTO user exit, IEAVMXIT.

Entry-Point IEE7903D:

Callers: IEE6703D

Exit-Normal: IEE0503D, IEE5603D

Exit-Error: None

External-References: IEEMB819

IEE8B03D — CONTROL Q Command Processor

Function: This module processes the CONTROL Q command for re-routing messages.

Entry from: IEE6703D.

Exit to: IEE0403D.

IEE8C03D — CONTROL (K) N,L=CC Command Processor

Function: This routine checks the syntax on a routed K N command.

Entry from: IEE8C03D.

Exit to: Caller.

External References: IEE5603D.

IEE8103D — CONTROL C,A/E/I Command Handler

Function: This module deletes retained action messages in response to a CONTROL C,A, CONTROL C,E, or CONTROL C,I command.

Entry from: IEE6703D.

Exit to: IEE5603D.

Called routines: IEAVQ700.

IEE8603D — SETDMN Command Processor

Function: This module passes new domain description values to the system resources manager (SRM).

Entry from: IEE0403D.

Exit to: IEE0403D.

IEE90110 – HALT/SWITCH Message Module

Function: This module assembles and edits messages for HALT and SWITCH commands, and issues a WTO.

Entry from: IEE70110.

Exit to: IEE70110.

IEE9403D – MSS Preprocessor

Function: If MSS is active this module prepares MSS commands for interfacing to MSS routines, then posts the MSS routines.

Entry from: IEE0403D, IEE1403D, IEE3203D, IEE7503D.

Exit to: Caller.

IEFAB4A0 – Common Unallocation Control

Function: This module controls the processing necessary to unallocate a request that has been allocated by means of either JCL or a dynamic allocation. The functions provided are:

- Disposition processing.
- Disconnecting of private catalogs.
- Data set release (DEQ).
- Unit unallocation.
- Volume release (DEQ).
- Release of DSAB/TIOT entries.

Entry from: IEFAB4DE, IEFAB477, IEFAB490, IEFAB492, IEFBB414, IEFBB416, IEFDB4A1, IEFDB413, IEFDB418.

Exit to: Caller.

Called routines: IEEAB401, IEFAB4A2, IEFAB4A4, IEFAB4A6, IEFAB4A8, IEFAB4FC, IEFAB4F4.

IEFAB4A3 – Special Scratch/Catalog Processing

Function: This module builds a dummy ETIOT needed by the scratch function of data management, when no DSABs and TIOTs exist. This module also interfaces with IEFAB4F5 to allocate and open step catalogs, if requested, and with IEFAB4F4 to unallocate all private catalogs when the disposition processing has completed, if the allocation of step catalogs was requested.

Entry from: IEFAB4A2.

Exit to: Caller.

Called routines: IEFAB4FC, IEFAB428, IEFAB4F4, IEFAB4F5.

IEFAB4A4 – Unit Unallocation Control

Function: This module controls the processing associated with unallocating devices for each request. The functions performed are:

- Releasing units no longer needed.
- Rewinding tape volumes.
- Deleting outstanding mount messages.
- Identifying volumes to be unloaded.
- Posting waiting allocations.

Entry from: IEFAB4A0, IEFAB4DE.

Exit to: Caller.

Called routines: IEFAB4FA, IEFAB4F7, IEFAB493, IEFHB410.

IEFAB4A6 – Data Set Release

Function: This module releases (dequeues) data sets that are no longer needed by the job.

Entry from: IEFAB4A0, IEFBB401.

Exit to: Caller.

Called routines: IEFAB4F7.

IEFAB4A8 – Volume Release

Function: This module is responsible for releasing volumes that were reserved for the job step by allocation and data management (OPEN/CLOSE/EOV).

Entry from: IEFAB4A0.

Exit to: Caller.

Called routines: None.

IEFAB4B0 – Disposition Message Routine

Function: This module issues disposition messages to the system output data set and to the system operator.

Entry from: IEFAB4A2.

Exit to: Caller.

Called routines: IEFAB4FD.

IEFAB4B2 — Alternate Disposition Message Routine

Function: This module issues disposition messages to the system output data set and/or to the system operator when no storage is available for a volume list.

Entry from: IEFAB4A2.

Exit to: Caller.

Called routines: IEFAB4FD.

IEFAB4C0 — Unit Eligibility Service

Function: IEFAB4C0 determines if each unit in the input is eligible for allocation and sets the corresponding reason code in the reason code list appropriately.

Entry-Point IEFAB4C0: See FUNCTION.

Callers: SMS VTOC/Data Set Service

Exit-Normal: Return to caller.

External-References: IEFAB4EX.

IEFAB4C1 — Unit Allocation/Unallocation Interface Service

Function: IEFAB4C1 prepares the interface and calls IEFDB440 to process a caller's request to mark devices as allocated or to release devices that the caller has allocatedately.

Entry-Point IEFAB4C0: See FUNCTION

Callers: SMS VTOC/Data Set Service

Exit-Normal: Return to caller.

External-References IEFAB4EX and IEFDB440.

IEFAB4C2 — TIOT Services Routine

Function: IEFAB4C2 updates DSAB/TIOT entries. It provides three functions:

1. To replace a dummy unit entry in the TIOT with the unit passed in by the caller.
2. To count the number of non-dummy UCB entries in the TIOT associated with the input DSAB address.
3. To build a list of allocated UCB addresses.

Entry-Point IEFAB4C2: See FUNCTION:

Callers: SMS VTOC/Data Set Services

Exit-Normal: Return to caller.

External-References IEFAB4EX, IEFAB428, and IEFAB440.

IEFAB4C3 — Data Set Integrity Service

Function: IEFAB4C3 accepts a data set name as input and a flag to indicate the type of serialization required and then calls the allocation reserve/release module, IEFGB4DC.

Entry-Point IEFAB4C3: See FUNCTION.

Callers: SMS VTOC/Data Set Service

Exit-Normal: Return to caller.

External-References IEFGB4DC and IEFAB4EX.

IEFAB4C4 — AVR Service Routine

Function: IEFAB4C4 is used as an interface to the allocation AVR module. The function of AVR is to perform a volume read, if possible, on a designated input device and to store the obtained volume serial number into the UCB.

Entry-Point IEFAB4C4: See FUNCTION.

Callers: IEE3139D.

Exit-Normal: Return to caller.

Exit-Error: Return to error.

External-References: IEFAB4EX, IEFAB428, and IEFAB440.

IEFAB4DC — Data Set Reservation/Release

Function: This module can perform the following functions:

- Updates the data set enqueue table to add or change an entry.
- Dequeues the dsname.
- Updates the data set enqueue table to delete an entry.

Entry from: IEFAB459, IEFDB411, IEFDB413, IEFDB418.

Exit to: Caller.

Called routines: IEFAB4F7.

IEFAB4DD – Job/Step Unallocation ESTAE Exit

Function: In the event of an abnormal termination, this module calls step and/or job unallocation, if needed, and if SWA records have been read successfully. This module also requests a dump, if necessary.

Entry from: RTM (recovery termination management).

Exit to: Caller.

Called routines: IEEAB400, IEEAB401, IEFBB414, IEFBB416.

IEFAB4DE – Common Unallocation ESTAE Exit Routine

Function: If an abnormal termination occurs in a module that had requested a retry, IEFAB4DE puts the address of the allocation ESTAE retry block (AERB) into the SDWACOMU field of the SDWA and returns to the caller. Otherwise, this module attempts to perform any common unallocation functions that had been requested but not yet attempted at the time of an abnormal termination.

Entry from: IEFAB4ED.

Exit to: Caller.

Called routines: IEFAB4A0, IEFAB4A4, IEFAB4FA.

IEFAB4EA – Housekeeping ESTAE Exit Routine

Function: This module reestablishes control block pointers and frees resources for JFCB housekeeping control (IEFAB451), allocate catalog control (IEFAB4F5), or unallocate catalog control (IEFAB4F4), if an ESTAE is scheduled in any of those routines as the result of an ABEND.

Entry from: IEFAB4ED.

Exit to: Caller.

Called routines: IEFAB4F4.

IEFAB4EB – PDI Read and Chain

Function: This module reads all the PDIBs (passed data set information blocks) that comprise the PDI (passed data set information) and chains the PDIBs together.

Entry from: IEFAB455, IEFAB459, IEFBB416.

Exit to: Caller.

Called routines: IEFAB4F7.

IEFAB4EC – Group Lock/Unlock Interface

Function: This module obtains locks for the groups represented by the list of UCBs passed to it. It performs this serialization via the allocation queue manager.

Entry from: IEFBB416.

Exit to: Caller.

Called routines: IEFAB4FA, IEFAB4UV.

IEFAB4ED – Allocation Common ESTAE Exit

Function: This module is the common ESTAE exit for all of allocation. It performs general recovery processing and then routes control to other exit routines for more specific recovery. If a specific recovery routine requests a retry, this module returns to RTM requesting a retry. Otherwise, after all of the recovery routines have completed processing, this module returns to RTM requesting percolation.

Entry from: RTM (recovery termination management).

Exit to: Caller.

Called routines: IEEAB401, IEFAB4DD, IEFAB4DE, IEFAB4EA, IEFAB4E2, IEFAB4E4, IEFAB4E7, IEFAB4E8, IEFDB402.

IEFAB4EE – Allocation Message Routine

Function: This module builds the allocation messages for the requests on the input SIOT chain.

Entry from: IEFAB490, IEFBB414.

Exit to: Caller.

Called routines: IEFAB4FD, IEFAB4F3.

IEFAB4EF – PCCB Routine

Function: This module handles all requests for the creation, modification, and deletion of PCCB(s) and their chain structure.

Entry from: IEFAB4F4, IEFAB4F5, IEFAB452.

Exit to: Caller.

Called routines: None.

IEFAB4EG – Allocation Resource Manager ESTAE Exit Routine

Function: If the allocation resource manager (IEFAB4F5) abnormally terminates and had requested a retry, IEFAB4EG puts the address of the allocation ESTAE retry block (AERB) into the SDWACOMU field of the SDWA.

Entry from: IEFAB4ED.

Exit to: Caller.

Called routines: None.

IEFAB4E0 – Test If Device Is Ready

Function: This module tests to determine whether a device is physically ready by issuing a NOP EXCP to the device.

Entry from: IEFAB473, IEFAB49C, IEFAB494, IEFAB495.

Exit to: Caller.

Called routines: None.

IEFAB4E1 – VM&V FRR Exit Routine

Function: This module receives control if an ABEND occurred while an MVCA was being unchained. It tries again to take the MVCA off the chain. If successful, it sets a bit to indicate that it should free MVCA storage.

Entry from: RTM (recovery termination management).

Exit to: Caller.

Called routines: None.

IEFAB4E2 – VM&V ESTAE Exit Routine

Function: If an abnormal termination occurs in a module that had requested a retry, IEFAB4E2 puts the address of the allocation ESTAE retry block (AERB) into the SDWACOMU field of the SDWA and returns to the caller. Otherwise, this module releases mount control blocks that still exist for this allocation.

Entry from: IEFAB4ED.

Exit to: Caller.

Called routines: IEFAB49A, IEFAB498.

IEFAB4E3 – Job/Step Allocation Retry Routine

Function: This retry routine receives the initiator's registers on entry. It sets a return code of 4 and branches on register 14, thereby simulating an error return to the initiator.

Entry from: RTM (recovery termination management).

Exit to: IEFSD162.

Called routines: None.

IEFAB4E4 – Job/Step Allocation ESTAE Exit

Function: This module sets bits in the LCT to indicate that an error has occurred. It then prepares for retry by storing the initiator's registers in the SDWA. These registers will be received by retry routine IEFAB4E3.

Entry from: IEFAB4ED.

Exit to: Caller.

Called routines: IEEAB400, IEEAB401, IEFAB4F7.

IEFAB4E5 – Allocation Resource Manager

Function: This module attempts to clean up system resources owned by allocation in the event of an abnormal memory termination. It does the following:

- Releases device groups held by the terminating address space.
- Frees cross memory communication areas.
- Posts other allocations waiting for devices.
- Unallocates UCBs allocated to this address space.

Entry from: RTM (recovery termination management).

Exit to: Caller.

Called routines: IEFAB4FA, IEFAB498, IEFHB410.

IEFAB4E6 – Allocation Recovery Routine

Function: This module handles abnormal terminations that occur when the allocation address space is being initialized or updated or when the UCB field UCBUSER is being updated. It also checks the counts in the UCB and DALT to ensure that they are correct.

Entry from: RTM.

Exit to: Caller.

Called routines: IEFHB410 by means of a program call (PC).

IEFAB4E7 – Group Lock/Unlock ESTAE Exit

Function: This module frees resources held by LOCK/UNLOCK processing (IEFAB4EC) if an ABEND occurs while IEFAB4EC's ESTAE is in effect.

Entry from: IEFAB4ED.

Exit to: Caller.

Called routines: IEFAB4FA.

IEFAB4E8 – Common Allocation ESTAE Exit Routine

Function: If an abnormal termination occurs in a module that had requested a retry, IEFAB4E8 puts the address of the allocation ESTAE retry block (AERB) into the SDWACOMU field of the SDWA and returns to the caller. Otherwise, this module performs cleanup processing if an abnormal termination has occurred during common allocation processing.

Entry from: IEFAB4ED.

Exit to: Caller.

Called routines: IEEAB401, IEFAB4FA, IEFAB49A, IEFAB498.

IEFAB4E9 – Dynamic Allocation Set DSABDCBM Mask Bits Subroutine

Function: For each filled-in DCB field in the JFCB, this module sets on the corresponding bit in DSABDCBM.

Entry from: IEFAB428, IEFDB411.

Exit to: Caller.

Called routines: None.

IEFAB4FA – Allocation Queue Manager

Function: This module serializes access to UCB (device) groups by device allocation and manages the WAIT-FOR-DEVICE queue.

Entry from: IEFAB4A4, IEFAB4DE, IEFAB4E5, IEFAB4E8, IEFAB471, IEFAB472, IEFAB485, IEFAB486, IEFAB490, IEFAB491, IEFAB492.

Exit to: Caller.

Called routines: IEA0PT01 (system post).

IEFAB4FC – TIOT Manager Control Routine

Function: This module is responsible for obtaining and maintaining the TIOT and the DSAB queue descriptor block (QDB). The functions of this routine are:

- Create the TIOT and DSAB QDB.
- Initialize TIOT and DSAB QDB.
- Allocate a TIOT DD entry.
- Release a TIOT DD entry.
- Update the TIOT and DSAB QDB.
- Unallocate the DSAB/TIOT(ETIOT) entry.
- Concatenate the DSAB/TIOT DD entry.
- Free the TIOT and QDB.

Entry from: IEESB601, IEFAB4A0, IEFAB4A3, IEFAB428, IEFBB401, IEFBB404, IEFBB410, IEFDB413, IEFDB450.

Exit to: Caller.

Called routines: None.

IEFAB4FD – System Message Interface Routine

Function: This module converts the input parameter list into a WTO parameter list, which it passes to IEEAB400. IEEAB400 buffers any WTO messages (routing code 11) and issues the WTO macro instruction if any other routing codes are specified.

Entry from: IEFAB4B0, IEFAB4B2, IEFAB4EE, IEFAB490, IEFBB401, IEFBB402, IEFBB404, IEFBB410.

Exit to: Caller.

Called routines: IEEAB400.

IEFAB4FE – SWA Reader Routine

Function: This module establishes addressability to SIOTs and their related JFCBs and JFCBXs.

Entry from: IEFBB401, IEFBB410, IEFDB400.

Exit to: Caller.

Called routines: IEFAB4F7.

IEFAB4F0 – Generalized Conditional ENQ/DEQ Routine

Function: This module does the following:

- Enqueues conditionally (either shared or exclusive) on the input volume serial number.
- Dequeues conditionally on the input volume serial number.

Entry from: IEFAB436, IEFAB497, IEFAB492.

Exit to: Caller.

Called routines: None.

IEFAB4F1 – Test Volume ENQ

Function: This module determines if the input volume serial number can be shared by this allocation (even though it is already being used by this job step through a prior entry to allocation).

Entry from: IEFAB421.

Exit to: Caller.

Called routines: None.

IEFAB4F2 – Update Algorithm Tables

Function: This module helps maintain the cover/reduce algorithm interface tables. Its functions are:

- Change a request to device-required.
- Indicate a request was just allocated.

Entry from: IEFAB432, IEFAB434, IEFAB442.

Exit to: Caller.

Called routines: IEFAB481.

IEFAB4F3 – Message Compression Routine

Function: This module eliminates consecutive and trailing blanks from the text received in the input buffer. The compressed text and the new text length are returned to the caller.

Entry from: IEFAB4EE, IEFAB421, IEFAB48A, IEFAB487, IEFAB488, IEFAB490, IEFBB401, IEFBB404, IEFBB410.

Exit to: Caller.

Called routines: None.

IEFAB4F4 – Catalog Unallocation Control

Function: This module calls other routines to perform the following functions, as requested by the caller:

- Close private catalogs.
- Unallocate private catalogs.
- Release private catalog control blocks (PCCBs).

Entry from: IEFAB4A0, IEFAB4EA, IEFAB4F5, IEFAB451.

Exit to: Caller.

Called routines: IDACAT12, IEFAB4EF, IEFDB400 (via SVC99).

IEFAB4F5 – Allocate Catalog Control

Function: This module can:

- Allocate a catalog.
- Cause an ACB to be created and opened.
- Cause a PCCB to be created and chained or modified.

Entry from: IEFAB469, IEFICATL (an initiator module), IGG0CLCA (a catalog management module), IGG0CLC9 (a catalog management module).

Exit to: Caller.

Called routines: IDACAT11, IEFAB4EA, IEFAB4EF, IEFAB4F4.

IEFAB4F6 – GSPACE/FSPACE Service Routines

Function: This module obtains storage and subdivides them as storage requests are made. Requests for release of the 4K blocks and the obtained sections of them are also serviced.

Entry from: Most allocation/unallocation modules, that is, modules with names beginning with IEFAB, IEFBB, or IEFDB.

Exit to: Caller.

Called routines: None.

IEFAB4F7 – SWA Manager Interface

Function: This module invokes the SWA Manager to assign, read, write, or delete blocks of the Scheduler Work Area and returns the necessary parameters to the caller.

Entry from: IEFAB4A4, IEFAB4A6, IEFAB4DC, IEFAB4EB, IEFAB4FE, IEFAB4SF, IEFAB427, IEFAB452, IEFAB453, IEFAB455, IEFAB457, IEFAB458, IEFAB459, IEFAB461, IEFAB464, IEFAB466, IEFBB401, IEFBB402, IEFBB410, IEFBB416, IEFDB4A1, IEFDB400, IEFDB410, IEFDB413, IEFDB414, IEFDB418, IEFDB450.

Exit to: Caller.

Called routines: None.

IEFAB4F8 – Direct Access Label Read

Function: This module reads direct access volume labels.

Entry from: IEFAB473, IEFAB49B.

Exit to: Caller.

Called routines: None.

IEFAB4F9 – Tape Label Read

Function: This module reads tape volume labels.

Entry from: IEFAB473.

Exit to: Caller.

Called routines: None.

IEFAB4FG – SJF Service Routine

Function:

IEFAB4FG provides an interface between Allocation and the SJF Access Function. This module could also be invoked for the purpose of only building SJF tokens.

Entry-Point IEFAB4FG: See FUNCTION.

Callers IEFAB42A, IEFAB431, IEFAB343, IEFAB457, IEFAB464, IEFAB469, IEFAB490, IEFAB492, IEFDB4A1, IEFDB410, IEFDB414, IEFDB418, and IEFDB470.

Exit-Normal: Return to caller.

External-References: SJF Access Function.

IEFAB4GB – Modify Device Online/Offline Status

Function: This module designates the device status in the UCB as either offline or pending offline, depending on the input function code.

Entry from: IEFAB429, IECVRDIO, IECVRSTI, IECVFDEV, IECVRRSV.

Exit to: Caller.

Called routines: None.

IEFAB4M4 – Volume Mount & Verify Messages

Function: This module contains the text used to construct the messages issued by volume mount and verify routines. This module contains no executable code.

IEFAB4M5 – WTO Message Module For AVR and Common Allocation

Function: This module contains the list forms of WTO and WTOR messages issued by common allocation, automatic volume recognition (AVR), and data set reservation/release.

Entry from: Not applicable (non-executable module).

Exit to: Not applicable (non-executable module).

Called routines: Not applicable (non-executable module).

IEFAB4M6 – Disposition Messages

Function: This module contains the text used to construct the disposition messages issued by common unallocation. This module contains no executable code.

IEFAB4M7 – Allocation Messages

Function: This module contains the allocation messages that are issued to the system output data set and, for unit record devices, to the console. This module contains no executable code.

IEFAB4M9 – Recovery Allocation Messages

Function: This module contains the text used to construct recovery allocation operator messages. This module contains no executable code.

IEFAB4SF – Spool File Allocation

Function: This module interfaces with the job entry subsystem to allocate or unallocate SYSIN and SYSOUT data sets. It is also called to segment a SYSOUT data set.

Entry from: IEFAB4A2, IEFAB427, SETPRT SVC.

Called routines: IEFAB4F7, subsystems.

IEFAB4UV – Unit Verification

Function: This routine is invoked by system routines that require information or validity checks based on the contents of the eligible devices table (EDT).

Entry from: IAPINMD, ICBME.

Exit to: Caller.

Called routines: None.

IEFAB42A – SMS Allocation Processor

Function: This service updates DSAB/TIOT entries. It provides three functions: invokes the SMS VTOC data set services routine to perform many of the traditional allocation functions for SMS merged requests.

Entry-Point IEFAB42A: See function.

Callers: IEFAB421.

Exit-Normal: Return to caller.

Exit-Error: Return to caller.

External-References:

- IEFAB428, IEFAB4FG, and IEFAB4F7.

IEFAB421 – Common Allocation Control

Function: This module is called to allocate one or more data set requests. It is responsible for controlling the following functions:

- Process pending offlines/unloads.
- Initialize work area (ALCWA).
- Allocate DUMMY requests.
- Allocate VIO requests.
- Allocate subsystem requests.
- Determine device requirements.
- Allocate to fixed DA devices.
- Allocate TP requests.
- Volume reservation.
- Allocate by device type.
- Recovery allocation.
- Dequeue from allocation resource.
- Cleanup processing.

Entry from: IEFAB490, IEFBB404, IEFDB413.

Exit to: Caller.

Called routines: IEEAB401, IEFAB4F3, IEFAB422, IEFAB423, IEFAB425, IEFAB427, IEFAB428, IEFAB429, IEFAB430, IEFAB431, IEFAB471, IEFAB485, IEFAB490, IEFAB491.

IEFAB422 – JES3 Interface

Function: This module interfaces with the JES3 subsystem and then updates allocation tables based on whether or not JES3 has made device selections.

Entry from: IEFAB421.

Exit to: Caller.

Called routines: None.

IEFAB423 – Determine Device Requirements

Function: The function of this module is to determine device eligibility for each individual request and for the step. Device requirements are represented by the volunit table and device eligibility is represented by the eligible device table (EDT) and eligible device list (EDL). This module builds the volunit table and calls IEFAB424 to build the EDL.

Entry from: IEFAB421.

Exit to: Caller.

Called routines: IEFAB424, IEFAB426.

IEFAB424 – Build Eligible Device List (EDL)

Function: This module is responsible for construction of an eligible device list (EDL) for each request.

Entry from: IEFAB423.

Exit to: Caller.

Called routines: IEFAB438.

IEFAB425 – Process TP Requests

Function: This module controls the processing necessary to allocate teleprocessing requests.

Entry from: IEFAB421.

Exit to: Caller.

Called routines: IEFAB434.

IEFAB426 – Volunit Affinity Processing

Function: This module updates volunit table entries to reflect volume affinity. It ensures that every member of a volume affinity has the same unit identification.

Entry from: IEFAB423.

Exit to: Caller.

Called routines: None.

IEFAB427 – Allocate Subsystem Data Sets

Function: This module controls allocation of subsystem data sets. The major functions performed are:

- Find subsystem requests.
- Sort subsystem requests.
- Issue the IEFSSREQ macro instruction.
- Create DSAB/TIOT entries.

Entry from: IEFAB421.

Exit to: Caller.

Called routines: IEFAB4F7, IEFAB4SF, Subsystems, IEFAB428.

IEFAB4C0 – Unit Eligibility Service

Function: Determines if each unit in the input is eligible for allocation and sets the corresponding reason code in the reason code list.

Entry-Point IEFAB4C0: See function.

Callers: SMS VTOC data set service.

Exit-Normal: Return to caller.

External-References: IEFAB4EX.

IEFAB4C1 – Unit Allocation/Unallocation Interface Service

Function: Prepares the interface and calls IEFDB440 to process a caller's request to mark devices as allocated or to release devices that the caller has allocated.

Entry-Point IEFAB4C1: See function.

Callers: SMS VTOC data set service.

Exit-Normal: Return to caller.

External-References: IEFAB4EX and IEFDB440.

IEFAB4C2 – TIOT Services Routine

Function: This service updates DSAB/TIOT entries. It provides three functions:

1. To replace a dummy unit entry in the TIOT with the unit passed in by the caller.
2. To count the number of non-dummy UCB entries.
3. To build a list of allocated UCB addresses.

Entry-Point IEFAB4C2: See function.

Callers: SMS VTOC data set service.

Exit-Normal: Return to caller.

External-References: IEFAB4EX, IEFDB440, and IEFAB428.

IEFAB4C3 – Data Set Integrity Service

Function: This service accepts a data set name as input and a flag to indicate the type of serialization required and then calls the allocation reserve/release module.

Entry-Point IEFAB4C3: See function.

Callers: SMS VTOC data set service.

Exit-Normal: Return to caller.

External-References: IEFGB4DC and IEFAB4EX.

IEFAB4C4 – AVR Service Routine

Function: This module is used as an interface to the allocation AVR module. AVR performs a volume read, if possible, on a designated input device and to store the obtained volume serial number into the UCB.

Entry-Point IEFAB4C4: See function.

Callers: SMS VTOC data set service.

Exit-Normal: Return to caller.

External-References: IEFAB473.

IEFAB4FG – SJF Interface Routine

Function: This module provides an interface between allocation and the SJF access function. IEFAB4FG can also be invoked to build SJF token(s).

Entry-Point IEFAB4FG: See function.

Callers: IEFAB42A, IEFAB431, IEFAB434, IEFAB457, IEFAB464, IEFAB469, IEFAB490, IEFAB492, IEFAB4A1, IEFAB410, IEFAB414, IEFAB418, and IEFAB470.

Exit-Normal: Return to caller.

External-References: SJF access function.

IEFAB42A – SMS Allocation Processor

Function:

IEFAB42A processes all allocation requests which are SMS managed. This module invokes the SMS VTOC/Data Set Services routine to perform many of the traditional allocation functions for SMS managed requests.

Entry-Point IEFAB42A: See FUNCTION.

Callers: IEFAB421.

Exit-Normal: Return to caller.

Exit-Error: Return to caller.

External-References: IEFAB428, IEFAB4FG, IEFAB4F7.

IEFAB428 – Build/Update/Rebuild Expandable TIOT

Function: This module creates, updates, or rebuilds the expandable TIOT (DSAB/TIOT DD entry).

Entry from: IEFAB4A3, IEFAB421, IEFAB427, IEFAB431, IEFAB434, IEFAB442.

Exit to: Caller.

Called routines: IEFAB4E9, IEFAB4FC.

IEFAB429 – Process Pending Offlines/Unloads

Function: This module performs offline/unload processing on units or consoles, after the device is unallocated.

Entry from: IEFAB421.

Exit to: IEFAB421.

Called routines: IEFAB4GB, IEFAB493, IEFENFFX.

IEFAB430 – Fixed Device (Main Path) Control

Function: This module controls the processing necessary to allocate requests to direct access volumes that are permanently resident or reserved.

Entry from: IEFAB421.

Exit to: Caller.

Called routines: IEFAB433, IEFAB436.

IEFAB431 – Allocate VIO Data Sets

Function: This module controls the allocation of VIO data set requests.

Entry from: IEFAB421.

Exit to: Caller.

Called routines: IEFAB428.

IEFAB432 – Affinity Processor

Function: This module processes any affinity requests to the volume or unit just allocated. Processing includes ensuring that the affinity requests are valid and, possibly, allocating the requests.

Entry from: IEFAB434.

Exit to: Caller.

Called routines: IEFAB4F2, IEFAB434, IEFAB441.

IEFAB433 – Specific Volume Allocation Control

Function: This module controls the processing necessary to allocate specific volume requests when the volume is mounted. A specific request is one that requires a particular volume — for example, a volume serial number was specified, the data set was passed, the data set is cataloged.

Entry from: IEFAB430, IEFAB475, IEFAB479, IEFAB485.

Exit to: Caller.

Called routines: IEFAB434, IEFAB442.

IEFAB434 – Allocate Request to Unit

Function: This module controls the processing necessary to allocate a request to the selected device. Functions include:

- Updating the UCB of the selected device (done by IEFAB435).
- Creating or updating a DSAB and TIOT entry (done by IEFAB428).
- Acquiring space on the volume (interfacing with DADSM).
- Unallocating a unit and decreasing the DALT use count (done by IEFHB410), if a DADSM allocate error occurs.
- Processing requests that specify affinity to the request being handled (done by IEFAB432).

Entry from: IEFAB425, IEFAB432, IEFAB433, IEFAB436, IEFAB441, IEFAB478, IEFAB479, IEFAB489.

Exit to: Caller.

Called routines: IEFAB4F2, IEFAB428, IEFAB432, IEFAB435, IEFHB410.

IEFAB435 – Update UCB Routine

Function: This module updates the UCB to reflect allocation of the current request to it. It also interfaces with VM&V control to unload volumes mounted on the needed unit.

Entry from: IEFAB434.

Exit to: Caller.

Called routines: IEFAB4E0, IEFAB441, IEFAB493, IEFHB410.

IEFAB436 – Nonspecific Volume Allocation Control

Function: This module controls the processing necessary to allocate nonspecific, nonprivate requests to public and storage volumes. It is responsible for building a list of devices that can satisfy the request, interfacing with the system resources manager to select the device to be allocated, and interfacing with IEFAB434 to allocate the request.

Entry from: IEFAB430, IEFAB475, IEFAB485.

Exit to: Caller.

Called routines: IEFAB4F0, IEFAB434, IEFAB440, IEFAB442.

IEFAB438 – Update DDR Count Routine

Function: This module retrieves the DDR count from the CSD. The count is used by module IEFAB424 to ensure that allocation is synchronized with DDR while the eligible device list (EDL) is being built.

Entry from: IEFAB424.

Exit to: Caller.

Called routines: None.

IEFAB440 – Build Allocated UCB List

Function: This module builds a list of allocated UCBs (ALCUCBLT) which is used in the interface with the system resources manager.

Entry from: IEFAB436, IEFAB478, IEFAB489, IEFAB490.

Exit to: Caller.

Called routines: None.

IEFAB441 – Volume Validity Checker

Function: This module determines if the input volume is currently mounted. If it is, this module checks to see if the volume can be used by the current request.

Entry from: IEFAB432, IEFAB435, IEFAB486, IEFGB4UV.

Exit to: Caller.

Called routines: IEFAB434, IEFAB442, IEFAB493.

IEFAB442 – Affinity Remover

Function: This module cancels unit affinity between the input request and other requests, when the volume being allocated to the input request is permanently resident or reserved.

Entry from: IEFAB433, IEFAB436, IEFAB441, IEFAB479.

Exit to: Caller.

Called routines: IEFAB4F2, IEFAB428.

IEFAB445 – Device Allocation Defaults Module

Function: This module contains space and unit name defaults used by device allocation. This module contains no executable code.

IEFAB451 – JFCB Housekeeping Control

Function: This module is responsible for retrieving the information necessary for allocation.

Entry from: IEFBB404, IEFDB413.

Exit to: Caller.

Called routines: IEFAB4EA, IEFAB4F4, IEFAB452.

IEFAB452 – DD Processing Control

Function: This module selects and initiates the processing of:

- STEPCAT requests.
- Other data set requests.

Entry from: IEFAB451.

Exit to: Caller.

Called routines: IEFAB4EF, IEFAB4F7, IEFAB453, IEFAB454.

IEFAB453 – DD Preparation

Function: This module does the following processing:

- Initializes the pointer to the JFCB for the requests.
- Processes data set requests that specified QNAME, TERM=TS, DUMMY, DSN=NULLFILE, or PGM=*, stepname.ddname or PGM=*, procstepname.ddname.
- Processes subsystem and VIO data sets.
- Retrieves unit information.

Entry from: IEFAB452.

Exit to: Caller.

Called routines: IEFAB4F7, IEFAB470.

IEFAB454 – DD Function Control

Function: This module determines what additional information is required for a DD. It then constructs a parameter list and routes control to the appropriate routine(s) to obtain that information.

Entry from: IEFAB452.

Exit to: Caller.

Called routines: IEFAB456, IEFAB457, IEFAB458, IEFAB459, IEFAB469.

IEFAB455 – Passed Data Set Information Scan

Function: This module scans the PDI (passed data set information) for the data set indicated. If the data set is found, it returns a pointer (SVAB) to the SIOT that passed the data set.

Entry from: IEFAB469.

Exit to: Caller.

Called routines: IEFAB4EB, IEFAB4F7.

IEFAB456 – Data Set Name Resolution

Function: This module resolves data set name information and provides volume and unit information for GDG requests.

Entry from: IEFAB454.

Exit to: Caller.

Called routines: IEFAB461, IEFAB466, IEFAB469.

IEFAB457 – Volume/Unit Resolution

Function: This module receives control for every request in order to resolve volume and unit information for subsequent allocation processing.

Entry from: IEFAB454.

Exit to: Caller.

Called routines: IEFAB4F7, IEFAB463, IEFAB464, IEFAB466, IEFAB469.

IEFAB458 – DCB Resolution

Function: This module establishes DCB control information in a JFCB.

Entry from: IEFAB454.

Exit to: Caller.

Called routines: IEFAB4F7, IEFAB469.

IEFAB459 – Disposition Processing

Function: This module does the following:

- Completes data set disposition information in the SIOT.
- Updates the PDI (passed data set information), if necessary.
- Marks tape data set requests as private, if necessary.
- Retrieves catalog information, if necessary.

Entry from: IEFAB454.

Exit to: Caller.

Called routines: IEFGB4DC, IEFAB4EB, IEFAB4FF, IEFAB469.

IEFAB461 – GDG Single Processing

Function: This module converts a GDG (generation data group) data set request with a relative data set name (generation number) to a fully qualified data set name. It obtains volume and unit information for a GDG single request of an existing data set.

Entry from: IEFAB456.

Exit to: Caller.

Called routines: IEFAB4F7, IEFAB469.

IEFAB463 – Multiple Device Type Determination

Function: This module scans the catalog return information (CRI) to:

- Determine if the data set is a multi-device type data set and, if so, to return the number of different device types minus one.
- Given the number of device types, find and return the respective device type and a pointer to the volume list entry for that device type.

Entry from: IEFAB457, IEFAB464.

Exit to: Caller.

Called routines: None.

IEFAB464 – Volume Unit Table Completion

Function: This module copies appropriate volume and unit information from the source determined by previous routines.

Entry from: IEFAB457.

Exit to: Caller.

Called routines: IEFAB4F7, IEFAB463, IEFAB470.

IEFAB466 – Table Creation

Function: The module creates SIOTs/JFCBs and initializes them.

Entry from: IEFAB456, IEFAB457, IEFAB469.

Exit to: Caller.

Called routines: IEFAB4FD, IEFAB4F7.

IEFAB469 – JLOCATE

Function: JLOCATE is a housekeeping service routine that provides data set name information for GDG single requests and necessary volume and unit information for a given data set name.

Entry from: IEFAB454, IEFAB456, IEFAB457, IEFAB458, IEFAB459, IEFAB461.

Exit to: Caller.

Called routines: IEFAB4EF, IEFAB4F5, IEFAB466.

IEFAB470 – Unit Name Conversion

Function: This module extracts necessary EDT or UCB device information from the current SIOT.

Entry from: IEFAB453, IEFAB464.

Exit to: Caller.

Called routines: None.

IEFAB471 – Generic Allocation Control

Function: This module controls the allocation of requests that could not be handled by fixed device allocation. (See IEFAB430.) It handles requests to any mounted volumes and/or to any online and unallocated units. It processes one generic device type at a time. The order of processing is determined by the installation device precedence list.

Entry from: IEFAB421.

Exit to: Caller.

Called routines: IEFAB4FA, IEFAB472, IEFAB473, IEFAB475.

IEFAB472 – Build Tables for Generic Processing

Function: This module builds the tables needed for generic allocation processing. The tables built are:

- Allocation queue manager request block.
- Cover/reduce algorithm interface tables.
- Request identification mask table.
- Workarea for manipulating group masks.

Entry from: IEFAB471.

Exit to: Caller.

Called routines: IEFAB4FA, IEFAB481.

IEFAB473 – AVR (Automatic Volume Recognition) Control

Function: This module reads the volume label of any volumes mounted on units whose UCBs are passed as input.

Entry from: IEFAB471.

Exit to: Caller.

Called routines: IEFAB4E0, IEFAB4F8, IEFAB4F9, IEFAB493, IEFXVNSL.

IEFAB474 – Process Multi-Unit/Multi-Generic Requests

Function: This module determines if a multi-unit (multi-generic) request processed by the algorithm is assigned to a single generic device type. If not, this module attempts to find a single generic device type that can satisfy the request.

Entry from: IEFAB476, IEFAB486.

Exit to: Caller.

Called routines: IEFAB481.

IEFAB475 – Dynamic Allocation Message Control

Function: This module processes all allocation owned messages as requested by the SVC 99 caller in the SVC 99 request block extension.

Entry-Point IEFAB475: See function.

Callers: IEFDB400.

Exit-Normal: Return to caller.

External-References: IEFDB476 and IEFDB477.

IEFAB476 – Dynamic Allocation Error Message Processor

Function: This module extracts allocation owned messages from the dynamic allocation error message module and outputs messages as requested by the SVC 99 caller.

Entry-Point IEFAB476: See function.

Callers: IEFDB475 and user programs requiring message services.

Exit-Normal: Return to caller.

External-References: IEFDB478.

IEFAB477 – Build User Message Block Routine

Function: This routine builds the message block(s) which are returned to the user when 'message return' is specified in the SVC 99 request block. SVC 99 caller.

Entry-Point IEFAB477: See function.

Callers: IEFDB475.

Exit-Normal: Return to caller.

External-References: None.

IEFAB478 – Common Dynamic Allocation (SVC 99) Failure

Function: This module diagnosis any dynamic allocation (SVC 99) failure and extracts an appropriate message.

Entry-Point IEFAB478: See function.

Callers: None.

Exit-Normal: Return to caller.

External-References: None.

IEFAB479 – Demand Allocation

Function: This module allocates requests that specify a specific unit address (for example, unit = 190) within the generic device type being processed by generic allocation.

Entry from: IEFAB475, IEFAB485, IEFAB491.

Exit to: Caller.

Called routines: IEFAB4F0, IEFAB433, IEFAB434, IEFAB442.

IEFAB48A – Process Offlines/Allocated Units

Function: This module controls the processing necessary:

- To determine whether a request can be satisfied by offline and/or allocated units.
- To inform the operator of the possible options.

Entry from: IEFAB487.

Exit to: Caller.

Called routines: IEFAB4F3, IEFAB488, IEFAB489.

IEFAB480 – Cover/Reduce Algorithm

Function: This module uses the hungarian algorithm to determine which group or groups of devices should be used to satisfy a generic allocation request.

Entry from: IEFAB476, IEFAB486, IEFAB488, IEFAB489, IEFAB491.

Exit to: Caller.

Called routines: None.

IEFAB481 – Eliminate Ineligible Groups and Generics

Function: This module marks all generic device types, except the one selected for the current request, as ineligible in the EDL. It also marks group list entries in the algorithm tables as ineligible, when the groups are not in the selected generic device type.

Entry from: IEFAB4F2, IEFAB472, IEFAB485, IEFAB486.

Exit to: Caller.

Called routines: None.

IEFAB485 – Recovery Allocation

Function: This module handles allocation requests that could not be allocated by either Fixed Device Control or Generic Allocation. Off-line and/or allocated units can be considered during these allocations, if necessary and if allowed by the caller.

Entry from: IEFAB421.

Exit to: Caller.

Called routines: IEFAB4FA, IEFAB433, IEFAB436, IEFAB476, IEFAB477, IEFAB479, IEFAB481, IEFAB486, IEFAB49C.

IEFAB486 – Offline/Allocated Device Allocation

Function: This module handles allocation processing when it is necessary to consider off-line and/or allocated devices.

Entry from: IEFAB485.

Exit to: Caller.

Called routines: IEFAB4FA, IEFAB441, IEFAB474, IEFAB477, IEFAB478, IEFAB480, IEFAB481, IEFAB487, IEFAB493.

IEFAB487 – Allocation Recovery Interface With Operator

Function: This module controls the processing necessary to inform the system operator that this allocation cannot proceed without waiting for devices to be unallocated by other jobs and/or bringing needed devices online.

Entry from: IEFAB486.

Exit to: Caller.

Called routines: IEFAB4F3, IEFAB48A, IEFAB488.

IEFAB488 – Recovery Reply Options Processor

Function: This module builds and issues the appropriate reply options message and processes the operator's reply.

Entry from: IEFAB48A, IEFAB487.

Exit to: Caller.

Called routines: IEFAB4F3, IEFAB480, IEFAB489, IEFENFFX.

IEFAB489 – Recovery Allocation of Online Devices

Function: This module determines, for each group of UCBs in the system, whether the number of online units in the group has increased as a result of the system operator's responses to the reply options message. If it has, the routine will attempt to allocate requests to the units in the group.

Entry from: IEFAB48A, IEFAB488.

Exit to: Caller.

Called routines: IEFAB434, IEFAB440, IEFAB480, ICBME.

IEFAB49A – VM&V DOMR & Cleanup Routine

Function: This module deletes mount messages, if necessary, and releases MVCAs used by this allocation when all mount messages have been deleted.

Entry from: IEFAB4E2, IEFAB4E8, IEFAB492, IEFAB496.

Exit to: Caller.

Called routines: IEFAB498.

IEFAB49B – Device End POST Handler

Function: This module verifies, if necessary, that:

- A device is now ready (that is, a volume has been mounted); or,
- The volume mounted on the device is acceptable in response to a mount message issued by mount control or a mount request to the 3850 mass storage system.

Entry from: IEFAB496.

Exit to: Caller.

Called routines: IEFAB4F3, IEFAB4F8, IEFAB493.

IEFAB49C – Non-Allocation Unload Interface

Function: This module interfaces with volume mount and verify (VM&V) control (IEFAB493) to have volumes unloaded and/or rewound for non-allocation callers. Before calling IEFAB493, this routine verifies the UCB pointers in each VM&V request block (RB).

Entry from: IATLVVR, IATSICA, IEEMB813, IEEVCHPR, IEEVPTHR.

Exit to: Caller.

Called routines: IEFAB493.

IEFAB490 – Common Allocation Cleanup

Function: This module performs cleanup functions when allocation has completed or when a terminating allocation error has been detected.

Entry from: IEFAB421.

Exit to: Caller.

Called routines: IEFAB4A0, IEFAB4EE, IEFAB4FA, IEFAB4FD, IEFAB4F3, IEFAB421, IEFAB492.

IEFAB491 – Wait Holding Resources

Function: This module waits for other job steps to unallocate devices in the device groups from which this allocation needs additional units. When posted because of an unallocation, this routine tries to satisfy the remaining requests.

Entry from: IEFAB421.

Exit to: Caller.

Called routines: IEFAB4FA, IEFAB478, IEFAB479, IEFAB480, IEFAB493.

IEFAB492 – Allocation/VM&V Interface

Function: This module interfaces with VM&V Control (IEFAB493) to process volume mount and verify requests (represented by VM&V request blocks build during generic and recovery allocation).

Entry from: IEFAB490.

Exit to: Caller.

Called routines: IEFAB4A0, IEFAB4FA, IEFAB4F0, IEFAB49A, IEFAB493, IEFAB498.

IEFAB493 – Volume Mount and Verify (VM&V) Control

Function: This module controls unload, mount, and verify processing; it scans VM&V request blocks and calls unload control (IEFAB494), mount control (IEFAB495), and verify control (IEFAB496) to process the requests.

Entry from: IEFAB4A4, IEFAB429, IEFAB435, IEFAB441, IEFAB473, IEFAB477, IEFAB49B, IEFAB49C, IEFAB491, IEFAB492, or IEFABB416.

Exit to: Caller.

Called routines: IEFAB494, IEFAB495, IEFAB496.

IEFAB494 – Volume Unload Control

Function: This module processes all requests to:

- Rewind or rewind and unload tape volumes.
- Unload direct access volumes.

Entry from: IEFAB493.

Exit to: Caller.

Called routines: IEFAB4E0, IEFAB4F3, IEFAB499, IEFENFFX.

IEFAB495 – Mount Control Routine

Function: This module builds mount control blocks. For mass storage system (MSS) volume requests, this module interfaces with the 3850 mass storage system to mount the volumes. For non-MSS volume requests, this module issues mount WTO or WTOR messages.

Entry from: IEFAB493.

Exit to: Caller.

Called routines: IEFAB4E0, IEFAB498, IEFAB499.

IEFAB496 – Verify Control Routine

Function: This module handles all verify device end and verify label requests by waiting for the specified units to become ready. In the case of verify label, this module also ensures that the requested volume has been mounted.

Entry from: IEFAB493.

Exit to: Caller.

Called routines: IEFAB49A, IEFAB49B, IEFAB498, IEFAB499.

IEFAB498 – MVCA Chain Processor

Function: This module is invoked to perform one of the following functions:

- Search the MVCA chain for the MVCA for this allocation.
- Add an MVCA to the MVCA chain.
- Delete an MVCA from the MVCA chain.

Entry from: IEFAB4E2, IEFAB4E5, IEFAB4E8, IEFAB49A, IEFAB492, IEFAB495, IEFAB496.

Exit to: Caller.

Called routines: None.

IEFAB499 – VM&V WTO/WTOR Format Routine

Function: This module builds WTO/R or multiline WTO/R parameter lists for MOUNT and UNLOAD and VERIFY messages using information supplied by the caller.

Entry from: IEFAB494, IEFAB495, IEFAB496.

Exit to: Caller.

Called routines: None.

IEFAB820 – TCTIOT Construction Interface

Function: This module determines whether SMF accounting functions are required, initializes the TCT storage map, and calls IEFIB660 to build the TCTIOT and the ETCTIOT.

Entry from: IEFSD263 via CALL (to IEFSMFAT, alias for IEFAB820).

Exit to: IEFSD263 via branch.

IEFAUINT – Assign/Unassign Initialization Service

Function: This module assigns one or all on-line assignable devices. It is invoked by JES2 during initialization to assign all assignable devices with the ASGNALL function. It is also invoked by JES3 during main device scheduling initialization using the ASGNALL function for devices that are not JES3-managed and the ASGNONE function for the individual assigning of JES3-managed devices.

Entry from: JES2 or JES3.

Exit to: Caller.

Called routines: IEFAUSRV, IOSVSUCB, IEFAB4GB.

IEFAUSRV – Hardware Assign/Unassign Service

Function: This module is invoked by IEFAUINT and other system components to assign a device, unassign a device, restore status of a device and build the CCWs to restore the assign status of a device.

Entry from: JES3, IEFAB488, IEFAB429, IECVIOPM, IEFAUINT, IOSVVARY, IOSSRRSV, IEECB809.

Exit to: Caller.

Called routines: None.

IEFBB4M1 – Job Status Messages

Function: This module contains the message text for the allocation job status operator messages. This module contains no executable code.

IEFBB4M2 – Allocation Step-Related Messages

Function: This module contains the message text of allocation step-related error messages that are issued to the system output data set. This module contains no executable code.

IEFBB4M3 – Allocation DD-Related Messages

Function: This module contains the message text of the allocation DD-related error messages that are issued to the system output data set. This module contains no executable code.

IEFBB4M4 — WTO Messages for IEFBB410

Function: This module contains the message text of the unallocation job status messages. This module contains no executable code.

IEFBB4M5 — Unallocation Messages

Function: This module contains the message text of the unallocation error messages. This module contains no executable code.

IEFBB401 — Initiator/Allocation Interface

Function: This module controls the interface between the initiator and step allocation control. (It determines the path that a step or job will take through the initiator.)

Entry from: IEFSD162.

Exit to: Caller.

Called routines: IEEAB401, IEFAB4A6, IEFAB4FC, IEFAB4FD, IEFAB4FE, IEFAB4F3, IEFAB4F7, IEFBB402, IEFBB404.

IEFBB402 — Test EXEC Statement Condition Codes

Function: This module determines, from the specifications of the COND parameter on the EXEC statement, whether the step should be bypassed or executed.

Entry from: IEFBB401.

Exit to: Caller.

Called routines: IEFAB4FD, IEFAB4F7.

IEFBB404 — Step Allocation Control

Function: This module receives control when a job step is to be allocated; it controls the following functions:

- JFCB housekeeping.
- Common allocation processing.
- Reordering and compressing the TIOT.
- Writing DD-related error messages.

Entry from: IEFBB401.

Exit to: Caller.

Called routines: IEFAB4FC, IEFAB4FD, IEFAB4F3, IEFAB421, IEFAB451.

IEFBB410 — Initiator/Unallocation Interface

Function: This module provides an interface between the initiator and step unallocation. SIOTs, JFCBs, and JFCBXs are located and chained, end-of-step messages are issued, and step unallocation is invoked.

Entry from: IEFSD164.

Exit to: Caller.

Called routines: IEEAB401, IEFAB4FC, IEFAB4FD, IEFAB4FE, IEFAB4F7, IEFAB412, IEFBB414, IEFBB416, IEFRPREP, IEFTB721.

IEFBB412 — Process Job Condition Codes

Function: This module compares the return code from the step just executed with the dependency codes specified in the COND parameter on the JOB statement.

Entry from: IEFBB410.

Exit to: Caller.

Called routines: None.

IEFBB414 — Step Unallocation Control

Function: This module creates the interface to Common Unallocation for the job step being terminated. This involves building a request block for each SIOT on the input SIOT chain and determining the processing required for each of these requests. Allocation messages are also built and issued if requested.

Entry from: IEFAB4DD, IEFBB410.

Exit to: Caller.

Called routines: IEFAB4A0, IEFAB4EE.

IEFBB416 — Job Unallocation

Function: This module performs disposition processing for passed unreceived data sets and unloads volumes no longer needed by this job.

Entry from: IEFAB4DD, IEFBB410.

Exit to: Caller.

Called routines: IEFAB4A0, IEFAB4EB, IEFAB4F3, IEFAB4F7, IEFAB493.

IEFCMAUT – JSS Common Authorization Check Routine

Function: Validates the userid and the groupid of a job when it is submitted, and if valid, propagates this information. Once validated, this information is not checked at job initiation.

Entry from: JES2, JES3

Exit Normal: Returns to the caller.

Exit Error: Issues a C0D abend when an unexpected return code is received from the RACROUTE macro.

IEFDB475 – Dynamic Allocation Message Control

Function: IEFDB475 Processes all Allocation owned messages as requested by the SVC 99 caller in the SVC 99 request block extension.

Entry-Point IEFDB475: See FUNCTION.

Callers: IEFDB400.

Exit-Normal: Return to caller.

Exit-Error: Return to caller.

External-References: IEFDB476 and IEFDB477.

IEFDB476 – Dynamic Allocation Error Message Processor

Function: IEFDB476 extracts Allocation owned messages from the Dynamic Allocation Error Message module and outputs messages as requested by the SVC 99 caller.

Entry-Point IEFDB476: See FUNCTION.

Callers: IEFDB475 and user programs requiring message services.

Exit-Normal: Return to caller.

Exit-Error: Return to caller.

External-References: IEFDB478.

IEFDB477 – Build User Message Block Routine

Function: IEFDB477 builds the message blocks to be returned to the user when 'message return' is specified in the SVC 99 request block.

Entry-Point IEFDB477: See FUNCTION:

Callers: IEFDB475.

Exit-Normal: Return to caller.

Exit-Error: return to caller.

IEFDB478 – Common Dynamic Allocation (SVC 99) Failure or DAIR Failure Message Extraction Routine

Function: IEFDB478 diagnoses any Dynamic Allocation (SVC 99) failure or DAIR failure and extracts an appropriate message.

Entry-Point IEFDB478: See FUNCTION.

Callers: None.

Exit-Normal: Return to caller.

Exit-Error: Return to caller.

External-References: TSMSGS (message csect within IEFDB478).

IEFDB4A0 – Dynamic Unallocation Control Routine

Function: IEFDB4A0 handles Dynamic Allocation (SVC99) requests for unallocation. It checks the validity of the request, determines what DSABs should be processed and what processing should be done for them (Unallocation or remove in-use), and invokes the necessary routine(s) to perform the processing.

Entry from: IEFDB400, IEFDB410.

Exit to: Caller.

Called routines: IEFDB4A1, IEFDB4FA, IEFDB4FC, IEFDB4FF, IEFDB460, IEFDB481.

IEFDB4A1 – Dynamic Unallocation Processor

Function: This module receives control from either Unallocation Control (IEFDB4A0), or Remove In-use Processor (IEFDB481) to perform Unallocation Processing for DSABs pointed to by the input list of DSABPTRs.

Entry from: IEFDB4A0, IEFDB481.

Exit to: Caller.

Called routines: IEFAB4A0, IEFGB4DC, IEFAB4F7, IEFDB4F9.

IEFDB4D0 – DAIR - Dynamic Allocation Interface Routine

Function: In a TSO environment, this module serves as an interface from various command processors to dynamic allocation.

Entry from: IKJDAIR.

Exit to: Caller of IKJDAIR.

Called routines: IEFDB400 (via SVC 99), IKJEHCIR (via LINK).

IEFDB4FA – Dsname Search Routine

Function: This module searches the DSABs and JFCBs for the data set name requested by the user. If the dsname is allocated to the current task, the address of its DSAB is returned to the caller.

Entry from: IEFDB4A0, IEFDB470.

Exit to: Caller.

Called routines: None.

IEFDB4FB – Change DDNAME/JES3 Exit

Function: This module interfaces with the JES3 subsystem whenever a DDNAME or relative position number change has been made by Dynamic Allocation.

Entry from: IEFDB410, IEFDB411, IEFDB450.

Exit to: Caller.

Called routines: None.

IEFDB4FC – Ddname Search Routine

Function: This module searches the TIOT for a specific ddname. It begins its search at the starting DSAB address passed as input by the caller.

Entry from: IEFDB4A0, IEFDB4FD, IEFDB414, IEFDB450, IEFDB460, IEFDB470, IEFDB490.

Exit to: Caller.

Called routines: None.

IEFDB4FD -- Ddname Generation Routine

Function: This routine generates an eight character ddname.

Entry from: IEFDB410, IEFDB414.

Exit to: Caller.

Called routines: IEFDB4FC.

IEFDB4FE – Obtain DSORG Routine

Function: This module determines the data set organization for the data set associated with the input DSAB.

Entry from: IEFDB410, IEFDB413, IEFDB470.

Exit to: Caller.

Called routines: None.

IEFDB4FF – Syntax Checker

Function: This module determines whether the user has entered an invalid key, mutually exclusive keys, or duplicate keys. It also checks whether an invalid number of parameters has been entered with each key, and whether the length of each parameter value is valid.

Entry from: IEFDB4A0, IEFDB412, IEFDB450, IEFDB460, IEFDB470, IEFDB480, IEFDB490.

Exit to: Caller.

Called routines: None.

IEFDB4F0 – Dynamic Allocation Key Verification

Function: This module invokes the dynamic allocation syntax checker (IEFDB4FF) to check the text unit with non-JDT defined keys. It also invokes the scheduler JCL facility (SJF) update routine (IEFSJUPD) to check the text unit with JDT defined keys.

Entry from: IEFDB412.

Exit to: Caller.

Error Exit RTM via X'05C' ABEND, reason code 701.

Called routines: IEFDB4FF, IEFSJCNL (via SJREQ).

IEFDB4F8 – Allocate TCTIOT Control

Function: Following a dynamic allocation, this module updates an existing TCTIOT or obtains and updates a new TCTIOT if the existing TCTIOT cannot be expanded.

Entry from: IEFDB4F9.

Exit to: Caller.

Called routines: None.

IEFDB4F9 – SMF Dynamic DD Routine

Function: This module updates the TCTIOT to reflect any changes made to the TIOT as a result of a dynamic allocation, unallocation, concatenation, or deconcatenation. If the function is unallocation, concatenation, or deconcatenation, an SMF type 40 record is built and written.

Entry from: IEFDB4A1, IEFDB413, IEFDB450, IEFDB460.

Exit to: Caller.

Called routines: IEEMB848, IEFDB4F8.

IEFDB400 – Dynamic Allocation Control Routine

Function: To process the input request, this module routes control to one of the following functions: dsname allocation, ddname allocation, concatenation, deconcatenation, remove in-use attribute, information retrieval or unallocation.

Entry from: SVC interrupt handler.

Exit to: Caller.

Called routines: IEFAB4FE, IEFAB4F7, IEFDB4A0, IEFDB4F9, IEFDB401, IEFDB410, IEFDB450, IEFDB460, IEFDB470, IEFDB480, IEFDB490.

IEFDB401 – Dynamic Allocation Installation Exit

Function: The Dynamic Allocation facility of the control program exits to this 'Validation Routine' before doing any processing on behalf of a Dynamic Allocation request. It is entered for all requests, foreground and background. This routine may test and modify the Dynamic Allocation input, and indicate through a return code whether processing is to continue or if the request is to be terminated. This IBM version arbitrarily accepts all requests.

Entry from: IEFDB400.

Exit to: Caller.

Called routines: None.

IEFDB402 – Dynamic Allocation ESTAE Exit

Function: This module performs clean-up processing when the task under which SVC 99 is executing abnormally terminates.

Entry from: IEFAB4ED.

Exit to: Caller.

Called routines: IEFDB418.

IEFDB403 – TCTIOT FRR Routine

Function: This exit routine receives control if an ABEND occurred while TCTIOT updating was taking place. This module sets to zero the pointer to the TCTIOT if it is unusable and frees TCTIOT storage.

Entry from: RTM (recovery termination management).

Exit to: Caller.

Called routines: None.

IEFDB410 – Allocate Function Control

Function: This module is the control routine for the dynamic allocation of a data set.

Entry from: IEFDB400.

Exit to: Caller.

Called routines: IEFAB4F7, IEFDB4A0, IEFDB4FB, IEFDB4FD, IEFDB4FE, IEFDB411, IEFDB412, IEFDB413, IEFDB460.

IEFDB411 – Dynamic Allocation Convert Routine

Function: This module modifies an existing allocation to satisfy the user's request.

Entry from: IEFDB410.

Exit to: Caller.

Called routines: IEFAB4E9, IEFDB4FB, IEFDB417, IEFGB4DC.

IEFDB412 – Dynamic Allocation Function Validity Checker

Function: This module validity checks the allocation function's input text against invalid, duplicate, or mutually exclusive keys; invalid number of keyword parameters or parameter lengths; absence of mutually inclusive parameters; invalid parameter values; and improper authorization to request a subsystem data set.

Entry from: IEFDB410.

Exit to: Caller.

Called routines: IEFDB4FF.

IEFDB413 – Normal Dynamic Allocation Control

Function: This routine controls the processing for a normal dynamic allocation request.

Entry from: IEFDB410.

Exit to: Caller.

Called routines: IEFAB4A0, IEFAB4FC, IEFAB4F7, IEFAB421, IEFAB451, IEFDB4FE, IEFDB4F9, IEFDB414, IEFDB418, IEFGB4DC.

IEFDB414 — Build SWA Tables For Dynamic Allocation Request

Function: This module creates and sets up a SIOT, JFCB, and if necessary, JFCBX(s), and/or a SSWA.

Entry from: IEFDB413.

Exit to: Caller.

Called routines: IEFAB4F7, IEFDB4FD, IEFDB4FC, IEFDB417, IEFSJCNL (via SJFREQ).

IEFDB417 — Dynamic Allocation JFCB - DCB Field Update

Function: This module updates fields in the JFCB for DCB keys specified in the user's request.

Entry from: IEFDB411, IEFDB414.

Exit to: Caller.

Called routines: None.

IEFDB418 — Dynamic Allocation Normal Error Subroutine

Function: This module serves as a subroutine for clean up processing if an error occurs in dynamic allocation. It releases resources acquired during processing of a dynamic allocation request and ensures that the 'WRITE-EPA' chain is written to SWA.

Entry from: IEFDB402, IEFDB413.

Exit to: Caller.

Called routines: IEFAB4A0, IEFGB4DC, IEFAB4F7.

IEFDB440 — Allocation Device User Routine

Function: This module is a general allocation service routine that provides the following three functions.

- The acquire function marks the specified device allocated.
- The release function marks the specified device unallocated.
- The retrieve function returns the allocation status information for a specified device.

Entry from: Various system services and user programs such as dumping services.

Exit to: Caller.

Called routines: IEFAB4FF, IEFHB410.

IEFDB450 — Concatenation Routine

Function: This module concatenates existing allocations in the order in which they are specified in the input text units.

Entry from: IEFDB400.

Exit to: Caller.

Called routines: IEFAB4FC, IEFAB4F7, IEFDB4FB, IEFDB4FC, IEFDB4FF, IEFDB4F9.

IEFDB460 — Deconcatenation Routine

Function: This module disassociates members of a dynamically concatenated group by restoring the pre-concatenation ddnames to the members and removing the 'Dynamically Concatenated' (DSABDCAT) and 'Member of a Concatenated Group' (DSABCATM) attributes from the respective DSABs.

Entry from: IEFDB4A0, IEFDB400, IEFDB410.

Exit to: Caller.

Called routines: IEFAB4FC, IEFDB4FF, IEFDB4F9.

IEFDB470 — Dynamic Information Retrieval

Function: This module provides the user with information about his current allocation environment.

Entry from: IEFDB400.

Exit to: Caller.

Called routines: IEFDB4FA, IEFDB4FC, IEFDB4FE, IEFDB4FF.

IEFDB480 — Remove In-Use Control Routine

Function: This module handles dynamic allocation (SVC 99) requests for remove in-use processing. It validity checks the request, determines what DSABs should be processed, and invokes the remove in-use processor (IEFDB481).

Entry from: IEFDB400.

Exit to: Caller.

Called routines: IEFDB4FF, IEFDB481.

IEFDB481 — Remove In-Use Processor

Function: This module performs remove in-use processing for the DSABs pointed to by the input list.

Entry from: IEFDB4A0, IEFDB480.

Exit to: Caller.

Called routines: IEFDB4A1.

IEFDB490 — Ddname Allocation

Function: This module assigns the in-use attribute to the resource associated with the specified ddname.

Entry from: IEFDB400.

Exit to: Caller.

Called routines: IEFDB4FC, IEFDB4FF.

IEFDOCNL

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Dynamic Output*

IEFDPOST — Device Interrupt Routine

Function: This module is invoked by IOS to post tasks waiting for devices to become ready.

Entry from: IOS (input/output supervisor).

Exit to: Caller.

Called routines: IEAOPT01.

IEFDSLST — Data Set Enqueue Parameter List Builder

Function: This module builds a data set enqueue parameter list for each job; to do this, it uses input passed by IEFDSTBL.

Entry from: IEFSD161 via branch.

Exit to: IEFSD161 via branch.

IEFDSTBL — Data Set Tree Builder

Function: This module eliminates duplicate data set names from the data set enqueue parameter list that exists for each job.

Entry from: IEFSD161 via branch.

Exit to: IEFSD161 via branch.

IEFEB400 — EDT Verification Routine

Function: This module verifies that the devices defined by the EDT contained in the input data set match the devices currently existing in the nucleus.

Entry from: Program manager.

Exit to: Caller.

IEFENFDM — Event Notification Control Table

Function: This module contains the event notification control table (IEFENFCT).

Entry from: Not applicable (non-executable module).

Exit to: None.

IEFENFFX — Event Notification Request Router

Function: This module checks the event parameter list and determines if the request is synchronous or asynchronous.

Entry from: Event notification user.

Exit to: Caller.

Called routines: IEFENFNM.

IEFENFIN — Event Notification Facility Interface

Function: This module provides an interface between a routine executing in either 24 or 31-bit mode and one of the following ENF modules (executing in 31-bit mode):

- IEFENFFX - ENF Service Router.
- Entry-point IEFENFLX and IEFENFSX in IEFENFNM (from the exit routines).

Entry-Point IEFENFIN:

> Callers: A system routine via ENFREQ macro or BALR using the address of this module in the ENFCT (ENFCFMOD).
>
> Exit-Normal: IEFENFFX.
>
> Exit-Error: No exit error conditions.

Entry-Point IEFENFLX:

> Callers: A listen or signal exit routine.
>
> Exit-Normal: IEFENFFX.
>
> Exit-Error: No exit error conditions.

Entry-Point IEFENFSX:

> Callers: A listen or signal exit routine.

Exit-Normal: IEFENFFX.

Exit-Error: No exit error conditions.

External-References: None.

IEFENFNM – Event Notification Mainline

Function: This module processes all event notification listen, signal, and delete listen requests.

Entry from: IEFENFWT, IEFENFFX.

Exit to: Caller.

IEFENFWT – Event Notification Wait Routine

Function: This module handles processing of asynchronous event notification requests.

Entry from: IEEMB860 via ATTACH and IEFENFFX via POST.

Exit to: None, this is a never ending task.

Called routines: IEFENFNM.

IEFGB400 – Allocation Put Routine

Function: This module issues a PUT macro which thereby allows modules in 31-bit addressing mode to issue a PUT macro.

Entry from: IEEAB401.

Exit to: Caller.

Called routines: None.

IEFGB4DC – Data Set Reservation/Release

Function: This module can perform the following functions:

- Updates the data set enqueue table to add or change an entry.
- Dequeues the dsname:
- Updates the data set enqueue table to delete an entry.

Entry from: IEFAB4DC, IEFAB459, IEFDB4A1, IEFDB411, IEFDB413, IEFDB418.

Exit from Caller.

Called routines: IEFAB4F7.

IEFGB4UV – Unit Verification

Function: This routine is invoked by system routines that require information or validity checks based on the contents of the eligible devices table (EDT).

Entry from: IEFAB4EC, IEFAB4UV, IEFAB441.

Exit to: Caller. (If the caller is IEFAB4UV, this module exits to the caller of IEFAB4UV.)

Called routines: IEFAB438.

IEFHB4I2 – Display Allocation Scavenge Routine

Function: See *Component Diagnosis: System Initialization* for a module description of IEFHB4I2.

IEFHB410 – Display Allocation Tables Manager

Function: This module handles all updates to the display allocation tables in the allocation address space (ALLOCAS). IEFHB410 executes in cross memory mode. It is entered by means of a program call (PC) to perform one or more of the following functions.

- Updates the DALT use count field.
- Returns the DALT use count for a unit.
- Returns the DALT use count for one address space and one unit.
- Clears the DALT for an address space.

Entry from: IEFAB4A4, IEFAB4E5, IEFAB4E6, IEFAB434, IEFAB435, IEFHB4I2, IEFAB477.

Exit to: Caller by means of a program transfer (PT).

IEFIB600 – Initiator/Subsystem SWA Interface

Function: This module builds the JSCB chain for a job, invokes the interpreter, and builds a CSCB for a job that is not a started task or LOGON.

Entry from: The master subsystem or the job entry subsystem via LINK.

Exit to: Caller via branch.

IEFIB605 – SWA Reconstruction Module

Function: This module reconstructs SWA for restarted jobs by invoking the SWA merge routine and the data set description record processor.

Entry from: IEFIB605 via CALL.

Exit to: Caller via branch.

IEFIB620 – ESTAE Exit Routine

Function: Depending upon what type of error caused this module to receive control, it records the error and then either takes a dump or prepares for a retry.

Entry from: IEAVTAS1 via a SYNCH macro.

Exit to: IEAVTAS1 via branch.

IEFIB621 – Initiator Task Recovery Retry Routine

Function: This module attempts to resume normal initiator processing after a program check, an ABEND, or machine check has occurred, or after an operator has pushed the RESTART key.

Entry from: This module receives control and executes as a result of an RB (request block) created by the recovery termination management routines.

Exit to: IEFSD164 via branch.

IEFIB645 – SWA Create Exit Routine

Function: This module receives control whenever an error has occurred during interpreter, restart, or SWA create interface processing. It records the error on the system error record data set and then attempts a retry.

Entry from: Recovery termination management via LINK.

Exit to: Caller via branch.

IEFIB650 – Initiator Message Module

Function: This module contains text for initiator messages.

IEFIB660 – Build TCTIOT Routine

Function: This module constructs a TCTIOT and an ETCTIOT based on the current TIOT. If a TCTIOT or ETCTIOT exists on entry to this module, it is freed. This situation could occur if dynamic allocation obtained these control blocks prior to initiation.

Entry from: IEFAB820.

Exit to: Caller.

Called routines: None.

IEFICATL – Initiator Interface to Allocate Catalog Control

Function: This module determines the data set names of each catalog to be used by a job step/task and then invokes IEFAB4F5 to open the catalogs.

Entry from: IEFSD162 via BALR.

Exit to: IEFSD162 via branch.

IEFICPUA – Assignment of CPU Task Affinity

Function: This module indicates to the calling module which processor(s) can be used to run a job step/task; it also notifies the calling routine if the required processor(s) is not on line.

Entry from: IEFSD101 for the first job step in a job and from IEFSD101 for subsequent steps.

Exit to: The calling module via branch.

IEFIIC – Initiator Interface Control

Function: This module puts the initiator routines into supervisor state and builds the initiator's SSIB, SSOB header, entrance, options, and exit list (IEL).

Entry from: IEFSD263 via ATTACH.

Exit to: IEFSD060 via branch.

IEFIMASK – Conversion of Bit Mask

Function: This routine converts bit positions in a string to hexadecimal characters for inclusion in some initiator messages.

Entry from: IEFSD101 or IEFSD161 via branch.

Exit to: The calling module via branch.

IEFIRECM – Initiator Resource Manager

Function: This module cleans up resources used by the initiator in a terminating address space. This routine deletes all CSCBs associated with this address space from the CSCB chain.

Entry from: RTM (via address in CVTIRECM field).

Exit to: RTM.

IEFISEXR – Initiator Subsystem ESTAE Exit Routine

Function: This module receives control after an abnormal situation occurs in the initiator or subsystem interface resource managers. It issues a DEQ for the CSCB chain and makes an error entry in the LOGREC data set.

Entry from: RTM (via address in CVTJRECM field).

Exit to: RTM.

IEFI922B – Builder of 922 Completion Code Interface

Function: This routine builds an interface to module IEFSD164 to enable the termination of a job step/task which has already successfully completed allocation processing.

Entry from: IEFIB621 via CALL.

Exit to: IEFIB621 via branch.

IEFJACTL – Pseudo Access Method Control

Function: I his routine, a part of the master subsystem, provides a data manipulation service for the converter and interpreter at a time when no access method services are available via the RPL/ACB interface.

Entry from: The converter or interpreter.

Exit to: Caller.

Error exit To caller's ESTAE routine with a user abend code X'0B1' or X'0B3'.

IEFJCDLT – Storage Deletion Routine

Function: This routine, a part of the master subsystem, frees the storage no longer needed by the master subsystem after creating SWA control blocks for a task.

Entry from: IEFJJOBS.

Exit to: Caller.

IEFJCNTL – JCLS to SWA Conversion

Function: This routine, a part of the master subsystem, converts a JCLS chain to SWA control blocks by invoking the converter, then the interpreter (via the SWA-create interface).

Entry from: IEFJJOBS.

Exit to: Caller.

Error exit To caller's ESTAE routine with a user abend code X'0B1', X'0B4', or X'0B5'.

IEFJDIRD – Pseudo Access Method Direct Read

Function: This routine, a part of the master subsystem, moves the specified record into a specified buffer so that the record can be updated.

Entry from: The converter via the RPL/ACB interface.

Exit to: Caller.

IEFJDSNA – Data Set Name Assignment

Function: This routine, a part of the master subsystem, assigns a data set name to each SYSOUT data set specified in the master JCL or in the JCL used to start a subsystem.

Entry from: Allocation.

Exit to: Caller.

IEFJDWRT – Pseudo Access Method Direct Write

Function: This routine, a part of the master subsystem, overlays an old record with an updated record from a buffer.

Entry from: The converter via the RPL/ACB interface.

Exit to: Caller.

IEFJJCLS – JCL to JCLS Conversion

Function: This routine, a part of the master subsystem, converts the master scheduler JCL (MSTRJCL) to a JCLS chain.

Entry from: IEFJJOBS.

Exit to: Caller.

Error exit To caller's ESTAE routine with a user abend code X'0B1'.

IEFJJOBS – Subsystem Initiation

Function: This routine, a part of the master subsystem, controls the creation of SWA control blocks when the master scheduler or a subsystem is being started.

Entry from: The initiator via the subsystem interface.

Exit to: The initiator.

Error exit To caller's ESTAE routine with a user abend code X'0B1'.

IEFJJTRM – Subsystem Job Termination

Function: This routine, a part of the master subsystem, is a dummy routine. It replaces the normal job termination function when the job being terminated is a subsystem.

Entry from: A system routine via the IEFSSREQ macro.

Exit to: The system routine.

IEFJRASP – Common Request Router

Function: This routine, a part of the master subsystem, handles requests that require processing for all active subsystems except the master subsystem.

Entry from: A system routine via the subsystem interface.

Exit to: The system routine.

IEFJREAD – Pseudo Access Method Sequential Read

Function: This routine, a part of the master subsystem, moves a record from a chain of records to a buffer and prepares for the next record to be read from the chain.

Entry from: The converter or interpreter via the RPL/ACB interface.

Exit to: Caller.

IEFJRECM – Subsystem Interface Resource Manager

Function: This module checks the resource manager parameter list for task or address space termination indicators. It then builds either a task or address space SSOB extension. If an address space is terminating, it also builds a list of job names associated with the terminating address space. It then issues the IEFSSREQ macro to communicate end-of-task or end-of-address space to all active subsystems. RTM enters this module at secondary entry point IEFJRECF to notify all active subsystems of an end-of-task. At this entry point, the module builds the end-of-task SSOB extension and issues IEFSSREQ macro. IEFJRECF is located before IEFJRECM in RTM's list of end-of-task resource managers.

Entry from:

For entry point IEFJRECM: RTM using the address in the JESRECM field of the JESCT.

For entry point IEFJRECF: RTM using the address in the JESRECF field of the JESCT.

Exit to: RTM.

IEFJSBLD – Subsystem Service Routine

Function: See *Component Diagnosis: System Initialization* for a module description of IEFJSBLD.

IEFJSDTN – Subsystem Determination

Function: This routine, a part of the master subsystem, determines if the task being started is a subsystem or if the subsystem being tested is active and capable of starting a job.

Entry from: The initiator via the subsystem interface.

Exit to: Return to the initiator.

IEFJSIMM – Subsystem Initialization and PARMLIB Messages

Function: This module contains messages for subsystem initialization and PARMLIB processing. These messages are accessed and issued by IEFJSIMW.

Entry from: None.

Exit to: None.

IEFJSIMW – Subsystem Initialization and Parmlib Processing Message Writer

Function: This module issues messages, which are contained in IEFJSIMM. It issues these messages during subsystem initialization (for IEFJSBLD and IEFJSIN2) and during the processing of the SET MPF command (for IEECB805).

Entry from: IEECB805 and modules used during subsystem initialization (IEFJSBLD and IEFJSIN2). (See *Component Diagnosis: System Initialization.*)

Exit to: Caller.

IEFJSINT – Subsystem Interface Initialization

Function: See *Component Diagnosis: System Initialization* for a module description of IEFJSINT.

IEFJSIN2 — Subsystem Initialization

Function: See *Component Diagnosis: System Initialization* for a module description of IEFJSIN2.

IEFJSREQ — Subsystem Interface

Function: IEFJSREQ handles requests for subsystem interface services from the IEFSSREQ macro. It determines which routine of which subsystem is to receive control and then passes control to that routine via a branch instruction.

Entry from: A system routine via the IEFSSREQ macro.

Exit to: The subsystem routine to perform the requested function. (See the figure "Subsystem Interface Summary," for the module names.)

Error exit Caller with nonzero return code.

IEFJSWT — STC Write JCL Routine

Function: This module writes the internal JCL text for a START, MOUNT, or LOGON command.

Entry from: IEESB605 via BALR.

Exit to: Caller via branch.

IEFJWRTE — Pseudo Access Method Sequential Write

Function: This routine, a part of the master subsystem, adds a record to a chain of records by obtaining storage, moving the new record to that storage area, and chaining the previous record to the new one.

Entry from: The converter via the RPL/ACB interface.

Exit to: The converter.

IEFJWTOM — Subsystem Initiation Message Writer

Function: This routine, a part of the master subsystem, writes a hard copy of the converter, interpreter, and allocation error messages (including the JCL card images in error) for tasks being started via the master subsystem.

Entry from: The converter, interpreter, or allocation via the RPL/ACB interface.

Exit to: Caller.

IEFNB901 — Data Management Interpreter Exit Routine

Function: This module builds a parameter list for the data management interpreter routine and then passes control to that module which processes the AMP keyword parameter.

Entry from: From IEFVDA via branch.

Exit to: Data management interpreter routine via LOAD and BALR, caller via branch.

IEFNB903 — Interpreter Initialization Routine

Function: This routine obtains storage for and partially initializes the interpreter work area (IWA), its I/O buffer, the JMR, and the DSENQ table. It also establishes an ESTAE environment for the interpreter.

Entry from: IEFIB600 via LINK.

Exit to: IEFVHE via branch or to the journal write routine via branch for restarted jobs.

IEFQBDIR — SWA Manager Diagnostic Information Routine

Function: IEFQBDIR assists modules which access above the line Scheduler Work Area (SWA) blocks in debugging ABEND situations.

Entry from: IEFQBDIR

Exit to: Caller.

Called routines: Allocation.

IEFQBINT — SWA Manager Initialization Routine

Function: See *Component Diagnosis: System Initialization* for a module description of IEFQBINT.

IEFQBJST — SWA Manager Storage Table for JES3

Function: IEFQBJST is a 256-byte translate table used by SWA Manager to assign blocks to JES3.

IEFQBMST — SWA Manager Storage Table

Function: IEFQBMST is a 256-byte translate table used by SWA Manager to assign blocks above and below the line.

IEFQB550 – SWA Manager Move Mode Interface Routine

Function: This module intercepts branches to SWA manager move mode routine and routes the requests to IEFQB551 in EPLPA.

Entry from: Initiator/terminator, scheduler restart, converter/interpreter, JES3.

Entry to IEFQB551 via BSM.

IEFQB551 – SWA Manager Move Mode Routine

Function: This module assigns, writes, reads, or deletes scheduler work area (SWA) blocks in move mode.

Entry from: IEFQB550, IEFQB580, or IEFQB585.

Exit to: Caller via BSM.

IEFQB555 – SWA Manager Locate Mode Interface Routine

Function: This module intercepts branches to SWA manager locate mode routine and routes the requests to IEFQB556 in EPLPA.

Entry from: Scheduler restart or JES3.

Exit to: IEFQB556 via BSM.

IEFQB556 – SWA Manager Locate Mode Routine

Function: This module assigns, writes, reads, or deletes scheduler work area (SWA) blocks in locate mode.

Entry from: IEFQB555 or directly from the issuer of the SWAREQ macro instruction.

Exit to: Caller or issuer of SWAREQ via BSM.

IEFQB580 – QMNGRIO Macro Interface Handler

Function: This module provides an interface to SWA manager functions from data management routines that use the QMNGRIO macro instruction. IEFQB580 builds and initializes a queue management parameter area (QMPA) for the module requesting SWA manager functions.

Entry from: Data management routines via QMNGRIO macro instruction.

Exit to: IEFQB551 via BASSM.

IEFQB585 – SWA Manager Interface Routine

Function: This module intercepts any references to previously existing queue manager modules. IEFQB585 inserts the appropriate function code in the queue management parameter area (QMPA) on behalf of the requesting module.

Entry from: Modules that reference previously existing queue manager modules.

Exit to: IEFQB551 via BSM.

IEFQMWR – Allocation/Termination Communication Area

Function: This non-executable CSECT contains the address of the nucleus-resident update DDR count routine (IEFAB438) and a pointer to the first MVCA on the MVCA chain. This area is called the allocation/termination communication area (ATCA) and is mapped by macro IEFZB432.

IEFRCSTP – Restart Codes Statement Processor

Function: This routine processes RESTART/NORESTART statements in the SYS1.PARMLIB member SCHEDxx. It modifies the standard list of ABEND codes eligible for automatic restart. The list of codes is contained in module IEFYRCDS, a data only module residing in SYS1.LINKLIB.

Entry-Point IEFRCADD: This routine adds user-defined ABEND codes to the list of ABEND codes (RESTART statement).

Callers: IEEMB888.

Exit-Normal: Return to the caller.

Exit-Error: None.

Entry-Point IEFRCDEL: This routine deletes ABEND codes from the list of ABEND codes (NORESTART statement).

Callers: IEEMB888.

Exit-Normal: Return to the caller.

Exit-Error: None.

External References: No routines.

IEFRPREP – Restart Preparation Routine

Function: This module determines if an ABENDed job step task can be restarted.

Entry from: IEFBB410.

Exit to: IEFBB410.

IEFSDPPT – Program Properties Table

Function: This routine contains non-executable code.

IEFSD060 (IEFSD160) – Initiator Control Initialization

Function: This module builds the linkage control block (LCT) for a job step/task to be processed by the initiator and a STAE parameter list (STEPL) for the initiator's STAE exit routine.

Entry from: IEESB605 via LINK for LOGON, START, and MOUNT commands; IEFIIC via branch for normal initiator processing; IEEVIPL via ATTACH for master scheduler initialization.

Exit to: IEFSD061 via branch.

IEFSD061 (IEFSD161) – Job Select Routine

Function: This routine requests jobs for the initiator from the job entry subsystem and begins STOP processing as required for started tasks. For a selected job to be warmstarted, IEFSD061 sets up for warmstart processing; for all other selected jobs this module assigns special protect keys and processes data set integrity, if required.

Entry from: IEFSD160 via branch for job selection or from IEFSD166 for job selection or stop processing.

Exit to: A generalized initiator exit via branch, LINK, or XCTL for internal stop processing; to IEFSD164 via branch for warmstart; to IEFSD101 for normal initiator processing.

IEFSD062 (IEFSD162) – Device Allocation Interface Routine

Function: This module invokes device allocation for a job step/started task, opens a JOBLIB, STEPLIB, or FETCHLIB required by the job step/started task, and completes assignments of special properties.

Entry from: IEFSD102 via branch.

Exit to: IEFSD103 via BALR, to IEFSD164 via BALR if an error occurred during allocation or OPEN catalog processing.

IEFSD064 (IEFSD164) – Step Delete Routine

Function: When a job step/task completes processing, this module closes the JOBLIB, STEPLIB, or FETCHLIB DCBs, performs job and job step timing calculations, and builds a dummy TCB to be used by termination routines.

Entry from: IEFSD161 via branch for warmstarted jobs; IEFSD162 via branch when an error occurs in allocation for a job step/started task; IEFSD263 via branch when a job step/started task completes processing; IEFIB621 to retry after ESTAE processing.

Exit to: IEFSD101 via branch to initiate the next step in a job; IEFSD166 via branch to delete a job when all job steps have completed or to suspend a job, if necessary.

IEFSD066 (IEFSD166) – Job Delete Routine

Function: This module deletes a job and its associated control blocks when the job completes processing or re-enqueues a job when necessary.

Entry from: IEFSD164 via branch.

Exit to: IEFSD161 via branch.

IEFSD101 – PPT Scan

Function: This routine scans the program properties table (PPT) and assigns special properties to a job step/started task as indicated.

Entry from: IEFSD161 via branch for the first job step in job and from IEFSD164 via branch for subsequent job steps.

Exit to: IEFSD102 via branch.

IEFSD102 – Data Set Enqueue

Function: This routine enqueues on the data sets required for an entire job.

Entry from: IEFSD101 via branch.

Exit to: IEFSD162 via branch.

IEFSD103 – ATTACH Interface Routine

Function: This module builds the ATTACH parameter list.

Entry from: IEFSD162 via branch.

Exit to: IEFSD263 via branch.

IEFSD263 – Initiator ATTACH Module

Function: This module gets a region for a job step/started task, if one is required, attaches the job step/started task, and then waits for an end-of-task or cancel ECB to be posted. When the appropriate ECBs are posted, this module detaches the job step/started task.

Entry from: IEFSD103 via branch.

Exit to: IEFSD164 via branch.

IEFSJBLD – Scheduler JCL Facility (SJF) Build SWB Routine

Function: This module builds a scheduler work block (SWB).

Caller IEFSJWRT.

Exit-Normal: Return to the caller.

Exit-Error: Return to the caller.

Entry-Point BLDRETRY: This routine performs clean up processing when an ABEND occurs during SJF build processing.

Caller Recovery termination manager (RTM).

Exit-Normal: None.

Exit-Error: Return to the caller.

External References: SWA manager locate mode (IEFQB556) and the alternate SWA manager routine.

IEFSJCNL – Scheduler JCL Facility (SJF) Control Routine

Function: This routine performs common initial processing for the SJF function, routes the SJFREQ macro request to the specified SJF function, and upon return from the specified function, performs common clean up processing.

Caller IEFSJRTE.

Exit-Normal: Return to the issuer of the SJFREQ macro.

Exit-Error: Return to the issuer of the SJFREQ macro.

Entry-Point RECOVERY: This routine is used to recover from an error that caused the exit to RTM.

Caller RTM.

Exit-Normal: Returns to RTM specifying the retry address stored in the SJF work area.

Exit-Error: Percolates to the caller's recovery routine if the ABEND did not occur while SJF was processing or if a previous ABEND occurred.

Entry-Point RECCLEAN: This routine performs clean up processing when an ABEND occurs during the SJF control routine's processing.

Caller RTM.

Exit-Normal: None.

Exit-Error: Return to the issuer of the SJFREQ macro.

External References: IEFSJDEF, IEFSJDEL, IEFSJEXT, IEFSJFND, IEFSJGET, IEFSJINT, IEFSJJDV, IEFSJPUT, IEFSJRET, IEFSJUPD, and IEFSJWRT.

IEFSJDEF – Scheduler JCL Facility Define JDVT Routine

Function: This routine creates a JCL definition vector table (JDVT) and adds this table to the JDVT chain anchored off the JES control table (JESCT).

Callers: IEFSJCNL and IEFSJINT.

Exit-Normal: Return to the caller.

Exit-Error: Return to the caller.

Entry-Point DEFRETRY: This routine performs clean up processing when an ABEND occurs during SJF define JDVT processing.

Caller RTM.

Exit-Normal: None.

Exit-Error: Return to RTM.

External References: IEFSJHTB.

IEFSJDEL – Scheduler JCL Facility Delete SWB Chain Routine

Function: This routine deletes a scheduler work block (SWB) chain.

Callers: IEFSJCNL and IEFSJUPD.

Exit-Normal: Return to the caller.

Exit-Error: Return to the caller.

Entry-Point DELRETRY: This routine performs clean up processing when an ABEND occurs during SJF delete processing.

Caller RTM.

Exit-Normal: None.

Exit-Error: Return to the caller.

External References: SWA manager.

IEFSJEXT – Scheduler JCL Facility (SJF) Extract Routine

Function: This routine extracts information from the JCL definition table (JDT).

Callers: IEFSJCNL, IEFSJRET, and IEFSJUPD.

Exit-Normal: Return to caller.

Exit-Error: Return to caller.

Entry-Point EXTRETRY: This routine performs clean up processing when an ABEND occurs during the SJF extract routine′s processing.

Caller RTM.

Exit-Normal: None.

Exit-Error: Return to the caller.

External References: IEFSJJDV.

IEFSJFND – Scheduler JCL Facility Find SWB Chain Routine

Function: This routine locates a scheduler work block (SWB) chain at a particular level of the scheduler work area (SWA) structure.

Callers: IEFSJCNL and IEFSJUPD.

Exit-Normal: Return to the caller.

Exit-Error: Return to the caller.

Entry-Point FNDRETRY: This routine performs clean up processing when an ABEND occurs during SJF find SWB processing.

Caller RTM.

Exit-Normal: None.

Exit-Error: Return to the caller.

External References: No routines.

IEFSJGET – Scheduler JCL Facility (SJF) Get SWB Chain Routine.

Function: This routine copies selected keywords from a SWB chain in text unit format into a storage area specified by the caller.

Caller IEFSJCNL.

Exit-Normal: Return to the caller.

Exit-Error: Return to the caller.

Entry-Point GETRETRY: This routine performs clean up processing when a ABEND occurs during SJF get processing.

Caller RTM.

Exit-Normal: Return to the caller.

Exit-Error: Return to the caller.

External References: IEFSJJDV and IEFSJRET.

IEFSJHTB – Scheduler JCL Facility (SJF) Hash Table Build Routine

Function: This routine builds hash tables to provide access to information in the JDTs.

Caller IEFSJDEF.

Exit-Normal: Return to the caller.

Exit-Error: Return to the caller.

Entry-Point HTBRETRY: This routine performs clean up processing when an ABEND occurs during SJF hash table build processing.

Caller RTM.

Exit-Normal: None.

Exit-Error: Return to the caller.

External References: No routines.

IEFSJINT – Scheduler JCL Facility JDVT Initialization Routine

Function: This routine builds the system default JCL definition vector table.

Caller IEFSJCNL.

Exit-Normal: Return to the caller.

Exit-Error: Return to the caller.

Entry-Point INTRETRY: This routine performs clean up processing when an ABEND occurs during SJF JDVT initialization processing.

Caller RTM.

Exit-Normal: None.

Exit-Error: Return to the caller.

External References: IEFSJDEF and IEFJSIMW.

IEFSJJDV – Scheduler JCL Facility (SJF) Find JDVT Routine

Function: This routine locates a JCL definition vector table (JDVT).

Callers: IEFSJCNL, IEFSJEXT.

Exit-Normal: Return to the caller.

Exit-Error: Return to the caller.

Entry-Point JDVRETRY: This routine performs clean up processing when an ABEND occurs during SJF find JDVT processing.

Caller RTM.

Exit-Normal: None.

Exit-Error: Return to the caller.

External References: No routines.

IEFSJPUT – Scheduler JCL Facility (SJF) Put SWB Routine

Function: This routine rebuilds a SWB chain from SWB keyword data found in text unit format.

Caller IEFSJCNL.

Exit-Normal: Return to caller.

Exit-Error: Return to caller.

Entry-Point PUTRETRY: This routine performs clean up processing when an ABEND occurs during SJF put processing.

Caller RTM.

Exit-Normal: None.

Exit-Error: Return to the caller.

External References: IEFSJUPD and IEFSJWRT.

IEFSJRET – Scheduler JCL Facility (SJF) Retrieve Routine

Function: This routine retrieves parameter information from a scheduler work block (SWB) chain.

Caller IEFSJCNL.

Exit-Normal: Return to the caller.

Exit-Error: Return to the caller.

Entry-Point RETRETRY: This routine performs clean up processing when an ABEND occurs during SJF retrieve processing.

Caller RTM.

Exit-Normal: None

Exit-Error: Return to the caller.

External References: IEFSJEXT.

IEFSJUPD – Scheduler JCL Facility (SJF) Update Routine

Function: This routine verifies the text units specified by the caller and if requested, updates the SWB chain with the information specified in text unit format.

Callers: IEFSJCNL and IEFSJPUT.

Exit-Normal: Return to the caller.

Exit-Error: Return to the caller.

Entry-Point UPDRETRY: This routine performs clean up processing when an ABEND occurs during SJF update processing.

Caller RTM.

Exit-Normal: None.

Exit-Error: Return to the caller.

External References: IEFSJDEL, IEFSJEXT, IEFSJFND, IEFSJWRT, and IEFXB501.

IEFSJRTE – Scheduler JCL Facility (SJF) Router Routine

Function: This routine provides an addressing mode interface between the issuer of the SJFREQ macro and the SJF control routine (IEFSJCNL).

Callers: Issuers of the SJFREQ macro.

Exit Normal Exit to: IEFSJCNL.

Exit Error None.

External References: IEFSJCNL.

IEFSJWRT – Scheduler JCL Facility (SJF) Write SWB Routine

Function: This routine locates a specific SWB and updates its data portion.

Callers: IEFSJCNL, IEFSJUPD.

Exit-Normal: Return to the caller.

Exit-Error: Return to the caller.

Entry-Point WRTRETRY: This routine performs clean up processing when a ABEND occurs during SJF write processing.

Caller RTM.

Exit-Normal: None.

Exit-Error: Return to the caller.

External References: IEFSJBLD.

IEFSMFIE – SMF Initialization Exit Support Module

Function: This module constructs a timing control table (TCT) and supports user's job initiation exit routines and step initiation exit routines. It also issues SMF job commencement record type 20 and/or type 30.

Entry from: IEFSD101 via BALR.

Exit to: IEFSD101 via branch.

Called routines: IEFUSI, IEFUJI, IEFSMF30, and IEFSMF32.

IEFSMF30 – Type 30 Setup Record for SMF

Function: This module builds the model for type 30 SMF records. IEFSMFIE calls the module at job start; IEFTB721 calls it at step termination if a warm start occurred and previous records were lost.

Entry from: IEFSMFIE and IEFTB721.

Exit to: Caller.

IEFSMF32 – Type 32 Record Setup for SMF

Function: This module is called at job start by IEFSMFIE to build the model for type 32 SMF records.

Entry from: IEFSMFIE.

Exit to: Caller.

IEFTB721 – SMF Job and Step Termination Processor

Function: This module provides the user with job and step information after the job or step completes. The module contains an ESTAE routine to maintain control block integrity should a user error occur in the user exit, IEFACTRT.

Entry from: IEFSD263.

Exit to: Caller.

Called routines: IEFTB722, IEFSMF30, IEFTB727, IEFTB726, IEFACTRT.

IEFTB722 – SMF Job and Step Termination Record Assembler

Function: This module assembles job and step information into an SMF record format (type 4, 5, 34, or 35) after the job or step completes.

Entry from: IEFTB721.

Exit to: Caller.

IEFTB723 – SMF Exit Write-to-Programmer Processor

Function: This module provides an interface that can be used by IEFACTRT accounting exits to issue messages to the programmer.

Entry from: May be called via BALR from IEFACTRT. (IEFYS is an alternate entry name for IEFTB723).

Exit to: IEFACTRT via branch.

IEFTB726 – Type 30 SMF Record Builder

Function: This module builds a type 30 SMF record at step end.

Entry from: IEFTB721.

Exit to: Caller.

Caller routines IEFTB728.

IEFTB727 – Type 32 SMF Record Builder

Function: This module builds and writes the step type 32 record at step end or interval expiration.

Entry from: IEFTB721 and IEEMB836.

Exit to: Caller.

IEFTB728 — Delta Type 30 SMF Record Builder

Function: This module builds a type 30 SMF record which contains data representing changes made since the last record or step start, whichever is later.

Entry from: IEEMB836 and IEFTB726.

Exit to: Caller.

IEFUJI — User Job Initiation Exit Routine
IEFUJV — Converter/Interpreter SMF User Exit Routine
IEFUSI — User Step Initiation Exit Routine
IEFUTL — User Time Limit Exit Routine
IEFVDA — Interpreter DD Card Processor

Function: This routine creates and initializes the control blocks for a DD statement. It also checks the validity of DD keyword values and then sets a job fail indicator if it finds an invalid keyword value.

Entry from: IEFVHE via branch.

Exit to: IEFVHE via branch.

IEFVDBSD — Interpreter DSENQ Table Processor

Function: This module creates a DSENQ table for all explicitly named (by DSNAME parameter) data sets and marks each as exclusive or shared.

Entry from: IEFVDA or IEFVEA via branch.

Exit to: Caller via branch.

IEFVEA — Interpreter EXEC Statement Processor

Function: This routine creates a step control table (SCT) for an EXEC statement and the override and refer-back tables required for the step. It also chains the SIOTs and JFCBs for a job library, job catalog, and/or concatenation of data sets for the step.

Entry from: IEFVHE via branch.

Exit to: Caller via branch.

IEFVFA — Converter Scan Routine

Function: This module scans the JCL card images for syntax errors and merges JCL from the JCL data set and a cataloged or in-stream procedure if required. It then converts each JCL statement into internal JCL text.

Entry from: IEFVHEB via branch.

Exit to: IEFVHF via branch.

IEFVFB — Converter Symbolic Parameter Routine

Function: This module assigns values specified on an EXEC procedure statement or PROC statement to symbolic parameters that appear on JCL statements in cataloged or in-stream procedures.

Entry from: IEFVFA via BALR.

Exit to: Caller via branch.

IEFVGK — Interpreter GET Key/Positional Utility Routine

Function: This routine sets a pointer to the next JCL keyword or positional parameter in the internal text buffer for the JCL statement processor routines, IEFVJA, IEFVEA, or IEFVDA.

Entry from: IEFVJA via branch, or from IEFVEA via branch, or from IEFVDA via branch.

Exit to: Caller via branch.

IEFVGM — Converter/Interpreter Message Module

Function: This module puts error messages for the entire converter/interpreter into the message data set, and/or any required JCL statements to the list data set.

Entry from: IEFVJA via branch, or from IEFVEA via branch, or from IEFVDA via branch.

Exit to: Caller via branch.

IEFVGT — Converter Test and Store Utility Routine

Function: This module performs miscellaneous functions for the JCL statement processors, IEFVJA, IEFVEA, and IEFVDA, based on indicators in the parameter descriptor table passed to it by the processor routines.

Entry from: IEFVJA, IEFVEA, or IEFVDA via branch.

Exit to: Caller via branch.

IEFVHA — Converter GET Routine

Function: This module reads into an input buffer JCL statements from the JCL data set, from the procedure library, and/or from in-stream procedures.

Entry from: IEFVHC or IEFVIND via branch.

Exit to: IEFVHC via branch.

IEFVHC – Converter Comment or Continuation Validation Routine

Function: This routine determines whether a valid comment or continuation is indicated on a JCL statement.

Entry from: IEFVHA via branch.

Exit to: to IEFVHEB via branch, if a continuation was expected and was or was not received; to IEFVHA via branch, if a comment was received; to IEFVHCB, via branch, if no continuation was expected.

IEFVHCB – Converter Verb Identifier Routine

Function: This routine identifies the verb on a JCL statement and merges JCL statements from the JCL data set and the procedure library.

Entry from: IEFVHC, IEFVHL, or IEFVHF via branch.

Exit to: IEFVHA, via branch, when the JCL statement does not contain a verb; to IEFVHA, via branch, when the procedure library buffer must be primed; to IEFVHEB, via branch, when the JCL statement contains a JOB, EXEC, DD, or NULL verb; to IEFVHM, via branch, when the verb is unrecognizable; to IEFVINA, via branch, when a PROC verb is found.

IEFVHE – Interpreter GET and Route Routine

Function: This routine gets JCL text strings from the internal text data set and routes them to the appropriate processor. It also determines when the SCT for a step or the JCT for a job should be written into SWA.

Entry from: IEFNB903 via branch.

Exit to: IEFVJA via branch when a JOB statement text string is encountered; to IEFVEA via branch when an EXEC statement text string is encountered; to IEFVDA via branch when a DD statement text string is encountered; to IEFVHH when an SCT or JCT is to be written into SWA.

IEFVHEB – Converter Pre-scan Routine

Function: This routine performs checkpoint/restart functions, enters JCL statements into the list data set, and initializes the JMR for SMF processing.

Entry from: IEFVHC or IEFVHCB via branch.

Exit to: When IEFVHEB is used as a subroutine, it returns to its caller via branch; when it has been used to initialize a JMR, or to process a SYSCHK DD statement or a PROC statement, it branches to IEFVEA.

IEFVHF – Converter Termination Routine

Function: This module frees storage space used by the converter, issues messages to the operator about the status of the job after converter processing, initializes SYSCHK DD statement processing and deactivates the converter ESTAE environment.

Entry from: IEFVFA via branch.

Exit to: IEFVHA when continuation of JCL text is expected; IEFVHCB when no continuation is expected or to continue SYSCHK DD processing; IEFVHEB if a procedure statement is being overridden.

IEFVHH – Interpreter ENQUEUE Routine

Function: This module writes job and step tables into SWA as required and then performs miscellaneous clean-up processing for the interpreter.

Entry from: IEFVHE via branch.

Exit to: IEFVHN via branch.

IEFVHL – Converter NULL Statement or END-OF-FILE Processor

Function: This routine performs end-of-file processing for JCL statements in the JCL data set or in a cataloged or in-stream procedure.

Entry from: IEFVHEB via branch.

Exit to: IEFVHA, via branch, to continue reading a procedure; IEFVHEB, via branch to continue processing a JCL statement from a procedure; IEFVHCB, via branch, to continue reading JCL statements from the JCL data set; IEFVHF, via branch, when both the JCL data set and all procedures are completed.

IEFVHM — Converter Command Verb Validation Routine

Function: This routine determines if a valid command has been specified on a JCL statement.

Entry from: IEFVHCB via branch.

Exit to: IEFVHA via branch.

IEFVHN — Interpreter Termination Routine

Function: This module frees storage space used by the interpreter, issues messages to the operator about job status at the end of interpreter processing, and deactivates the ESTAE environment over the interpreter.

Entry from: IEFVHH via branch.

Exit to: The interpreter's original caller, SWA create interface, IEFIB600, via branch.

IEFVHQ — Converter SWA Manager Interface

Function: This routine is the interface between either the converter or interpreter and the SWA manager or between the interpreter and the journal merge routine.

Entry from: IEFVDA, IEFVEA, IEFVFB, IEFVHA, IEFVHH, IEFVINA, or IEFVJA via branch.

Exit to: Caller via branch.

IEFVHR — Converter/Interpreter Operator Message Module

Function: This module puts converter and interpreter error messages for the operator into the message data set.

Entry from: IEFVHA, IEFVHF, IEFVHM, or IEFVHN via branch.

Exit to: Caller via branch if recovery is possible; to IEFVHN, via branch, if recovery is not possible.

IEFVH1 — Converter Initialization Routine

Function: This routine performs initialization processing for the converter, obtains storage space for work areas and buffers, and establishes the ESTAE environment over the converter.

Entry from: The job entry subsystem or the master subsystem.

Exit to: IEFVHA via branch.

IEFVINA — Converter In-stream Procedure Control and GTF Routine

Function: This module reads in-stream procedures from the JCL data set, records each procedure, and saves each one in compressed form in the scheduler work area (SWA).

Entry from: IEFVHCB via branch.

Exit to: IEFVHA, via branch, for each successive statement in an in-stream procedure.

IEFVINB — Converter In-stream Procedure Directory Search Routine

Function: This routine searches through the procedure directory for a procedure name specified on a JCL statement. When it locates the procedure name, it returns to the calling routine the address at which that procedure begins.

Entry from: IEFVINA via branch.

Exit to: Caller via branch.

IEFVINC — Converter In-stream Procedure Directory Build Routine

Function: This module builds an entry for a specified procedure in the in-stream procedure directory.

Entry from: IEFVINA via branch.

Exit to: Caller via branch.

IEFVIND — Converter In-stream Procedure Expand Interface Routine

Function: This module functions as an access method for IEFVHA. Whenever IEFVHA issues a READ macro instruction for a record from an in-stream procedure, IEFVIND sets up required parameter lists and then invokes IEZDCODE to expand the record.

Entry from: IEFVHA via branch.

Exit to: IEZDCODE, via branch, to expand a record; to IEFVHQ, via branch, to read another record; to caller, via branch, when processing has completed.

IEFVINE — Converter In-stream Procedure Syntax Checker

Function: This routine checks the label field of a PROC or PEND statement appearing in an in-stream procedure.

Entry from: IEFVINA via branch.

Exit to: Caller via branch.

IEFVJA – Interpreter JOB Statement Processor

Function: This module creates and initializes the control blocks and tables required by a JOB statement and also checks the validity of JOB statement keyword values.

Entry from: IEFVHE via branch.

Exit to: Caller via branch.

IEFVJDTI – JDT-Defined JCL Interpreter

Function: This module interprets the JDT-defined JCL internal text and builds the associated scheduler work blocks (SWBs).

Entry from: IEFVGK.

Exit to: Caller.

IEFVJDTV – JDT-Defined JCL Verb Processor

Function: This module processes the JDT-defined verb statements.

Entry From IEFVHE.

Exit To Caller.

IEFVKMSG – Test EXEC Dependency Codes Message

Function: This module contains the message text of the message issued by IEFBB402, if a step is not run because of the COND parameter on the EXEC statement.

Entry from: Not applicable (non-executable module).

Exit to: Not applicable (non-executable module).

Called routines: Not applicable (non-executable module).

IEFXBPUT – Journal Write GET/PUT Interface Routine

Function: This module issues the GET, PUT, or ENDREQ macros to JES to access the job journal for the journal write routine (IEFXB501).

Entry from: IEFXB501.

Exit to: IEFXB501.

IEFXB500 – Journal Write Interface Routine

Function: This module intercepts branches to the journal write routine and routes the requests to IEFXB501 in EPLPA.

Entry from: VIO.

Exit to: IEFXB501 via BSM.

IEFXB501 – Journal Write Routine

Function: This module writes critical control blocks to the job journal for purposes of restart or termination.

Entry from: IEFXB500, initiator, scheduler JCL facility.

Exit to: Caller via BSM.

IEFXB601 – Journal Merge Routine

Function: This module merges the control blocks from the job journal to the scheduler work area (SWA) to establish a pre-job failure-environment.

Entry from: IEFIB605.

Exit to: IEFIB605.

IEFXB602 – Restart Interface Routine

Function: This module builds a virtual address table (VAT) to be used when the journal merge routine reconstructs SWA.

Entry from: IEFVHQ.

Exit to: IEFVHQ.

IEFXB603 – Automatic/Checkpoint Restart Message Module

Function: This module contains message texts for journal service and checkpoint routines. The module is non-executable.

Entry from: IEFXB500, IEFRPREP, IEFXB609, IEFXB601.

Exit to: Caller.

IEFXB604 – Step Header Create

Function: Except for automatic step or automatic checkpoint restarts, this module builds the step header record for each job step, before putting anything else in the job journal for the step.

Entry from: IEFSD162.

Exit to: IEFSD162.

IEFXB609 – Data Set Descriptor Record Processor

Function: This module makes the SWA entries for restarting jobs reflect the job environment at the time a checkpoint was issued.

Entry from: IEFIB605.

Exit to: IEFIB605.

IEFXB610 – Open Interface Module

Function: This routine opens the checkpoint data set.

Entry from: IEFXB609.

Exit to: IEFXB609.

IEFXVNSL – User-Replaceable Non-Standard Label Routine

Function: This user-replaceable module returns a code of 4 if it is called because no user non-standard label handling routine exists to process an NSL label volume.

Entry from: IEFAB473.

Exit to: Caller.

Called routines: None.

IEWFETCH – Program Fetch

Function: This module loads and relocates executable program modules and overlay segments into virtual storage for contents supervisor and overlay supervisor.

Entry-Point IEWMSEPT: Loads a program module.

> Callers: Contents supervisor.
>
> Exit-Normal: Returns to the caller.
>
> Exit-Error: No exit error conditions.

Entry-Point IEWMBOSV: Loads an overlay segment.

> Callers: Overlay supervisor.
>
> Exit-Normal: Returns to the caller.
>
> Exit-Error: No exit error conditions.

External-References: IEWFTCE, IEWFTSRB, IEWFTXCK, IEWFTPRG, IEWFTSTA.

IEWSUOVR – Overlay Supervisor - Resident Module

Function: This module routes SVC 37 and SVC 45 entry requests to IEWSWOVR.

Entry-Point IGC037:

> Callers: SVC interrupt handler.
>
> Exit-Normal: Returns to the caller.
>
> Exit-Error: When IEWSWOVR passes a nonzero return code, IEWSUOVR issues the ABEND codes listed in the ABEND CODES section.

Entry-Point IGC045:

> Callers: SVC interrupt handler.
>
> Exit-Normal: Returns to the caller.
>
> Exit-Error: When IEWSWOVR passes a nonzero return code, IEWSUOVR issues the ABEND codes listed in the ABEND CODES section.

External-References: IEWSWOVR.

IEWSWOVR – Overlay Supervisor Processor - Nonresident Module

Function: This module examines the status of the requested segment.

Entry-Point IEWSWOVR or IEWSZOVR (Alias):

> Callers: IEWSUOVR.
>
> Exit-Normal: Returns to the caller.
>
> Exit-Error: No exit error conditions.

External-References: IEWFETCH, IEAVLOCK.

IEZDCODE – Interpreter In-stream Procedure Decompression Subroutine

Function: This module re-expands the in-stream procedure records that were compressed by IEZNCODE.

Entry from: IEFVIND via branch.

Exit to: Caller via branch.

IEZNCODE – Interpreter In-stream Procedure Compression Subroutine

Function: This module compresses the data records that comprise an in-stream procedure by replacing blanks with count fields. It then blocks the compressed records.

Entry from: IEFVINA via branch.

Exit to: Caller via branch.

IFADSMF – SMF DISPLAY Command Processor

Function: This module displays the status of the SMF data sets or the SMF options in effect for the current session.

Entry from: IFASMF.

Exit to: Caller.

Called routines: IEEMB824, IEEMB833.

IFAEASI – SMF System Address Space Processing

Function: This module, which is scheduled as an SRB, records SMF accounting data for system address spaces that do not go through full-function start.

Entry from: SRB scheduler, scheduled by IFAINIT; IFAEASIL.

Exit to: Caller.

Called routines: IEEMB839 (SMF timer enqueue), IEEMF830 (branch entry to SMFEWTM).

IFAEASIL – SMF System Address Space Setup

Function: This module schedules an SRB to set up the SMF timer elements for address spaces that do not go through full-function start and are started after SMF initialization is complete.

Entry from: IEEPRWI2.

Exit to: Caller.

Called routines: IFAEASI, IEEMB839.

IFAINIT – SMF System Address Space Processing Initialization

Function: This module is called during SMF initialization. It checks all address spaces and schedules an SRB to the system address spaces that have not gone through full function start. The SRB causes SMF accounting data to be recorded for the address space.

Entry from: IEEMB823.

Exit to: Caller.

Called routines: IFAEASI, IEEMB839.

IFAPCWTR – SMF Record Buffering

Function: This module moves the SMF records from the user area into buffers in the SMF address space then schedules the writer SRB (IEEMB834) to write full buffers to the SMF data sets. It also controls the obtaining and freeing of SMF buffers.

Entry from: IEEMB830.

Exit to: Caller.

Called routines: Schedules the SMF Writer SRB (IEEMB834), posts IFASMF.

IFARPORT – SMF Summary Activity Report for IFASMFDP

Function: This module writes a summary report of records processed by IFASMFDP (the SMF dump program).

Entry from: IFASMFDP.

Exit to: Caller.

IFASMF – SMF Control Task

Function: This module completes the initialization of the SMF address space by preparing the SMF data sets for recording. After initialization, it controls the execution of SMF commands and IEEMB829 events by waiting on ECBs. It also obtains and frees buffers when requested by IFAPCWTR.

Entry from: Attached by the initiator to complete SMF initialization. Posted by IEEMB835, IEECB913, and IEECB916 to process SMF commands. Posted by IFASMFDP, IEEMB830, and IFAPCWTR to process IEEMB829 events.

Exit to: Caller.

Called routines: IEEMB822, IEEMB824, IEEMB829, IFADSMF, IFASSMF, IFATSMF.

IFASMFDP – SMF Dump Processing

Function: This module parses input, supplies defaults, and lists resultant options. It dumps input data sets in response to the keyword DUMP and clears VSAM input data sets in response to the keyword CLEAR.

Entry from: Problem program.

Exit to: Caller.

Called routines: IEEMB832, IEEMB833, IEEMB824, IFARPORT via BALR.

IFASSMF – SMF SETSMF Command Processor

Function: This module processes the SETSMF command. It calls IEEMB821 to parse the SMF options specified on the command and changes the SMCA values as indicated by the command options, then issues an SSI call to the subsystem whose options have been modified.

Entry from: IFASMF.

Exit to: Caller.

Called routines: IEEMB821, IEEMB823, IEEMB824, IEEMB829.

IFASTART – SMF Address Space Initialization

Function: This module is the SMF address space initialization routine, linked to by IEEPRWI2 when the SMF address is started. IFASTART initializes the SMF control blocks (SMCA/ACT/SLCA) and calls IEEMB821 to handle SMF parameter processing. If SMF recording is to be done, IFASTART obtains a temporary buffer area, loads the MAXDORM/STATUS processor, and starts timer services. IFASTART also calls IEEMB823 to write the SMF initialization records that describe the current IPL (0,8,22). Data set initialization is completed by the SMF control task (IFASMF).

Entry from: IEEPRWI2.

Exit to: Caller.

Called routines: IEEMB821, IEEMB823, IEEMB824, IEEMB839 via LINK.

IFATSMF – SET SMF Command Processor

Function: This module processes the SET SMF command by establishing new SMF options or restarting SMF. It derives the options from values specified in the SMF PARMLIB member specified on the SET command, SMF default values, and corrections entered at the console. Other functions that might perform include obtaining or writing BQEs, completing data set initialization, initializing timer services, writing a type 90 IPL record, and switching or closing SMF data sets.

Entry from: IFASMF.

Exit to: Caller.

Called routines: IEEMB821, IEEMB822, IEFJSREQ, IEEMB823, IEEMB824, IEEMB829, IEEMB839 via LINK.

IGC0001G – I/O Request Restore Routine

Function: This routine restores I/O requests that have been purged by SVC 16.

Entry-Point IGC017:

Callers: User or system programs that issue the RESTORE macro.

Exit-Normal: Returns to the caller.

External-References: Driver restore routines.

IGC0003C – HALT I/O SVC 33 Routine

Function: This module halts I/O activity on a TP, CTC, or 327X device by invoking the halt subchannel routine (IOSVHSCH) or by changing a CCW to a NOP.

Callers: User or system programs.

Exit-Normal: Returns to the caller.

Exit-Error: To RTM.

Entry-Point HALTDIE (Disabled Interrupt Exit): Indicates by a return code how IOS should process the start IOSB associated with the halted device.

Callers: IOS.

Exit-Normal: Returns to the caller.

Exit-Error: To RTM.

Entry-Point HALTPOST (Post Status Routine): Notifies IGC0003C that the halt request is complete.

Callers: IOS.

Exit-Normal: Returns to SRB dispatcher.

External-References: IECVEXPR(+4), IOSVHSCH, IOSVSTSQ.

IGC0008E – NSLREPOS Interface

Function: This module interfaces with the user written routine, NSLREPOS, to verify NSL tapes and to determine if re-positioning of the tape is required.

Entry from: Supervisor call instruction (SVC) interrupt handler.

Secondary entry point IGC0408E (return from user written routine, NSLREPOS).

Exit to: Caller.

Called routines: NSLREPOS (user written).

IGC0203E – Write to Programmer Routine

Function: To process WTP message received via WTO/WTOR.

Entry-Point IEAVC701:

Callers: IEAVVWTO or IEAVMWTO

Exit-Normal: Return to caller via BR 14

Exit-Error: IEAVVWTO or IEAVMWTO

External-References:

Routines:

- IEFGB400 - glue routine for PUT macro

Control Blocks:

- ACB
- ASCB
- CVT
- JESCT
- JSCB
- PSCB
- RPL
- SDWA
- SVRB
- TCB
- TIOT
- UCM
- UPT
- VRA
- WPL
- XVSAV

IGE0002E – Channel-To-Channel Error Routine

Function: This routine receives control from ERP loader (IECVERPL) to perform error recovery for channel-to-channel devices.

Callers: ERP loader (IECVERPL).

Exit-Normal: To one of the following:

Message writer (IGE0025C) via branch.

Outboard recorder (IGE0025F) via branch.

IOS post status (IECVPST) via SVC 15, then SVC 3.

IGE0010E – Local 3270 Error Recovery Routine

Function: This routine is the ERP for locally attached devices on the 3270 display system; it performs recovery services.

Entry from: Scheduled by IOS.

Exit-Normal: Returns to the supervisor via an SVC 3, returns to IOS (IGE0025C or IGE0025D) via a branch entry to XCTL.

Called routines: SVC 15 error EXCP.

IGE0025C – WTO Routine For ERPs

Function: IGE0025C formats messages for IOS and for the device-dependent ERPs. It invokes the appropriate message exit to insert device-dependent information into the message and issue the message.

Callers: ERP loader (IECVERPL).

Exit-Normal: To one of the following:

Statistics update routine (IGE0025D) via branch.

Outboard recorder (IGE0025F) via branch.

IOS-post status (IECVPST) via SVC F, then SVC 3.

External-References: CDE search routine, LPDE search routine, Device-dependent message exit, IOS storage manager.

IGE0025D – Statistics Update Routine

Function: This routine is used by the ERP routines to update the device-dependent statistics table.

Callers: ERP loader (IECVERPL).

Exit-Normal: To one of the following:

Outboard recorder (IGE0025F) via branch.

IOS post status (IECVPST) via SVC 15, then SVC 3.

External-References: None.

IGE0660A – Error Recovery Procedure (ERP) Interface

Function: After a permanent I/O error occurs, this module tests the existing conditions to determine whether DDR can be used to correct the error. If so, it schedules the DDR task.

Entry from: Outboard recorder (OBR).

Exit to: Dispatcher via an SVC 3.

Called routines: IGFDE0, IGFDT1, IGFDD1.

IGFDD0 – Direct Access Device Select Routine

Function: This module processes a swap request for a direct access device.

Entry from: IGFDI0.

Exit to: Caller.

Called routines: IGFDE0, IGFDD1, IGFDV0, IGFDM0.

IGFDD1 – DDR Disk Validate Routine

Function: This routine validates the disk devices involved in a disk swap.

Entry from: IGFDD0, IGE0660A, IGFDV0.

Exit to: Caller.

Called routines: IGFDV1, IGFDE0.

IGFDE0 – DDR Extended Specify Task Abnormal Exit (ESTAE) Controller

Function: This module establishes or removes the error recovery environment for DDR.

Entry from: IGF2503D, IGFDD0, IGFDV0, IGE0660A, IGFDT0, IGFDS1, IGFDU0, IGFDI0, IGFDI1.

Exit to: Caller.

Called routines: None.

IGFDE1 – DDR Extended Specify Task Abnormal Exit (ESTAE) Handler

Function: This routine releases the dynamically allocated storage and allocated resources of a module that has experienced an error. It also determines for the recovery termination manager whether a retry attempt should be made.

Entry from: Recovery termination manager (RTM).

Exit to: Caller.

Called routines: IGFPxxS, if this special exit routine exists.

IGFDI0 – DDR Initiator

Function: This module controls processing of the DDR queues. It locates a request for device swap, calls the appropriate routine to handle it, removes it from the queue, and repeats this process until the queue is empty. IGFDI0 also controls the repositioning of a tape volume for a tape swap.

Entry from: Dispatcher, attached by IGFDS1.

Exit to: Caller.

Called routines: IGFDU0, IGFDD0, IGFDT0, IGFDI1, IGFDE0, IGFDT2.

IGFDI1 – DDR Termination Module

Function: This module terminates processing of individual DDR requests and removes the DDR control block (DDRCOM) from the queue. It may also terminate the entire queue.

Entry from: IGFDI0 or IGFDE1.

Exit to: Caller.

Called routines: IGFDE0, IGFDM0.

IGFDL1 – DDR Candidate Lookup

Function: This module locates a compatible, on-line, unallocated device to be used in a system-initiated device swap.

Entry from: IGFDV0.

Exit to: Caller.

Called routines: IGFDT1, IGFDD1, IGFDE0.

IGFDMTXT — DDR Message Text Module

Function: This module contains the text segments used to build all messages issued by the DDR message writer (IGFDM0).

Called routines: None.

IGFDM0 — DDR Message Writer Routine

Function: This module issues a DDR message to inform the operator of the status of DDR, to prompt the operator for information necessary for the swap to continue, or to inform the operator of the status of the swap.

Entry from: IGFDD0, IGFDT0, IGFDU0, IGFDV0.

Exit to: Caller.

Called routines: IGFDE0.

IGFDM1 — DDR I/O Module

Function: This module performs all DDR I/O, except writing console messages.

Entry from: IGFDV0.

Exit to: Caller.

Called routines: IGFDE0.

IGFDR0 — DDR Recorder

Function: This module formats DDR LOGREC records and requests LOGREC recording.

Entry from: IGFDV0.

Exit to: Caller.

Called routines: IGFDE0, IGFDM1.

IGFDS0 — DDR Service Request Block (SRB) Routine

Function: This module creates the DDR task.

Entry from: Dispatcher, SRB scheduled by IGF2503D, IGE0660A, IGFDI0.

Secondary entry point DS0FRR, entered by the recovery termination manager when IGFDS0 terminates abnormally.

Exit to: Caller.

Called routines: None.

IGFDS1 — DDR Attacher

Function: This module issues an ATTACH macro instruction to give control to IGFDI0, the DDR task initiator.

Entry from: Stage 2 exit effector via an SRB.

Exit to: Caller.

Called routines: IGFDE0.

IGFDT0 — Tape Select Routine

Function: This routine processes a swap request for a magnetic tape device.

Entry from: IGFDI0.

Exit to: Caller.

Called routines: IGFDT1, IGFDV0, IGFDM0, IGFDE0, IGFDV1.

IGFDT1 — DDR Tape Validate Routine

Function: This module validates the tape device involved in a tape swap.

Entry from: IGE0660A, IGFDT0, IGFDV0.

Exit to: Caller.

Called routines: IGFDE0.

IGFDT2 — DDR Tape Swap Routine

Function: This module repositions the tape device involved in a tape swap.

Entry from: IGFDI0.

Exit to: Caller.

Called routines: IGFDE0, IGFDM0, DDR Device-dependent routine.

IGFDU0 — Unit Record Select Routine

Function: This routine processes a swap request for a unit record device.

Entry from: IGFDI0.

Exit to: Caller.

Called routines: IGFDV0, IGFDE0, IGFDM0.

IGFDU0 – Unit Record Select Routine

Function: This routine processes a swap request for a unit record device.

Entry from: IGFDI0.

Exit to: Caller.

Called routines: IGFDV0, IGFDE0, IGFDM0.

IGFDU1 – DDR Unit Record Validate Routine

Function: This module validates the unit record devices involved in a unit record swap.

Entry from: IGFDU0, IGFDV0.

Exit to: Caller.

Called routines: IGFDV1, IGFDE0.

IGFDV0 – DDR Common Select Routine

Function: This routine calls the appropriate routine to determine if the devices can be swapped, asks the operator for a device or gets approval of the one already selected, and controls the actual device swap.

Entry from: IGFDD0, IGFDT0, IGFDU0.

Exit to: Caller.

Called routines: IGFDE0, IGFDM0, IGFDM1, IGFDL1, IGFDD1, IGFDT1, IGFDU1, IGFDR0, IGFDW0.

IGFDV1 – DDR Device Independent Validate

Function: This module determines whether the devices involved in the DDR swap are valid and compatible, and whether IOS has purged the user's swap request.

Entry from: IGFDT0, IGFDD1, IGFDU1.

Exit to: Caller.

Called routines: IGFDE0, IEEVDEV.

IGFDW0 – DDR Unit Control Block (UCB) Swap

Function: This module swaps the physical unit control blocks of the failing and replacement devices by invoking IOSVSWAP.

Entry from: IGFDV0.

Exit to: Caller.

Called routines: IGFDE0, IGFDR0, IGFDM1, IOSVSWAP.

IGFPBUCR – Machine Check Control Block Allocation and Control Register Initialization

Function: This module allocates machine check control blocks during initialization and CONFIG CPU on-line; and frees machine check control blocks during CONFIG CPU off-line. For all entries it updates control registers 14 and 15.

Called routines: IGFPEXIT.

Entry From IGFRIM00 during initialization, IEEVWKUP during CONFIG CPU on-line.

Exit to: Caller.

IGFPEMER – Machine Check Handler Emergency Console and SYS1.LOGRRC Writer

Function: This module sends the message passed to IGFPTERM to the operator and writes the LOGREC buffer to SYS1.LOGREC. They both contain information on why the system is being terminated.

Entry from: IGFPTSIG.

Exit to: Caller.

Called routines: IGFPOFF.

IGFPEXIT – Machine Check Handler Exit Routine

Function: This module performs processor-related initialization functions when entered from IGFPBUCR at system initialization time and CONFIG CPU on-line. For all entries it updates control registers 14 and 15.

Entry from: IGFPMRTM, IGFPBUCR, IGFP2603D.

Exit to:

IEAVTRTN for hard machine checks; interrupted process for soft machine checks.

The caller of IGFPBUCR (IEEVWKUP, IGFRIM00).

External FLIH.

IGFPKREF – Storage Key Refresh Routine

Function: This module provides an interface which enables a storage key to be refreshed. It allows continued system execution when a machine check occurs due to an uncorrected storage key error.

Entry from: IARXMCKS.

Exit to: Caller.

Called routines: IGFPMFRS.

IGFPMAIN – Machine Check First Level Interrupt Handler (DAT-ON)

Function: This module performs machine-check recursion analysis and routes control to the various error analysis subroutines.

Entry from: IGFPMCIH in the DAT OFF nucleus.

Secondary entry-point IGFPMCPC: At this entry point the module intercepts program checks that occur while IGFPMAIN is executing.

Secondary entry-point IGFPMCRI: The module uses this entry-point when a restart interruption occurs while IGFPMAIN is executing.

Exit to: IGFPTERM for processor termination, IGFPMRTM to continue machine check processing.

Called routines: IGFPMHCA, IGFPMKTA, IGFPMSCA.

IGFPMCIH – Machine Check First Level Interrupt Handler (DAT-OFF)

Function: This module saves the registers associated with the interrupted process in the LOGREC buffer. System trace is suspended. IGFPMAIN, in the DAT ON nucleus is invoked to continue processing the machine check.

Entry from: The hardware loads the machine check new PSW.

Exit to: IGFPMAIN for normal machine check processing. IGFPWAIT to load a X'A28' wait state if unable to pass control to IGFPMAIN.

IGFPMFRS – Functional Recovery Routine Stacker

Function: This module allows machine check handler modules to establish and use internal recovery routines in an attempt to recover from a machine check, a program check, or a restart interruption that occurs while they are executing.

Entry from: IGFPMHCA, IGFPMKTA, IGFPMMSG, IGFPMTHA, IGFPMSCA, IGFPKREF.

Secondary entry point IGFPMSUP This is the super FRR associated with the MCH FRR stack which IGFPMRTM temporarily makes the current stack.

Entry from: RTM.

Exit to: Caller.

Called routines: IEAVEABD.

IGFPMHCA – Machine Check Handler Hard Machine Check Analysis

Function: This module analyzes the machine check interrupt code (MCIC) to determine if hard machine check conditions are present. If present, it sets indicators in the machine check error indicator area of the LOGREC buffer for subsequent analysis, and schedules message IGF971I or IGF972I.

Entry from: IGF970I and IGFPMAIN.

Exit to: Caller.

Called routines: IGFPMFRS, IGFPMTHA, IGFPMMSG, IEAVVMCH.

IGFPMKTA – Machine Check Handler Clock Timer Analysis

Function: This module analyzes the machine check interrupt code (MCIC) to determine if clock and timer damage conditions are present. If it finds an error in the time-of-day (TOD) clock, clock comparator, or processor timer, it sets an indicator in the machine check error indicator area of the LOGREC buffer. If damage has occurred, this module calls IGFPMTHA to perform threshold analysis.

Entry from: IGFPMAIN.

Exit to: Caller.

Called routines: IGFPMTHA, IGFPMFRS.

IGFPMMSG – Machine Check Handler Message Scheduler

Function: This module locates an available message buffer in the message buffer pool, and prepares it for subsequent processing by the machine check asynchronous message writer (IGFPWMSG).

Entry from: IGFPMHCA.

Exit to: Caller.

Called routines: IGFPMFRS.

IGFPMRTH – Machine Check Analyzer Routine

Function: This module interfaces with hardware repair routines and records an MCH LOGREC buffer to SYS1.LOGREC for all CALLRTM TYPE=MACHCK entries. The module also accumulates data describing the error and its repair status for inclusion in the recovery routine interface.

Entry from: IEAVTRTM or IEAVTACR

Exit to: Caller.

IGFPMRTM – Machine Check Handler Recovery Termination Management Interface

Function: This module prepares the LOGREC buffer for subsequent recording on SYS1.LOGREC and invokes IGFPMRTH via the CALLRTM macro to determine whether the machine check is hard or soft.

Entry from: IGFPMAIN.

Exit to: IGFPEXIT.

Called routines: IGFPMRTH (via CALLRTM), IEAVETSC.

IGFPMSCA – Machine Check Handler Soft Machine Check Analysis

Function: This module analyzes the machine check interrupt code (MCIC) to determine if soft machine check conditions (system recovery condition or degradation error) are present. If so, it sets indicators in the machine check error indicator area of the LOGREC buffer for subsequent analysis. If a system recovery condition, degradation error, or external damage is present, this module calls IGFPMTHA to perform threshold analysis.

Entry from: IGFPMAIN.

Exit to: Caller.

Called routines: IGFPMFRS, IGFPMTHA, IOSRMCH.

IGFPMTHA – Machine Check Threshold Analysis

Function: This module counts recoverable storage retry errors on soft machine checks, and compares this count with the threshold values contained in IGFTHB. When the threshold is reached, it schedules an operator message. IGFPMTHA also analyzes the following hard machine checks: instruction processing damage, system damage, an invalid PSW or register, time-of-day clock damage, processor timer damage, and clock comparator damage. When the threshold is reached, this module informs the caller, which then invokes IGFPTERM to perform processor termination.

Entry from: IGFPMSCA, IGFPMKTA, IGFPMHCA.

Exit to: Caller.

Called routines: IGFPMMSG, IGFPMFRS.

IGFPOFF – Machine Check Handler DAT OFF Service Routine

Function: This module searches IEEVG800 for the CCWs that are needed by IGFPEMER to send the message to the operator.

Entry from: IGFPEMER in the DAT ON nucleus.

Exit to: Caller.

IGFPTAIM – Machine Check Handler Alternate CPU Recovery Interface

Function: This module initiates alternate CPU recovery (ACR) by giving control to IEAVTACR.

Entry from: IGFPTSIG, IGFPXMFA.

Exit to: IEAVTACR.

IGFPTERM – MCH Processor/System Termination

Function: This module performs either system termination or processor termination. If an unrecoverable machine check occurs, it puts the executing processor in a stop state. If system termination is requested, it puts all on-line processors in disabled wait states.

Entry from: IGFPMAIN, ILRMSG00, IEAVESPR, IEAVTACR, IEAVTCR1, IEAVNIPM, IEAVTRS0, IEAVN700, IEEVLDWT, IEFAB4I0, IGVVSCEL, IOSRMCH.

Exit to: IEEVSTOP, IGFPTSIG.

IGFPTSIG – Emergency Signal (EMS) External Second Level Interrupt Handler

Function: During processor termination, this module invokes alternate CPU recovery (IEAVTACR) via IGFPTAIM. During system termination, it invokes the emergency recorder and console writer routine, and puts the system in a disabled wait state.

Entry from:

IGFPTERM for processor or system termination initiated by the machine check handler.

IGFPXMFA when a malfunction alert is received.

IEAVEES when an emergency signal is received.

Exit to: IGFPTAIM or enters wait state.

Called routines: IGFPEMER.

IGFPWAIT – DAT OFF Processor Termination

Function: This module places the executing processor into a disabled wait state and attempts to invoke IGFPTSIG on all other processors. The wait state code is passed as input from IGFPWAIT's caller.

Entry from: IGFPMCIH, IEAVEPC.

Exit to: None, the processor is placed into a disabled wait state.

IGFPWMSG – Machine Check Handler Asynchronous Message Writer

Function: This module writes an operator message scheduled by the mainline machine check handler (IGFPMMSG). It executes asynchronously to the mainline machine check handler and operates under the master scheduler task.

Entry from: IEAVTRET.

Exit to: Caller.

IGFPXMFA – Malfunction Alert External First Level Interrupt Handler

Function: This module receives control as an result of a malfunction alert generated when a processor enters a check/stop state. processor.

Entry from: IEAVEEXT.

Exit to: IGFPTSIG or IGFPTAIM.

Called routines: IEEVDCCR.

IGF2503D – SWAP Command Processor

Function: This module processes all MODE commands.

Entry from: IEE0403D.

Exit to: IEE0003D, IEE0503D.

IGF2603D – MODE Command Processor

Function: This module processes all MODE commands.

Entry from: IEE0403D.

Exit to: Caller, IEE0503D.

IGVAAQAT – Adjustment/Check Routine for AQAT Allocation Bits

Function: This module turns on the AQAT allocation bits, checks these bits, and turns them off for both SQA and LSQA subpools. If necessary, it builds new AQATs for LSQA.

Entry-Point IGVEAAQA: Turns on the AQAT allocation bits for SQA subpools.

Callers: IEAIP04, IEAVNP08, IGVFSQA, IGVGSQA, IGVRLSQA, IGVRSQA.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVEBAQA: Turns on the AQAT allocation bits and, if necessary, builds AQATs for the LSQA subpools.

Callers: IGVGLSQA, IGVRDFE.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVECAQA: Checks the AQAT allocation bits for both SQA and LSQA subpools.

Callers: IGVFLSQA, IGVFSQA, IGVRDFE, IGVRSRTN.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVEUAQA: Turns off the AQAT allocation bits for both SQA and LSQA subpools.

Callers: IGVEBAQA, IGVFLSQA, IGVFSQA, IGVGSQA, IGVRLSQA, IGVRSQA.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IARVF, IARVG.

IGVADFE – DFE Allocation Routine

Function: This module removes storage from the double free element (DFE) size and address queues.

Entry-Point IGVEADFE:

Callers: IGVFLSQA, IGVFSQA, IGVGLSQA, IGVGSQA, IGVRLSQA, IGVRSQA.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: None.

IGVBDFE – DFE Build Routine

Function: This module returns storage to the double free element (DFE) address and size queues.

Entry-Point IGVEBDFE:

Callers: IGVFLSQA, IGVFSQA, IGVGLSQA, IGVGSQA, IGVRDFE, IGVRLSQA, IGVRSQA.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: None.

IGVCPBLD – Cell Pool Build Routine

Function: This module builds a cell pool when the user issues a CPOOL BUILD macro.

Entry-Point IGVCPBDP: Builds a cell pool when control is received through program call (PC) linkage.

Callers: Users of the CPOOL BUILD macro that specify LINKAGE=SYSTEM.

Exit-Normal: Returns to the caller.

Exit-Error: X'C78' ABEND codes.

Entry-Point IGVCPBDB: Builds a cell pool when control is received through branch entry linkage.

Callers: Users of the CPOOL BUILD macro that specify LINKAGE=BRANCH.

Exit-Normal: Returns to the caller.

Exit-Error: X'C78' ABEND codes.

External-References: None.

IGVCPDEL – Cell Pool Deletion Routine

Function: This module deletes a cell pool when the user issues a CPOOL DELETE macro.

Entry-Point IGVCPDLP: Deletes a cell pool when control is received through program call (PC) linkage.

Callers: Users of the CPOOL DELETE macro that specify LINKAGE=SYSTEM.

Exit-Normal: Returns to the caller.

Exit-Error: X'C78' ND codes.

Entry-Point IGVCPDLB: Deletes a cell pool when control is received through branch entry linkage.

Callers: Users of the CPOOL DELETE macro that specify LINKAGE=BRANCH.

Exit-Normal: Returns to the caller.

Exit-Error: X'C78' ABEND codes.

External-References: None.

IGVCPEXT – Cell Pool Extension Routine

Function: This module extends an existing cell pool by adding more cells to the cell pool.

Entry-Point IGVCPEXP: Extends a cell pool when control is received through program call (PC) linkage.

Callers: IGVCPOOL calls IGVCPEXP to process requests for users of the CPOOL GET-UNCOND macro that specify LINKAGE=SYSTEM.

Exit-Normal: Returns to the caller.

Exit-Error: X'C78' ABEND codes.

Entry-Point IGVCPEXB: Extends a cell pool when control is received through branch entry linkage.

Callers: IGVCPOOL calls IGVCPEXB to process requests for users of the CPOOL GET-UNCOND macro that specify LINKAGE=BRANCH.

Exit-Normal: Returns to the caller.

Exit-Error: X'C78' ABEND codes.

External-References: None.

IGVCPOOL — Cell Pool GET and FREE Routine

Function: This module gets or frees cells from the VSM cell pool.

Entry-Point IGVCPGTC: Gets a cell from a cell pool for CPOOL GET conditional requests.

Callers: Users of the CPOOL GET macro.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVCPGUB: Gets a cell from a cell pool for CPOOL GET unconditional requests when LINKAGE = BRANCH is specified.

Callers: Users of the CPOOL GET macro.

Exit-Normal: Returns to the caller.

Exit-Error: X'C78' ABEND codes issued by entry-point IGVCPEXB in IGVCPEXT.

Entry-Point IGVCPGUP: Gets a cell from a cell pool for CPOOL GET unconditional requests when LINKAGE = SYSTEM and REGS = USE are specified.

Callers: Users of the CPOOL GET macro.

Exit-Normal: Returns to the caller.

Exit-Error: X'C78' ABEND codes issued by entry-point IGVCPEXP in IGVCPEXT.

Entry-Point IGVCPGUS: Gets a cell from a cell pool for CPOOL GET unconditional requests when LINKAGE = SYSTEM and REGS = SAVE is specified and saves the secondary ASID in register 15.

Callers: Users of the CPOOL GET macro.

Exit-Normal: Returns to the caller.

Exit-Error: X'C78' ABEND codes issued by entry-point IGVCPEXP in IGVCPEXT.

Entry-Point IGVCPFRE: Returns a cell to a cell pool.

Callers: Users of the CPOOL FREE macro.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IGVCPEXT.

IGVDAQAT — AQAT Deletion Routine for Local AQAT Stacks

Function: This module removes all address queue anchor tables (AQATs) that describe only free storage from the local AQAT stack.

Entry-Point IGVDAQAT:

Callers: IGVGRRGN, IGVGVRGN.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IARVF.

IGVFAPVT — Private Area FREEMAIN Routine for Authorized Region Subpools

Function: This module frees the storage in authorized subpools 229, 230, 236, and 237.

Entry-Point IGVEFAUK: Frees the storage in authorized user key subpools 229 and 230.

Callers: IGVVSMRF, IGVRPVT.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVEFSWA: Frees the storage in scheduler work area subpools 236 and 237.

Callers: IGVVSMRF, IGVRPVT.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IGVFDQE, IGVFSDQE.

IGVFCSA — FREEMAIN Service Routine for CSA Subpools

Function: This module frees the storage in the common services area (CSA) subpools 227, 228, 231, and 241.

Entry-Point IGVEFCSA:

Callers: IGVRCSA, IGVVSMRF, IGVVSMRT.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IARVF, IGVFSDQE, IGVFSFBQ.

IGVFDQE — FREEMAIN Service Routine for DQE Storage

Function: This module returns private area storage to the free block queue element (FBQE) queue and calls real storage management (RSM) to release the real storage backing it.

Entry-Point IGVEFDQE:

Callers:

- IGVFAPVT
- IGVFPVT
- IGVFSP
- IGVRFSP
- IGVRPVT

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References:

- IARVFRMN (entry point in IARVF)
- IARVPGTI (entry point in IARVP)
- IGVEFSFB (entry point in IGVFSFBQ)

IGVFLSQA — FREEMAIN Service Routine for LSQA Subpools

Function: This module frees storage in local system queue area (LSQA) subpools 233, 234, 235, 253, 254, and 255.

Entry-Point IGVEFLSQ:

Callers: IGVFSP, IGVRLSQA, IGVVSMRF.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IARVF, IGVAAQAT, IGVADFE, IGVBDFE, IGVFSFBQ.

IGVFPVT — Private Area FREEMAIN Routine for User Region Subpools

Function: This module frees the storage in user region subpools 0-127, 240, 250, 251, and 252.

Entry-Point IGVEFPVT:

Callers: IGVVSMRF, IGVRPVT.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IGVFDQE, IGVFSDQE.

IGVFRRGN — Free Real Region Routine

Function: This module frees a real region at the termination of a job step by resetting the V = R control blocks.

Entry-Point IGVFRRGN:

Callers: IGVGRRGN, IGVGVRGN.

Exit-Normal: Returns to the caller.

Exit-Error: X'C78' ABEND code.

External-References: IARXE, IGVFSFBQ.

IGVFSDQE — FREEMAIN DQE/FQE Queue Search Routine

Function: This module frees the storage from the input subpool and returns it to the DQE/FQE queue.

Entry-Point IGVEFSDQ:

Callers: IGVFAPVT, IGVFCSA, IGVFPVT.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: None.

IGVFSFBQ — FREEMAIN FBQE Search Routine

Function: This module returns one or more pages to the free block queue element (FBQE) queue.

Entry-Point IGVEFSFB:

Callers: IGVFCSA, IGVFDQE, IGVFLSQA, IGVFRRGN, IGVFSQA, IGVFVIRT, IGVFVRGN, IGVGLSQA, IGVRCSA, IGVRLSQA, IGVRSQA.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: None.

IGVFSP — Subpool FREEMAIN Routine

Function: This module frees the storage for the entire subpool specified in the FREEMAIN request associated with the input task control block (TCB).

Entry-Point IGVEFSP:

Callers: IGVVSMRF.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References:

- IGVEFDQE (entry point in IGVFDQE)
- IGVEFLSQ (entry point in IGVFLSQA)
- IGVVSGTC (entry point in IGVVSCEL)

IGVFSQA – FREEMAIN Service Routine for SQA Subpools

Function: This module frees storage in system queue area (SQA) subpools 226, 239, and 245.

Entry-Point IGVEFSQA:

Callers: IGVRSQA, IGVVSMRF, IGVVSMRT.

Exit-Normal: Returns to the caller.

Exit-Error: ABEND codes: 7xx-01 and 7xx-09.

External-References: IARVF, IGVAAQAT, IGVADFE, IGVBDFE, IGVFSFBQ.

IGVFVIRT – Deferred Release Queue FREEMAIN Routine

Function: This module frees the cells on the deferred release queue and returns them to the VSM cell pool. It then returns the virtual storage to the common service area (CSA) free block queue element (FBQE) queue.

Callers: RSM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVRVIRT: Processes recovery for entry-point IGVFVIRT.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

External-References: IGVFSFBQ.

IGVFVRGN – Free Virtual Region Routine

Function: This module frees a virtual region at the termination of a job step by resetting the virtual address space control blocks.

Entry-Point IGVFVRGN:

Callers: IGVGRRGN, IGVGVRGN.

Exit-Normal: Returns to the caller.

Exit-Error: X'C78' ABEND code.

External-References:

- IARUSLOT (entry point in IARUS)
- IGVEFSFB (entry point in IGVFSFBQ)

IGVGAPVT – Private Area GETMAIN Routine for Authorized Region Subpools

Function: This module allocates storage for authorized region subpools 229, 230, 236, and 237.

Entry-Point IGVEGAUK: Allocates storage from the authorized region subpools 229 and 230.

Callers: IGVVSMRT.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVEGSWA: Allocates storage from the scheduler work area subpools 236 and 237.

Callers: IGVVSMRT.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IARVC, IARVF, IARVP.

IGVGCAS Create/Terminate Address Space Routine

Function: This module creates or terminates an address space.

Entry-Point IGVGCAS: Creates an address space.

Callers: IEAVEMCR (Supervisor).

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVGFAS: Posts waiting initiators so that other V=R jobs can start.

Callers: IEAVTMMT (Supervisor).

Exit-Normal: Returns to the caller.

Exit-Error: Returns to the caller.

Entry-Point IGVRGCAS: Processes recovery for entry-point IGVGCAS.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

External-References:

- IARMASCR (entry point in IARMA)
- SDIR (size queue anchor table directory)

- SQ55 (size queue anchor table template)

IGVGCSA — GETMAIN Service Routine for CSA Subpools

Function: This module allocates storage for CSA subpools 227, 228, 231, and 241.

Entry-Point IGVEGCSA:

Callers: IGVVSMRT.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IARVG, IARVP.

IGVGLSQA — GETMAIN Service Routine for LSQA Subpools

Function: This module allocates storage from the local system queue area (LSQA) subpools 233, 234, 235, 253, 254, and 255.

Entry-Point IGVEGLSQ:

Callers: IGVVSCEL, IGVVSMRT.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IARVC, IARVF, IARVG, IGVAAQAT, IGVADFE, IGVBDFE, IGVFSFBQ.

IGVGPVT — Private Area GETMAIN Routine for User Region Subpools

Function: This module allocates storage for user region subpools 0-127, 240, 250, 251, and 252.

Entry-Point IGVEGPVT:

Callers: IGVVSMRT.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IARVP.

IGVGRRGN — Get Real Region Routine

Function: This module initializes a V = R region.

Entry-Point IGVGRRGN:

Callers: IEFSD263.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVRGNR: Performs recovery from ABENDs during the V = R region processing.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

External-References: IARXE, IGVDAQAT, IGVFRRGN, IGVFVRGN, IGVLIMIT.

IGVGSQA — GETMAIN Service Routine for SQA Subpools

Function: This module allocates storage from the system queue area (SQA) subpools 226, 239, and 245.

Entry-Point IGVEGSQA:

Callers: IGVVSCEL, IGVVSMRT.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IARVG, IGVAAQAT, IGVADFE, IGVBDFE.

IGVGVRGN — Get Virtual Region Routine

Function: This module initializes a V = V region.

Entry-Point IGVGVRGN:

Callers:

- IEAVNIP0 (Nucleus initialization)
- IEEPRWI2 (Started task control)
- IEFIB612 (Scheduler restart)
- IEFSD263 (Initiator ATTACH routine)

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVRVRGN: Processes recovery for the get virtual region processing.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

External-References:

- IGVDAQAT (entry point in IGVDAQAT)
- IGVFRRGN (entry point in IGVFRRGN)
- IGVFVRGN (entry point in IGVFVRGN)
- IGVLIMIT (default region addressing mode interface routine)

IGVLIMIT – Default Region Addressing Mode Interface Routine

Function: This module invokes the IEALIMIT routine which operates in 24-bit addressing mode.

Entry-Point IGVLIMIT:

Callers: IGVGRRGN, IGVGVRGN.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IEALIMIT.

IGVLOCP – VSMLOC Recovery and Serialization Routine

Function: This module provides recovery and serialization for the users of VSMLOC and initializes the input needed for IGVSLOC to perform storage allocation verification.

Entry-Point IGVLOCP:

Callers: Users of the VSMLOC macro that specify LINKAGE=SYSTEM.

Exit-Normal: Returns to the caller.

Exit-Error: X'C78' ABEND code.

Entry-Point IGVRLOC: Processes recovery for entry-point IGVLOCP.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

External-References:

- IGVSLOC (entry point in IGVSLOC)

IGVRCP – Cell Pool Functional Recovery Routine

Function: This module performs recovery from ABENDs that occur in modules IGVCPBLD, IGVCPDEL, or IGVCPEXT.

Entry-Point IGVERCP:

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

Entry-Point IGVERRCP: Processes recovery for entry-point IGVERCP.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

External-References: None.

IGVRCSA – CSA Functional Recovery Routine

Function: This module recovers from ABENDs that occur during the processing of GETMAIN/FREEMAIN requests for CSA subpools.

Entry-Point IGVERCSA:

Callers: IGVRVSM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IARVF, IGVFCSA, IGVFSFBQ, IGVRSRCH.

IGVRDBL – Double-Threaded Queue Verification Routine

Function: This module verifies a double-threaded queue and repairs the queue if an error is found.

Entry-Point IGVERDBL:

Callers: IGVRDFE, IGVRQVR1, IGVRQVR2.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IGVRSRTN.

IGVRDFE – DFE Queue Verification Routine

Function: This module verifies the DFE queue that was being processed when an ABEND occurred.

Entry-Point IGVERDFE:

Callers: IGVRVSM.

Exit-Normal: Returns to the caller.

Exit-Error:

- MEMTERM is scheduled for the address space.
- The field RCWKPERC in the recovery work area (RCWK) is set to force percolation.
- The field RCWKBACK in the recovery work area (RCWK) is set to prevent backout processing.

External-References: IARVC, IARVF, IARVG, IGVAAQAT, IGVBDFE, IGVRDBL, IGVVSCEL.

IGVRFSP – Subpool FREEMAIN Functional Recovery Routine

Function: This module continues with the processing of a subpool FREEMAIN request after an ABEND has occurred.

Entry-Point IGVERFSP:

Callers: IGVRVSM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References:

- IGVEFDQE (entry point in IGVFDQE)
- IGVERLSQ (entry poing in IGVRLSQA)
- IGVERSFB (entry point in IGVRSRCH)
- IGVERSFQ (entry point in IGVRSRCH)

IGVRLSQA – LSQA Functional Recovery Routine

Function: This module recovers from ABENDs that occur during the processing of GETMAIN/FREEMAIN requests for LSQA subpools.

Entry-Point IGVERLSQ:

Callers: IGVRFSP, IGVRVSM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IARVF, IGVAAQAT, IGVADFE, IGVBDFE, IGVFLSQA, IGVFSFBQ, IGVRSRCH.

IGVRPVT – Private Area Functional Recovery Routine

Function: This module performs backout for private area GETMAIN requests, and performs recovery to continue the private area FREEMAIN request.

Entry-Point IGVERPVT:

Callers: IGVRVSM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References:

- IARVFRMN (entry point in IARVF)
- IGVEFAUK (entry point in IGVFAPVT)
- IGVEFSWA (entry point in IGVFAPVT)
- IGVEFDQE (entry point in IGVFDQE)
- IGVEFPVT (entry point in IGVFPVT)
- IGVERSFB (entry point in IGVRSRCH)
- IGVERSFQ (entry point in IGVRSRCH)

IGVRQVR1 – Queue Verification Routine #1

Function: This module verifies the descriptor queue element (DQE) queue, the free queue element (FQE) queue, and subpool queue element (SPQE) queue, and the subpool table (SPT), and, if necessary, corrects any errors that it finds.

Entry-Point IGVERDQE: Verifies the descriptor queue element (DQE) queue and corrects any errors it finds.

Callers: IGVERSPQ (Entry-point in IGVRQVR1), IGVERSPT (Entry-point in IGVRQVR1).

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVERFQE: Verifies the free queue element (FQE) queue and corrects any errors that it finds.

Callers: IGVERDQE (Entry-point in IGVRQVR1).

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVERSPQ: Verifies the subpool queue element (SPQE) queue and corrects any errors that it finds.

Callers: IGVRVSM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVERSPT: Verifies the subpool table (SPT) and corrects any errors that it finds.

Callers: IGVRVSM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IGVRDBL, IGVRSRCH, IGVRSRTN.

IGVRQVR2 – Queue Verification Routine #2

Function: This module verifies the allocated element (AE) queue, the VSM cell pool, and the free block queue element (FBQE) queue.

Entry-Point IGVERAE: Verifies the allocated element (AE) queue and corrects any errors that it finds.

Callers: IGVRVSM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVERCPL: Verifies the VSM cell pool and corrects any errors that it finds.

Callers: IGVRVSM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVERFBQ: Verifies the free block queue element (FBQE) queue and corrects any errors that it finds.

Callers: IGVRVSM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IGVRDBL, IGVRSRTN.

IGVRRTR – Router Functional Recovery Routine

Function: This module recovers from processing performed in the router modules IGVVSMRF, IGVVSMRT, IGVVSM24, IGVVSM31 and IGVVSTOR. IGVRRTR also recovers from processing performed in the cell pool management module, IGVVSCEL, and the ABEND generation module, IGVVSERR.

Entry-Point IGVERRTR: Processes recovery for the router modules IGVVSM24, IGVVSM31, IGVVSMRF, IGVVSMRT, and IGVVSTOR.

Callers: IGVRVSM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVERCEL: Processes recovery for IGVVSCEL.

Callers: IGVRVSM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVERERR: Processes recovery for IGVVSERR.

Callers: IGVRVSM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IGVRSRTN, IGVVSCEL, IGVVSMRF.

IGVRSQA – SQA Functional Recovery Routine

Function: This module recovers from ABENDs that occur during the processing of requests for SQA subpools.

Entry-Point IGVERSQA:

Callers: IGVRVSM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IARVF, IGVAAQAT, IGVADFE, IGVBDFE, IGVFSFBQ, IGVFSQA, IGVRSRCH.

IGVRSRCH – Queue Search and IGVFSDQE Functional Recovery Routine

Function: This module searches either an FBQE queue, a DQE/FQE queue, or a DFE queue to determine if the specified storage is represented by an element on the queue. It also performs recovery for IGVFSDQE.

Entry-Point IGVERSFB: Searches the FBQE queue to determine if the specified storage is represented by an FBQE.

Callers: IGVRCSA, IGVRFSP, IGVRLSQA, IGVRPVT, IGVRSQA.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVERSFQ: Searches the DQE/FQE queues to determine if the specified storage is represented by either a DQE or an FQE.

Callers: IGVRCSA, IGVRFSP, IGVRPVT.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVERSDF: Searches the DFE address queue to determine if the specified storage is represented by a DFE.

Callers: IGVRLSQA, IGVRSQA.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVERFSD: Performs recovery for IGVFSDQE.

Callers: IGVRQVR1.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: None.

IGVRSRTN – Functional Recovery Service Routine

Function: This module verifies that an area is in the proper storage, verifies that a single-threaded queue is noncircular, and, if necessary, produces a dump.

Entry-Point IGVERADR: Verifies that an area is in the proper storage.

Callers: IGVRDBL, IGVRQVR1, IGVRQVR2, IGVRRTR.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVERSNG: Verifies that a single-threaded queue is noncircular.

Callers: IGVRQVR1, IGVRQVR2.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVERDMP: If it is necessary, produces a dump.

Callers: IGVRVSM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IGVAAQAT.

IGVRVSM – Functional Recovery Routine for GETMAIN/FREEMAIN and STORAGE Modules

Function: This module is the functional recovery routine (FRR) for all GETMAIN, FREEMAIN and STORAGE functions. It performs recovery, and, if possible, attempts retry.

Entry-Point IGVERVSM: See module function.

Callers: RTM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVERFRR: Performs recovery for ABENDs occurring in IGVRVSM.

Callers: RTM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVERLCK: Releases all locks and initializes the registers for retry.

Callers: RTM.

Exit-Normal: Branches to the retry point in VSM.

Exit-Error: No exit error conditions.

External-References:

- IGVERCSA (entry point in IGVRCSA)
- IGVERDFE (entry point in IGVRDFE)
- IGVERFSP (entry point in IGVRFSP)
- IGVERLSQ (entry point in IGVRLSQA)
- IGVERPVT (entry point in IGVRPVT)
- IGVERSPQ (entry point in IGVRQVR1)
- IGVERSPT (entry point in IGVRQVR1)
- IGVERAE (entry point in IGVRQVR2)
- IGVERCPL (entry point in IGVRQVR2)
- IGVERFBQ (entry point in IGVRQVR2)
- IGVERCEL (entry point in IGVRRTR)
- IGVERERR (entry point in IGVRRTR)
- IGVERRTR (entry point in IGVRRTR)
- IGVERSQA (entry point in IGVRSQA)
- IGVERDMP (entry point in IGVRSRTN)
- IGVVSEND (entry point in IGVVSERR)

IGVSFGLB – Format Global Data Routine

Function: This module accesses, formats, and prints the global VSM control blocks.

Entry-Point IGVSFGLB:

Callers: IGVSFMAN.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: AMDPRUIM, IGVSFSVC.

IGVSFLCL – Format Local Data Routine

Function: This module accesses, formats, and prints the local VSM (private area) control blocks.

Entry-Point IGVSFLCL:

Callers: IGVSFMAN.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: AMDPRUIM, IGVSFSVC.

IGVSFMAN – Mainline Format Routine

Function: This module obtains the global data area (GDA) and the address space control blocks (ASCBs) to format specific global and address space data.

Entry-Point IGVSFMAN:

Callers: Print dump and IPCS.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: BLSQROUT, IGVSFGLB, IGVSFLCL, IGVSFPAR.

IGVSFPAR – VSMDATA Parse Routine

Function: This module parses the keywords on the VSMDATA verb.

Entry point IGVSFPAR:

Callers: IGVSFMAN.

Exit-Normal: Returns to caller.

Exit-Error: No exit error conditions.

External-References: BLSQROUT, IKJEYFF19, IKJPARS.

IGVSFSVC – Print Dump Service Routines

Function: This module formats and prints the VSM control blocks.

Entry-Point IGVESQAT: Formats and prints the size queue anchor table (SQAT), the DFE size queue, and the DFE address queue supplied as input.

Callers: IGVSFGLB, IGVSFLCL.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVEAQIN: Formats and prints the address queue anchor table (AQAT) entries that are active (that is, that represent allocated storage).

Callers: IGVSFGLB, IGVSFLCL.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVERD: Formats and prints the region descriptor (RD) supplied as input and the FBQEs that are anchored in the RD.

Callers: IGVSFGLB, IGVSFLCL.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVESPQA: Formats and prints the queue anchors (either the SPQA or the SPT entry) supplied as input, the DQEs anchored in the queue anchor, and the FQEs anchored in the DQEs.

Callers: IGVSFGLB, IGVSFLCL.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVEFBDF: Formats and prints the FBQEs on the deferred release queue.

Callers: IGVSFGLB, IGVSFLCL.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVEFSTD: Formats and prints the control blocks supplied as input.

- IGVSFGLB supplies the GDA, the fixed and pageable VSWK, the fixed and pageable GETMAIN/FREEMAIN communications area, the global cell pool headers, and the region request blocks (RGRs).
- IGVSFLCL supplies the LDA, the local VSWK, the local GETMAIN/FREEMAIN communications area, the local cell pool headers, the AEs, and the SPQEs.

Callers: IGVSFGLB, IGVSFLCL.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: AMDPRUIM.

IGVSLIST – VSMLIST Verification Routine

Function: This module verifies that the parameters specified on the VSMLIST macro are valid and calls the appropriate entry point in IGVSLIS1 to process he VSMLIST request.

Entry-Point IGVLISTP: Processes program call (PC) entry requests to the VSMLIST routine.

Callers: Users of the VSMLIST macro that specify LINKAGE = SYSTEM.

Exit-Normal: Returns to the caller.

Exit-Error: X'C78' ABEND code.

Entry-Point IGVRLIST: Performs recovery for entry-point IGVLISTP.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

Entry-Point IGVRRLIS: Performs recovery for entry-point IGVRLIST.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

Entry-Point IGVSLIST: Processes branch entry requests to the VSMLIST routine.

Callers: Users of the VSMLIST macro that specify LINKAGE = BRANCH.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: IGVSLIS1.

IGVSLIS1 – VSMLIST Information Routine

Function: This module lists information describing the allocation of virtual storage for SQA, LSQA, CSA, or private area subpools, or for a user specified subpool list. This information is listed in the data area portion of the work area that was passed to IGVSLIST by its caller.

Entry-Point PROCAQAT: Lists information describing the allocation of virtual storage in either SQA or LSQA subpools.

Callers: IGVSLIST.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point PROCSPT: Lists information describing the allocation of virtual storage in the CSA subpools.

Callers: IGVSLIST.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point PROCTCB: Lists information describing the allocation of virtual storage in the private area subpools.

Callers: IGVSLIST.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point PROCRD: Lists unallocated virtual storage information for either CSA or private area regions.

Callers: IGVSLIST.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: None.

IGVSLOC – GETMAIN Verification Routine

Function: This module verifies that a storage area in the SQA, CSA, LSQA, or the private area has been allocated through a GETMAIN request, or a cell pool resides in a cell pool extent (primary or secondary).

Entry-Point IGVSLOC:

Callers: IGVLOCP, users of the VSMLOC macro that specify LINKAGE = BRANCH.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: None.

IGVSREGN – Region Descriptor Routine

Function: This module obtains the information contained in the region descriptor and the extended region descriptor and passes it to the caller.

Entry-Point IGVSREGN:

Callers: Users of the VSMREGN macro.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References: None.

IGVSTSKI – Task Initialization Routine

Function: This module performs subpool and task control block (TCB) processing as specified.

Entry-Point IGVSTSKI:

Callers: IEAVEAT0, Memory initialization.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVRSTSK: Performs recovery for entry-point IGVSTSKI.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

External-References:

- ITVVSPCK (entry point in ITVVSPCK)

IGVSTSKT – Task Termination Routine

Function: This module performs VSM task termination for a terminating task.

Entry-Point IGVQSPET:

Callers: IEAVEOR (Task &q1.address space&eq1. termination)

Exit-Normal: Returns to the caller.

Exit-Error: X'C78' ABEND code.

Entry-Point IGVRQSPE: Performs recovery for entry-point IGVQSPET.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: No exit error conditions.

External-References:

- IGVVSCBR (entry point in IGVVSCEL)

IGVTRACE

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Virtual Storage Management.*

IGVTRCTL

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Virtual Storage Management.*

IGVVSCAP – Addressing Mode Interface Routine

Function: IGVVSCAP is a capping module. It is the interface between callers operating in any addressing mode to GETMAIN/FREEMAIN routines operating in 31-bit addressing mode.

Entry-Point CBBRANCH: Forms an interface with IGVVSCEL to execute either a GETMAIN or a FREEMAIN request for a VSM control block. Passes control in 31-bit addressing mode.

Callers: Users of the macro: GETMAIN/FREEMAIN P BRANCH=YES.

Exit-Normal: Branches to IGVECBBR in IGVVSM24.

Exit-Error: No exit error conditions.

Entry-Point CRBRANCH: Local branch entry-point for SVC 120 requests. Passes control in 31-bit addressing mode to IGVVSM31.

Callers: Users of the macro: GETMAIN/FREEMAIN RC/RU/VRC/VRU BRANCH=YES.

Exit-Normal: Branches to IGVVSB31 in IGVVSM31.

Exit-Error: No exit error conditions.

Entry-Point FMBRANCH: Local branch entry-point for SVC 5 requests. Passes control in 31-bit addressing mode to IGVVSM24.

Callers: Users of the macro: FREEMAIN EC/EU/LC/LU/VC/VU BRANCH=YES.

Exit-Normal: Branches to IGVEFMBR in IGVVSM24.

Exit-Error: No exit error conditions.

Entry-Point GLBRANCH: Global branch entry-point for SVC 120 requests. Passes control in 31-bit addressing mode to IGVVSM31.

Callers: Users of the macro: GETMAIN/FREEMAIN RC/RU/VRC/VRU BRANCH=(YES,GLOBAL).

Exit-Normal: Branches to IGVVSGBR in IGVVSM31.

Exit-Error: No exit error conditions.

Entry-Point GMBRANCH: Local branch entry-point for SVC 4 requests. Passes control in 31-bit addressing mode to IGVVSM24.

Callers: Users of the macro: GETMAIN EC/EU/LC/LU/VC/VU BRANCH=YES.

Exit-Normal: Branches to IGVEGMBR in IGVVSM24.

Exit-Error: No exit error conditions.

Entry-Point RMBRANCH (or QCBRANCH): Local branch entry-point for SVC 10 requests. Passes control in 31-bit addressing mode to IGVVSM24.

Callers: Users of the macro: GETMAIN/FREEMAIN R BRANCH=YES.

Exit-Normal: Branches to IGVERMBR in IGVVSM24.

Exit-Error: No exit error conditions.

External-References: IGVVSM24, IGVVSM31.

IGVVSCEL – Cell Pool Management Routine

Function: This module either obtains or releases a cell, replenishes the cell pool or formats the cell pool block.

Entry-Point IGVVSCBR: Obtains or releases a cell.

Callers: IGVSTSKT, IGVVSM24.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVVSFMT: Formats a cell pool block.

Callers: IEAIPL04, IGVRRTR, IGVRDFE, IGVVSGTC (Entry-point in IGVVSCEL).

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point IGVVSGTC: Replenishes a cell pool.

Callers: IGVFSP, IGVVSCBR (entry-point in IGVVSCEL), IGVVSMRF, IGVVSMRT.

Exit-Normal: Returns to the caller.

Exit-Error: If there is not enough real or virtual storage for the SQA cell pool, it calls IGFPTERM to issue a message (IEA946W) and to terminate the system.

External-References: IGFPTERM, IGVGLSQA, IGVGSQA, IGVVSERR.

IGVVSERR – ABEND Generation Routine

Function: This module generates ABEND and reason codes and issues the ABEND for errors that occur during the processing of the GETMAIN and the FREEMAIN requests.

Entry-Point IGVVSABN: Generates ABEND and reason codes for IGVVSM24 and IGVVSM31.

Callers: IGVVSM24, IGVVSM31.

Exit-Normal: No exit normal conditions.

Exit-Error: No exit error conditions.

Entry-Point IGVVSEND: IGVVSEND is the entry-point for modules that have generated ABEND and reason codes. IGVVSEND releases the VSMFIX or the VSMPAG lock (if either was obtained), deletes the FRR, sets standard information in the message area, and issues an ABEND.

Callers: IGVRVSM, IGVVSCEL, IGVVSMRF, IFVVSMRT, IGVVSM24, IGVVSM31.

Exit-Normal: No exit normal conditions.

Exit-Error: IGVVSERR exits through EXIT-ERROR for all conditions.

External-References: None.

IGVVSMRF – FREEMAIN Router

Function: This module routes FREEMAIN requests to the appropriate service routines, sets a return code, and returns control to the caller. The subpool translation table (SPTT) entry corresponding to the input subpool identifier specifies the type of storage (SQA, CSA, LSQA, or Private) to be freed. The type of storage to be freed and the type of request (either FREEMAIN or subpool FREEMAIN) identifies the appropriate service routine that IGVVSMRF will call.

Entry-Point IGVVSMFR:

Callers: IGVRRTR, IGVVSM24, IGVVSM31.

Exit-Normal: Returns to the caller.

Exit-Error: See ABEND codes section.

External-References: IGVFAPVT, IGVFCSA, IGVFLSQA, IGVFPVT, IGVFSP, IGVFSQA, IGVVSCEL, IGVVSERR.

IGVVSMRT – GETMAIN Router

Function: This module routes GETMAIN requests to the appropriate service routine, sets a return code, and returns control to the caller. It returns the address and the size of the obtained area in the VSM work area (VSWK). The subpool translation table (SPTT) entry corresponding to the input subpool identifier specifies the type of storage (SQA, CSA, LSQA, or private area) to be obtained. This storage type identifies the appropriate service routine that IGVVSMRT will call.

Entry-Point IGVVSMGT:

Callers: IGVVSM24, IGVVSM31.

Exit-Normal: Returns to the caller.

Exit-Error: See ABEND codes section.

External-References: IGVFCSA, IGVFSQA, IGVGAPVT, IGVGCSA, IGVGLSQA, IGVGPVT, IGVGSQA, IGVVSCEL, IGVVSERR.

IGVVSM24 – Interface Routine for 24-Bit GETMAIN/FREEMAIN Requests

Function: This module routes SVC 4, SVC 5, SVC 10, their equivalent branch entries, GETMAIN P, and FREEMAIN P requests. It formats pertinent information about the requests and stores this information in the VSM work area (VSWK). Then, IGVVSM24 routes the requests to either IGVVSMRF (the FREEMAIN router), IGVVSMRT (the GETMAIN router), or IGVVSCEL (the cell pool management routine). When control is returned from the GETMAIN/FREEMAIN service routines, IGVVSM24 passes back a return code, and for GETMAIN requests, the address of the allocated storage.

Entry-Point IGC004: Routes SVC 4 entries from EC, EU, LC, LU, VC, or VU GETMAIN requests from a program executing in 24-bit addressing mode.

Callers: SVC 4.

Exit-Normal: Returns to the caller.

Exit-Error: Calls IGVVSERR to issue the requested ABEND.

Entry-Point IGC005: Routes SVC 5 entries from EC, EU, LC, LU, VC, or VU FREEMAIN requests from a program executing in 24-bit addressing mode.

Callers: SVC 5.

Exit-Normal: Returns to the caller.

Exit-Error: Calls IGVVSERR to issue the requested ABEND.

Entry-Point IGVEGMBR: Routes SVC 4 equivalent branch entries from EC, EU, LC, LU, VC, or VU GETMAIN request.

Callers: GMBRANCH (entry-point in IGVVSCAP).

Exit-Normal: Returns to the caller.

Exit-Error: Calls IGVVSERR to issue the requested ABEND.

Entry-Point IGVEFMBR: Routes SVC 5 equivalent branch entries from EC, EU, LC, LU, VC, or VU FREEMAIN requests.

Callers: FMBRANCH (entry-point in IGVVSCAP).

Exit-Normal: Returns to the caller.

Exit-Error: Calls IGVVSERR to issue the requested ABEND.

Entry-Point IGC010: Routes SVC 10 entries from either R GETMAIN or R FREEMAIN requests.

Callers: SVC 10.

Exit-Normal: Returns to the caller.

Exit-Error: Calls IGVVSERR to issue the requested ABEND.

Entry-Point IGVERMBR: Routes SVC 10 equivalent branch entries from either R GETMAIN or R FREEMAIN requests.

Callers: RMBRANCH (entry-point in IGVVSCAP), QCBRANCH (entry-point in IGVVSCAP).

Exit-Normal: Returns to the caller.

Exit-Error: Calls IGVVSERR to issue the requested ABEND.

Entry-Point IGVECBBR: Routes requests to get or free a cell (GETMAIN P or FREEMAIN P requests) for a VSM control block to IGVVSCEL.

Callers: CBBRANCH (entry-point in IGVVSCAP).

Exit-Normal: Returns to the caller.

Exit-Error: Calls IGVVSERR to issue the requested ABEND.

External-References: IGVVSCEL, IGVVSERR, IGVVSMRF, IGVVSMRT.

IGVVSM31 – Interface Routine for 31-Bit GETMAIN/FREEMAIN Requests

Function: This module routes SVC 120 GETMAIN and FREEMAIN requests. It formats pertinent information about the requests and stores this information in the VSM work area (VSWK). Then, it routes the requests to either IGVVSMRF or IGVVSMRT. When control is returned from the GETMAIN/FREEMAIN service routines, IGVVSM31 passes back a return code and, for a GETMAIN request, the address of the allocated storage.

Entry-Point IGC120: Routes SVC 120 entries from VRC, VRU, RC, and RU 31-bit GETMAIN and FREEMAIN requests.

Callers: SVC 120.

Exit-Normal: Returns to the caller.

Exit-Error: See ABEND CODES section.

Entry-Point IGVVSGBR: Routes SVC 120 global branch entry requests for global subpools in CSA, extended CSA, SQA, and extended SQA.

Callers: GLBRANCH (entry-point in IGVVSCAP).

Exit-Normal: Returns to the caller.

Exit-Error: See ABEND CODES section.

Entry-Point IGVVSB31: Routes SVC 120 equivalent local branch entries from VRC, VRU, RC, or RU GETMAIN and FREEMAIN requests.

Callers: CRBRANCH (entry-point in IGVVSCAP).

Exit-Normal: Returns to the caller.

Exit-Error: See ABEND CODES section.

External-References: IGVVSERR, IGVVSMRF, IGVVSMRT.

IGVVSTOR

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Virtual Storage Management.*

IGX00013 – MFSTART Mainline Processor

Function: Controls the initialization and termination of routines that perform RMF functions.

Entry from: IGC0010I.

Exit to: IGC0010I.

IGX00014 – MFDATA SVC Mainline Processor

Function: Controls the operation of measurement gathering routines that operate once each interval.

Entry from: IGC0010I.

Exit to: IGC0010I.

IGX00025 – TSO Command Counter Module

Function: This module accumulates, into the type 32 record, data pertaining to specified TSO commands. The data collected is:

- Number of times the command is used.
- TCB time for the command.
- SRB time for the command.
- TPUTs for the command.
- EXCPS for the command.
- Transactions for the command.

Entry from SVC FLIH.

Exit to Callers:

IKJDAIR – Dynamic Allocation Interface Routine Addressing Mode Interface Module

Function: This module passes control to the Dynamic Allocation Interface Routine (IEFSB4D0) in 31-bit addressing mode.

Entry from Various command processors and user programs.

Exit to IEFDB4D0.

Called routines None.

IKJEES20 – SEND Command Message Module

Function: This module contains the messages for the SEND command processor.

Entry from IEEVSEND, IKJEES10, IKJEES40.

Exit to Callers:

IKJEFLF – System-Initiated Cancel Schedule Routine

Function: This module schedules an SRB routine and checks the validity of the input to this routine.

Entry from IEE3703D, TIOC routines.

Exit to Callers:

IKJL4T00 – System-Initiated Cancel SRB Routine

Function: This module synchronizes events between TIOC, LOGON scheduler and its subtasks to provide orderly cancellation of a TSO user.

Entry from SRB dispatcher.

Exit to SRB dispatcher.

IKJ5803D – SEND Command Scanner

Function: This routine scans the SEND command parameter.

Entry from IEE0403D.

Exit to IEE0803D, IEE0503D.

ILRACT – Activate VIO ASPCT Routine

Function: Retrieves saved ASPCT from SYS1.STGINDEX and rebuilds it in the user's address space (LSQA). The new LGN is stored with the retrieved ASPCT base.

Entry from ILRGOS.

Exit-Normal: ILRGOS.

Exit-Error: ILRGOS

ILRASHCD – ASM Header Create/Delete Routine

Function: ILRASHCD contains entry points which create the ASM Header (ASMHD) at address space create and which delete the ASMHD at address space termination (or address space create backout).

Entry-Point ILRASHC: Creates the ASM Header.

Callers: RSM

Exit-Normal: Returns to the caller.

Exit-Error: Returns to the caller.

Entry-Point ILRASHD: Deletes the ASM Header.

Callers: RSM

Exit-Normal: Returns to the caller.

Exit-Error: Returns to the caller.

External-References: None

ILRCMP – ASM I/O Completion

Function: DIE appendages: Returns the AIA's to ILRPAGCM after the I/O has completed. If an error occurred and it is possible to retry, retries the I/O. PCI exit: The PCI exit is called by IOS for any PCI interrupt on an ASM CCW, or fora a suspend interruption on an ASM CCW, for a page fault, it returns the AIA directly to RSM. Otherwise, it returns the AIAs to ILRPAGCM. Notification exit: Forces a suspended channel program to terminate.

Entry-Point ILRCMP (Termination Exit): Marks all AIAs for retry and returns them to ILRPAGCM.

Callers: IOS Post Status, or an SRB routine scheduled by ILRCMP01.

Exit-Normal: BSM to Post Status or the Dispatcher.

Exit-Error: No exit error conditions.

Entry-Point ILRCMPD1 (Disabled Interrupt Exit): Processes successfully completed I/O requests and returns them to ILRPAGCM.

Callers: IOS

Exit-Normal: Returns to IOS from the first entry and from the second entry.

Exit-Error: Returns to IOS from the first entry and indicates "go" to Post Status to handle the problem.

Entry-Point ILRCMPAE (Abnormal and Appendage): Processes errors that occurred, returns any completed requests to ILRPAGCM, and retries any remaining requests.

Callers: IOS Post Status.

Exit-Normal: Returns to Post Status.

Exit-Error: Returns to Post Status.

Entry-Point ILRCMPNE: Entered after DIE detects error condition of unit exception, wrong length record, or invalid CCW address with clean I/O status. This entry is really a BR14 back to POST STATUS with IOSEX on causing the DASD ERP to be invoked. It can also be entered after ERP successfully completes retry, in this case, processes the successful I/O.

Callers: I/O SLIH

Exit-Normal: Returns to the SLIH.

Exit-Error: Returns to the SLIH.

Entry-Point ILRCMPNT: Causes a suspended channel program to go to termination.

Callers: IOS

Exit-Normal: Returns to IOS.

Exit-Error: No exit error conditions.

External-References:

- ILRPAGCM - ASM Page Completion
- ILRMSG00 - ASM's Message Module
- IOSVRSUM - IOS Resume Service
- IARGIO, entry point IARGIOCM - RSM back end
- IARUTRV - Translate real to virtual addresses

ILRCMP01 – I/O Completion Recovery Routine

Function: Recovery routine for ILRCMP, the I/O Completion routine. This routine attempts to clean up whatever resources have been check-pointed in the ATA, takes a dump and forces reprocessing for any requests not yet attempted. If an Abend 089 or 08A was issued from ILRCMP, records data in the SDWA, records the SDWA in SYS1.LOGREC, and retries back to ILRCMP.

Entry-Point ILRCMP01: Handles the errors in the I/O completion routine (ILRCMP) or records data in SYS1.LOGREC via the SDWA VRA for an Abend 089 or 08A issued in ILRCMP.

Callers: RTM

Exit-Normal: To RTM via R14.

Exit-Error: No exit error condition.

External-References:

- ILRMSG00 to issue an error message.
- ILRVIORB and ILRVAIA - Element verification routine.
- ILRVAIAQ, ILRVSCWQ, ILRVPCWQ - Queue verification.
- ILRCMSRB to invoke ILRPAGCM.
- IRLUTRV to translate a real address to a virtual address.

ILRCMSRB – SRB Routine to Invoke ILRPAGCM

Function: This routine (running as an SRB) is scheduled by the ASM functional recovery routines (ILRCMP01, ILRDRV01, ILRIOFRR, and ILRSRB01) and ILRPOS to invoke ILRPAGCM. RTM restrictions prohibit the ASM functional recovery routines from directly calling ILRPAGCM.

Callers: Dispatcher.

Exit-Normal: Dispatcher.

Exit-Error: No exit error conditions.

ILRCPBLD – ASM Channel Program Build Routine

Function: This routine builds a channel program for a burst of I/O requests passed from ILRIODRV and then starts the I/O.

Entry-Point ILRCPBLD: Builds a channel program for a burst of I/O requests and starts the I/O.

Callers: ILRIODRV

Exit-Normal: Returns to ILRIODRV.

Entry-Point ILRSIO: Issues the STARTIO macro

Callers: Scheduled by ILRCPBLD and ILRPGFLT.

External-References:

- IOSVRSUM - Resume service.

ILRDRV01 – Functional Recovery Routine for ILRIODRV, ILRCPBLD, and ILRPGFLT

Function: Provides recovery for ILRIODRV, ILRCPBLD, and ILRPGFLT. Validity checks ASM work queues and dequeues control blocks which are invalid or loop-causing. Restructures the environment so that the remaining requests can be handled correctly.

Callers: RTM.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References:

- ILRCMSRB - SRB routine to invoke ILRPAGCM.

- ILRIODRV (entry point ILREDRV) - ASM I/O Driver.
- ILRFRSLT (entry point ILRFRSL1) - ASM Free Slot module.
- ILRMSG00 - ASM Message module.
- ILRPOS01 - ASM ILRPOS Recovery Routine.
- ILRRSMRB - SRB Routine to invoke IARGIOCM.
- ILRFRR01 (entry point ILRAIAQ) - Verify AIA Queue Routine.
- ILRFRR01 (entry point ILRVIORB) - Verify IORB Routine.
- ILRFRR01 (entry point ILRVPCWQ) - Verify PCCW Queue Routine.

ILRFRR01 — ASM Recovery Service Routines

Function: This module contains service routines common to ASM recovery routines. There are three types of service routines: 1) queue verification routines that verify and correct ASM queues, 2) control block verification routines that verify ASM-related control blocks, and 3) ASM's resource manager termination routine (parameter for the PURGEDQ macro).

Entries:

ILRPSRMT — Reschedule an SRB for ILRCMSRB, ILRSIO, ILREDRV, or ILRSWPDR.

ILRVACE — To verify a potential ACE.

ILRVACEQ — To verify a queue of ACEs.

ILRVACQ2 — To verify and correct a queue of ACEs.

ILRVAIA — To verify a potential AIA.

ILRVAIAC — To verify a potential AIA/ACE.

ILRVIORB — To verify a potential IORB-IOSB-SRB-SRB chain.

ILRVLGE — To verify a potential LGE.

ILRVLPRQ — To verify LGE process queue (LGEPROCQ).

ILRVPCCW — To verify a potential PCCW.

ILRVPCWQ — To verify a queue of PCCWs.

ILRVSCCW — To verify a potential SCCW.

ILRVSCWQ — To verify a queue of SCCWs.

ILRVSPAQ — To verify a queue of swap AIAs.

ILRVSWTQ — To verify swap wait queue (SARWAITQ).

Entry from:

For all entries except ILRPSRMT: ILRGOS01, ILRCMP01, ILRSRB01, ILRDRV01, ILRSWP01, ILRTERMR, ILRTMI01, ILRIOFRR.

For ILRPSRMT: IEAVEPDQ.

Exit to:

For all entries except ILRPSRMT: Callers:
For ILRPSRMT: IEAVEPDQ.

ILRFRSLT

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Auxiliary Storage Management.*

ILRFRS01 — Recovery Routine for Free Slot

Function: This routine records the entry to ILRFRS01, indicates that the ASMGL lock should be released on return to RTM, requests percolation, and returns to RTM.

Entry from: RTM.

Exit to: RTM.

ILRFTASM — Format ASMVT and Related Control Blocks

Function: ILRFTASH has two functions:

1. Formats the ASMVT, SRBs/AIAs (if in use), error record, message buffer, and cache array.
2. Adds symptoms to the dump header record if certain unusual conditions exist.

Callers: ILRFTMAN.

Exit-Normal: ASMVT, SRBs/AIAs (if in use), error record, message buffer, and cache array are formatted.

Exit-Error: There are no exit error conditions.

External-References:

- ADPLESRV - IPCS service router routine.
- ADPLFRMT - IPCS format routine.

- ADPLMEMA - IPCS memory access routine.
- ADPLPRNT - IPCS print routine.
- ILRFTICB - Common routine to format invalid ID control blocks (entry point in ILRFTRTN).

ILRFASMV – Find ASM Vector Table Routine

Function: Finds the ASM Vector Table (ASMVT).

Callers: IPCS

Exit-Normal: Returns to the caller.

Exit-Error: No exit error condition.

External-References:

- BLSRACCN
- BLSRESSA
- BLSUALLO
- BLSUFRE1

ILRFTMAN – Mainline Formatter for ASM Control Blocks

Function: ILRFTMAN routes control to the appropriate routines (ILRFTASM, ILRFTPAG, ILRFTSWP, ILRFTVIO) to format ASM control blocks contained in the dump data set.

Callers: IPCS

Exit-Normal: All accessible ASM control blocks are formatted.

Exit-Error: There are no exit error conditions.

External-References:

- ADPLESRV - IPCS service router routine.
- ADPLMEMA - AMDPRDMP Print routine.
- ILRFTASM - Formats ASMVT, SRBs/AIAs, error record, message buffer, and cache array.
- ILRFTICB - Common routine to format invalid control blocks (entry point in ILRFTRTN).
- ILRFTPAG - Formats paging-related control blocks.
- ILRFTSWP - Formats swapping-related control blocks.
- ILRFTVIO - Formats VIO-related control blocks.

ILRFTPAG – Format Paging Related Control Blocks

Function: This routine formats the PART, PCTs, AIAs, the PCCW queue anchored in the ASMPCCWQ, and for each PART entry formats the PAT, IORB-IOSB-SRB-SRB chain, PCCW/AIA anchored in the IORB, and the DEIB.

Entry from: ILRFTMAN.

Exit to: ILRFTMAN.

ILRFTRTN – ASM Common Format Routines

Function: ILRFTRTN contains common routines which are used by ILRFTMAN, ILRFTASM, ILRFTPAG, ILRFTSWP, and ILRFTVIO to format ASM control blocks.

Entry-Point ILRFTICB: To format and print an invalid ID control block and 480 bytes preceding an following the control block in hexadecimal and EBCDIC.

Callers: ILRFTMAN, ILRFTASM, ILRFTPAG, ILRFTSWP, and ILRFIVIO.

Exit-Normal: Callers:'s indicated control block formatted.

Exit-Error: Error: in formatting the control block.

External-References:

- Print routine (IPCS routine).
- Format routine (IPCS routine).

ILRFTSWP – Format Swapping Related Control Blocks

Function: The function of ILRFTSWP is to format the SART, SDCT, AIAs (anchored in SARWAITQ), and the SCCW anchored in the SARSCCWQ. and the following for each SART entry: SAT, SCCW/AIA chain, IORB-IOSB-SRB-SRB chain, and SCCW/AIAs anchored in the IORB.

Entry from: ILRFTMAN.

Exit to: ILRFTMAN.

ILRFTVIO – Format VIO Related Control Blocks

Function: This routine formats the LGVT and all ASMHDs. For each ASMHD, ILRFTVIO formats the swap queue of AIAs, capture queue of AIAs, VIO Restart SRB, and the LGE queue. For each LGE on the queue, it formats the ASPCT, the ASST extension, the LPME extension, and the ACE/AIAs on the LGE process queue.

Entry from: ILRFTMAN.

Exit to: ILRFTMAN.

ILRGOS – Group Operation Starter

Function: Initiates assign, save, activate and release requests for a logical group. Calls the appropriate routine to process the request.

Callers: Call.

Exit-Normal: Returns to caller.

Exit-Error: Returns to caller.

Entry-Point ILRFRELG: Releases and frees an LGE and an LGVTE.

Callers: ILRRLG or ILRGOS01

Exit-Error: Returns to caller.

External-References:

- ILRACT
- ILRCGOSE
- ILRGOS01
- ILRRLG
- ILRSAV
- ILRSRBC

ILRGOS01 – ASM VIO Group Operators Recovery Routine

Function: ILRGOS01 is the recovery routine for ILRGOS and the routines which run under ILRGOS (ILRSAV, ILRRLG, ILRACT, and ILRVSAMI).

Callers: RTM

Exit-Error: Returns to RTM via R14.

Entry-Point ILRCGOSE: Provides ESTAE recovery for errors which occur in the logical group operator routines for save or activate LG requests.

Callers: RTM

Exit-Normal: Returns to RTM via R14.

Exit-Error: Returns to RTM via R14.

External-References:

- ILRFRELG
- ILRVACE
- ILRVLGE
- ILRVLPRQ
- ILRALS00
- ILRAFS00
- IEAVEADV

ILRIODRV – ASM I/O Driver Routine

Function: ILRIODRV receives all I/O requests for page data sets except for page faults (single reads), which are handled by ILRPGFLT. It selects slots on auxiliary storage for all write requests. It groups the requests by data set, bursts them for performance, and passes them to ILRCPBLD to start the read or write I/O to page data sets.

Entry point ILRIODRV Handles read or write I/O requests from RSM.

Callers: RSM

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry point ILRSWPIN Starts stage-one swap-in work.

Callers: ILRSWAP

Exit-Normal: Returns to the caller.

Exit-Normal: No exit error conditions.

Entry-Point ILRSWLIO: Starts an LSQA swap-out in SRB mode.

Callers: Scheduled by ILRSWAP.

Exit-Normal: Returns to the Dispatcher.

Exit-Error: No exit error conditions.

Entry-Point ILRSWAPO: Starts an LSQA swap-out.

Callers: ILRSLSQA.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point ILRMIGRT: Performs migration read and write I/O.

Callers: ILRPDPAG and ILRPDSWP.

Exit-Normal: Returns to the caller with a return code 0.

Exit-Error: Returns to the caller with a return code 4.

Entry-Point ILRVIODR: Restarts VIO requests which were queued by ILRPOS.

Callers: ILRPOS.

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

External-References:

- ILRCPBLD
- ILRFRSLT
- ILRMSG00
- ILRPOS
- ILRCVRTP
- ILRRSMRB

ILRIOFRR – ASM I/O Control Recovery Routine

Function: This routine is called by RTM anytime an error occurs during ASM's swap processing, initial page processing, and page completion processing. The mainline of the routine is a router to the appropriate recovery routine or subroutine representing the ASM function in control at the time of the error.

Entry ILRPOS01: This routine attempts recovery for ILRPOS.

Callers: RTM

Exit-Normal: Returns to the caller.

External-References:

- ILRPOS01
- IARGIOCM
- IARSCOMP
- ILRCMSRB
- ILREDRV
- IEAVEGAS
- IEAVEADV
- ILRSWP01
- ILRCSWAP
- ILRCSLSQ
- ILRCMIGR
- ILRVLPRQ
- ILRVSWTQ
- ILRVAIAQ
- ILRVSPAQ
- ILRVAIA
- ILRVACE
- ILRMIGRS

ILRJTERM – ASM Job Termination Resource Manager

Function: This routine will schedule asynchronous operations to deactivate any VIO data sets still active at job deletion time. The LGE queue is searched to locate these VIO data sets.

Entry ILRJTM01: This routine is the recovery routine for ILRJTERM.

Entry from:

For entry ILRJTERM: IEFSD166 (initiator job deletion module).

For entry ILRJTM01: IEAVTRTS (RTM).

Exit to: Callers:

ILRMSG00 – ASM Message Module

Function: Entries ILRMSG00 and ILRMSGSP: This routine writes messages concerning the status of page and swap data sets. It will terminate the system if the condition of these data sets makes it impossible for ASM to continue.

Entry ILRMSG01: This routine is the recovery routine for the termination subroutine of ILRMSG00.

Entry from:

For entry ILRMSG00: ILRCMP, ILRCMP01, ILRDRV01, ILRIODRV, ILRSWP01.

For entry ILRMSGSP: ILRTMI00 (an ASM initialization routine).

For entry ILRMSG01: IEAVTRTS (RTM).

Exit to:

For entries ILRMSG00 and ILRMSGSP: Callers, if the system is not to terminate. IGFPTERM, if the system is to terminate.

For entry ILRMSG01: IEAVTRTS.

ILROPS00 – ASM Page or Swap Open Routine

Function: ILROPS00 can perform different functions for the caller, as specified by the OPSFUNC function code:

- Open page/swap data set.
- BUILD/PCT/SDCTE and PAT/SAT.
- Free data set control blocks.
- Unallocate page/swap data set.

- Perform NIP-time cache/UCB processing.

  Callers: Call.

  Exit-Normal: The requested function is performed.

  Exit-Error: ILROPS00 will return to the caller with a non-zero return code whenever an error condition occurs.

External-References:

- VSAM Locate.
- NIPMOUNT (IEAVMNIP).
- Unit allocation/unallocation service (IEFDB440).
- IOCMSCQE.
- ILRPREAD.

ILRPAGCM – ASM Page I/O Completion

Function: This routine analyzes an input chain of AIAs, separates them, and places them onto internal queues for the page and VIO subroutine or the swap subroutine. Each subroutine processes its chain of AIAs. Successful AIAs are returned to RSM. Unsuccessful AIAs may be redriven.

Entry from: ILRCMP or ILRCMSRB.

Exit to: ILRCMP or ILRCMSRB.

External-References:

- IARGIOCM.
- IARSCOMP.
- ILREDRV.
- ILRFRSL1.
- ILRFRSW1.
- ILRSLSQA.
- ILRVIOCM.
- ILRMIGRS.

ILRPDPAG

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Auxiliary Storage Management.*

ILRPDSWP – SWAP Data Set Migration Driver

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Auxiliary Storage Management.*

ILRPDVIO

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Auxiliary Storage Management.*

ILRPEX – ASM Pool Extender

Function: This routine attempts to extend the ASM virtual storage pool indicated (ACE, BWK or SWK). ASMVT pool controller information is used and updated.

Entry from: Any ASM module using ILRGMA macro.

Exit to: Callers:

ILRPGDEL

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Auxiliary Storage Management.*

ILRPGDSP – DISPLAY ASM Command Processor Routine

Function: This module processes the DISPLAY ASM command, which produces a display of information about the page and/or swap data sets that are currently in use by the system.

Entry ILRPGDSP

Callers: IEECB808

Exit-Normal: To caller.

Exit-Error: To caller.

External-References:

- IEE0503D
- Console Message Service (IEECB808)

ILRPGEX

Function: The diagnosis information for this module is in *MVS/ESA Component Diagnosis: Auxiliary Storage Management.*

ILRPGFLT – Page Fault and Segment Fault Request Handler for RSM

Function: If the page is VIO, ILRPGFLT calls ILRPOS. It associates an IORB with the request, allocates a PCCW, builds the channel program and invokes IOS.

Entry from: IARFP, IARFS.

Exit to: Callers:

ILRPOS — Page Operations Starter

Function: ILRPOS processes VIO page operations. It performs two main types of operation.

1. Paging I/O operations - performs ASPCT and LPME manipulations to map an LPID to an LSID on auxiliary storage. Obtains additional ASST and LPME extensions in the ASPCT when required.
2. Transfer Page operations - performs ASPCT manipulation to record the transfer of a page from one auxiliary storage slot to another.

Callers: ILRIODRV, ILRPGFLT, and ILRTRPAG.

Exit-Normal: Startable AIAs passed back to the caller in the ATA.

Exit-Error: The AIAs in error are marked in error and scheduled to the RSM back end, using ILRRSMRB.

Entry-Point ILRESTRT: Restarts unstarted AIAs which are found on the LGE process queue.

Callers: ILRSRBC.

Exit-Normal: Completed processing for all input AIAs.

Exit-Error: The AIAs in error are marked in error and scheduled to the ASM back end, using ILRCMSRB.

Entry-Point ILRTRANS: Processes a single transfer page ACE. Moves the source LSID fro the ACE to the LPME.

Callers: ILRSRBC.

Exit-Normal: Completed processing for a transfer page ACE.

Entry-Point ILRTRPAG: Creates, initializes, and processes a transfer page ACE. Copies fields from the input ACA to the ACE.

Callers: RSM.

Exit-Normal: Completed processing for a transfer page ACE.

Exit-Error: The ACE, if obtained, is freed.

External-References:

- ILRFRSLT
- ILRIOFRR
- ILRVIODR
- ILRCMSRB
- ILRRSMRB

ILRPOS01 — ILRPOS Recovery Routine

Function: Performs recovery processing for ILRPOS.

Callers: ILRIOFRR and ILRDRV01.

Exit-Normal: Returns to the caller.

External-References:

- IEAVEADV
- ILRVAIA
- ILRVLPRQ
- ILRVAIAQ

ILRPREAD — ASM Special I/O Driver Routine

Function: ILRPREAD operates as an I/O driver. It performs the following functions:

- Builds temporary copies (freed prior to return) of an IOSB, a copy of the current IOSB, and SRB, a copy of the current SRB, and the channel program and its associated parameters.
- Establishes a recovery exit via an ESTAE.
- Uses the SETRP interface in the recovery exit to return parameters to RTM.
- Builds a channel program for the requested I/O. Uses the STARTIO interface to invoke IOS to perform the I/O and WAITs for the I/O to complete.
- Sets the status indicators in the cache array data area (ILRCACHE) when dealing with cached auxiliary storage subsystems.

Callers: ILRASRIM, ILRASRM1, ILRASRM2, ILROPS00, ILRPGEXP, ILRQSRIT, and ILRTMI00.

Exit-Normal: Return via Register 14.

Exit-Error: Return via Register 14.

Entry-Point ESTAEXIT: To free storage.

Callers: RTM.

Exit-Normal: Return via Register 14.

Exit-Error: Return via Register 14.

ILRRLG – Release Logical Group Operator

Function: This routine releases auxiliary storage (unsaved slots) and control blocks (active ASPCTs) relating to a logical group.

Callers: ILRGOS or ILRSRBC.

Exit-Normal: Returns to the caller.

Exit-Error: Returns to the caller.

External-References:

- ILRFRELG
- ILRFRSLU
- IARXWRLS

ILRRSMRB – SRB Routine to Invoke RSM Module IARGIOCM

Function: Invokes the RSM back end, IARGIOCM, to return AIAs to RSM without going through ASM's completion process.

Callers: ILRDVR01, ILRPOS, and ILRPOS01.

Exit-Normal: Returns to the Dispatcher.

External-References:

- IARGIOCM

ILRSAV – ASM Save Logical Group Operator

Function: This routine stores a logical group's ASPCTs in the SYS1.STGINDEX data set and flags the individual LPMEs (slot information) as saved.

Callers: ILRGOS.

Exit-Normal: Returns to the caller.

Exit-Error: Takes a record-only abend, and returns to the caller.

External-References:

- ILRFRSLU
- ILRVSAMI

ILRSRBC – VIO SRB Controller

Function: ILRSRBC finds VIO group, page or PAGEDEL requests pending in the current address space. It determines which requests can be started, then separates the startable requests into group operations and page operations and PAGEDEL requests chains and starts the requests.

Callers: ILRGOS, ILRGOS01, ILRIOFRR, ILRJTERM, ILRPDVIO, ILRSRBC, ILRSRB01, and ILRVIOCM.

Exit-Normal: Returns to caller.

Exit-Error: Returns to the Dispatcher.

Entry-Point ILRSRBRM: SRB Resource Manager.

Callers: Purge dequeue.

Exit-Normal: Returns to the caller.

External-References:

- ILRSRB01
- ILRESTRT
- ILRTRANS
- ILRRLG
- IEAOPT02
- ILRPAGCM
- ILRCVRTV

ILRSRB01 – Recovery Routine for ILRSRBC

Function: This routine processes all errors which occur in ASM's SRB controller.

Callers: IEAVTRTS (RTM).

Exit-Normal: Returns to caller.

Exit-Error: There are no exit error conditions.

External-References:

- ILRVACEQ
- ILRVAIAC
- ILRVAIAQ
- ILRVLPRQ

ILRSWAP – ASM Swap Control Module

Function: ILRSWAP accepts I/O requests from RSM and controls the placement of swap pages on swap or page data sets.

Entry-Point ILRSWAP: Processes swap-out and swap-in requests for RSM.

Callers: RSM: IARSO for swap-out requests and IARSI for swap-in requests.

Exit-Normal: Returns to the caller (RSM).

Exit-Error: No error exit.

Entry-Point ILRSLSQA: Prepares LSQA requests for swapping.

Callers: ILRSWAP mainline or ILRPAGCM.

Exit-Normal: Returns to the caller (ILRSWAP or ILRPAGCM).

Exit-Error: There are no exit error conditions.

Entry-Point ILRMIGRS: Prepares migration requests for swapping.

Callers: ILRPDSWP or ILRPAGCM.

Exit-Normal: Returns to the caller (ILRPDSWP).

Exit-Error: There are no exit error conditions.

External References:

- ILRFRSW1
- ILRCTBDS
- ILRCVRTP
- ILRCVRTS
- ILRSWAP0
- ILRSWLIO
- ILRSWPIN
- ILRMIGRT

ILRSWPDR – Swap Driver

Function: This routine drives IOS via the STARTIO macro for pages being written to or read from a swap data set.

Entry from: IEAVEDS0 for SRB scheduled by ILRSLSQA (entry of ILRSWAP) or ILRCMP.

Exit to: IEAVEDS0.

ILRSWP01 – ASM Swap Recovery Routine

Function: ILRSWP01 and its entry-points receive control from ASM's I/O FRR, ILRIOFRR, to process errors which occur in ASM's

Entry-Point ILRSWP01: Recovers for errors in ILRSWPDR.

Callers: ILRIOFRR.

Exit-Normal: Label SWPDCRTN.

Entry-Point ILRCSWAP: Recovers for errors in ILRSWAP.

Callers: ILRIOFRR.

Exit-Normal: Return to caller.

Entry-Point ILRCSLSQ: Recovers for errors in ILRSLSQA.

Callers: ILRIOFRR.

Exit-Normal: Label SWPDCRTN.

Entry-Point ILRCMIGR: Recovers for errors in ILRMIGRS.

Callers: ILRIOFRR.

Exit-Normal: Return to caller.

External-References:

- ILRFRR01
- ILRMSG00

ILRSWSCN – ASM Pre-swap Scan Routine

Function: ILRSWSCN checks the status of VIO data set I/O for each VIO data set in the current address space. It sets indicators to tell RSM which VIO I/O requests have not yet been started.

Entry-Point ILRSWSCN: Sets indicators in the AIA for any VIO I/O requests in the current address space which have not yet been started.

Callers: RSM.

Exit-Normal: All AIAs processed. Returns to the caller.

Exit-Error: No exit error conditions.

External-References: None.

ILRTERMR – ASM Address Space Termination Resource Manager

Function: ILRTERMR receives control from RSM termination during termination of any address space. ILRTERMR frees ASM resources for the address space (auxiliary storage slots) or marks them to be freed when in-process operations complete. ILRTERMR returns non-pagedel VIO requests, unstarted swap-out requests and completed swap-out requests to RSM.

Entry-Point ILRTERMR: Recovers ASM resources during address space termination.

Callers: RSM termination (IARGT entry-point IARGTERM).

Exit-Normal: Returns to the caller.

Exit-Error: No exit error conditions.

Entry-Point TERMRFRR: Recovers for errors in ILRTERMR.

Callers: RTM.

Exit-Normal: Returns to the caller.

External-References:

- ILRFRSW1

- ILRFRSL1
- ILRVSPAG
- ILRVLPRQ

ILRTMI01 – ESTAE for ILRTMRLG and ILRTMI00

Function: Provides recovery recovery for ILRTMRLG and its calls to ILRTMI00 and ILRVSAMI. Attempts to complete ILRTMI00 processing (via retry points) and prevents the ILRTMRLG task from terminating.

Callers: RTM

Exit-Normal: Return to RTM on Register 14.

Exit-Error: Branch to ILRCRTM1 in ILRTMRLG.

External-References:

- ILRAFS00
- ILRCRTMX
- ILRTMI00
- ILRTMRLG
- ILRVACE
- ILRVACQ2
- IEAVEADV
- MSGPROC

ILRTMRLG – Task Mode Release Logical Group

Function: This routine loads and calls ILRTMI00 to complete ASM initialization. Erases saved ASPCTs and release the slots assigned in those ASPCTs).

Callers: IEEMB860.

Entry-Point ILRCRTM5: Retry point used by the recovery module (ILRTMI01) should ILRTMI00 fail.

Callers: ILRTMI01.

External-References:

- ILRFRSLT
- ILRTMI00
- ILRVSAMI

ILRVIOCM – VIO Completion

Function: This routine stores the newly-assigned LSID in an ASPCT for each complete I/O processing of a VIO page.

Entry from: ILRPAGCM.

Exit to: ILRPAGCM.

ILRVSAMI – VSAM Interface Routine

Function: This routine interfaces with VSAM to complete any I/O to SYS1.STGINDEX involved in processing save, activate, and release logical group operations of ASM.

Entry from: ILRACT, ILRSAV, or ILRTMRLG.

Exit to: Callers:

IOSCONSL – IOS Console Command Routine.

Function: This module processes the following console commands:

- DISPLAY IOS.
- SETIOS.
- SET IOS=xx.

Callers: None.

Exit-Normal: Returns to caller.

Exit-Error: To RTM (see entry point ICONSTAE).

Entry-Point IEECB938: Processes the SETIOS console command.

Callers: None.

Exit-Normal: Returns to caller.

Exit-Error: To RTM (see entry point ICONSTAE).

Entry-Point IEECB953: Processes the SET IOS=xx console command.

Callers: None.

Exit-Normal: Returns to caller.

Exit-Error: To RTM (see entry point ICONSTAE).

Entry-Point ICONSTAE: Maximizes the MIH change processing.

Callers: None.

Exit-Normal: Returns to RTM to either retry mainline processing or to continue with termination.

Exit-Error: To RTM (see entry point ICONSTAE).

External-References:

- IOSCPARZ
- IEEMB878

IOSCPARZ – IOS MIH/HOTIO Parser

Function: This module receives the IOSDCONS (CONS) table from module IEAVNPF2 (NIP time) or from module IOSCONSL (CONSL) via COMTASK. The CONS table will contain a MIH or HOTIO control record. The record source will be from:

- A specific member IECIOSxx of SYS1.PARMLIB.
- The 'SETIOS MIH,...,' console command.

  Note: SETIOS is treated as a single card from IECIOSxx. There is no special code for SETIOS. A SETIOS is treated as a single record from IECIOSxx.
- The 'DISPLAY IOS,MIH...,' console command.

Callers: IEAVNPF2 and IOSCONSL.

Exit-Normal: Returns to caller.

Exit-Error: To RTM.

Entry-Point PARZSTAE: To maximize MIH change processing.

Callers: RTM.

Exit-Normal: Return to RTM to either retry mainline processing or to continue with termination.

External-References: WTO.

IOSERDTE – IOS Detailed Edit Report Formatter Routine

Function: IOSEROUT, the IOS Formatter Router module, invokes IOSERDTE to format the Detailed Exit report for Cx SYS1.LOGREC error records.

Callers: IOSEROUT.

Exit-Normal: Return to caller.

External-References: IFCPRINT.

IOSERDTS – IOS Detailed Summary Report Formatter Routine

Function: IOSEROUT, the IOS Formatter Router module, invokes IOSERDTS to increment the error record counters for Cx SYS1.LOGREC error record summary report when all the error records have been counted.

Callers: IOSEROUT.

Exit-Normal: Return to caller.

External-References: IFCPRINT.

IOSEREVH – IOS Event History Report Formatter Routine

Function: IOSEROUT, the IOS Formatter Router module, invokes IOSEREVH to format the Event History report for Cx SYS1.LOGREC error records.

Callers: IOSEROUT.

Exit-Normal: Return to caller.

External-References: None.

IOSERIT – IOS Cx Class Record PRI Tables

Function: This module defines the Program Control Tables for the Program Record Interface (PRI) for the EREP program. It specifies the routing modules that control the formatting of the Cx software error records for the EREP reports. The following is the entry point that define the PRI tables for these two records:

- IFCRV2C2 (record type 'C2'X)

Callers: None.

External-References: None.

IOSEROUT – IOS Formatter Router Module

Function: EREP invokes this module through the Program Record Interface (PRI) for the Cx class of software error records for the System Summary, Event History, Detailed Edit and Detailed Summary EREP reports. This module routes control to the appropriate formatting routine for the report type specified by the function code passed by EREP.

Callers: EREP via the PRI PCT interface.

Exit-Normal: Return to caller.

External-References:

- IOSERSSU
- IOSEREVH
- IOSERDTE
- IOSERDTS

IOSFQVIR – IOS Queue Verification Interface Routine

Function: Provides an interface between any module wishing to verify an IOQ chain and the queue verification routine (IEAVEQV2). The caller need not be a recovery routine.

Callers: Any module wishing to verify an IOQ chain.

Exit-Normal: Returns to the caller.

Entry-Point IOSFQIOQ: Element verification called by queue verifier.

Callers: System queue verification routine.

Exit-Normal: Returns to the caller.

External-References: IEAVEQV2.

IOSPURGA – I/O Request Purge Routine

Function: IOSPURGA validates the caller's authorization to use routines that purge I/O from the following:

- Supervisor interrupt request block (SIRB)
- Asynchronous exit queue.
- Unit control blocks (UCBs).
- Dynamic device reconfiguration (DDR) queues.
- Service priority lists (LSPL, GSPL, LSMQ, and GSMQ).
- I/O purge interface block (IPIB).

IOSPURGA also calls IGC016 to decrease the count of outstanding I/O operations in an IPIB.

Entry-point IGC016

Callers: Issuers of SVC 16

Exit-Normal: Returns to the issuer of the SVC 16.

Exit-Error: To RTM to record and percolate the error.

Entry-Point: IOSPGRVR

Callers: RTM

Exit-Normal: Returns to RTM.

Entry-Point: IOSPTME1

Callers: Dispatcher

Exit-Normal: Returns to the caller.

External-References: IOSVSMGR, IFGDEBVR, IOSFQVIR, IOSPURGD, IOSPURGB

IOSPURGB – Functional Purge Routine

Function: IOSPURGB marks, dequeues, or halts I/O applicable to the current purge operation. Each entry point purges I/O from a different system data area as follows:

- The supervisor interrupt request block (SIRB).
- The asynchronous exit queue.
- The DDR queues.
- The service priority list (SPL).
- The I/O purge interface block (IPIB).

Entry-Point: IOSPBSP2.

Callers: IOSPURGA and IOSPURGC

Exit-Normal: Return to the caller.

Exit-error To RTM to record and percolate.

Entry-Point: IOSPAPCK

Callers: IOSPURGC

Exit-Normal: Return to the caller.

Exit-error To RTM to record and percolate or to retry.

Entry-Point: IOSPDVPG

Callers: IOSPURGA

Exit-Normal: Return to the caller.

Exit-Error: To RTM to record and percolate.

Entry-Point: IOSVPRGA

Callers: IOSVIRBA

Exit-Normal: Return to the caller.

Exit-Error: To RTM to record and percolate.

Entry-Point: PURGBFRR

Callers: RTM

Exit-Normal: Returns to RTM.

Entry-Point: IOSPGRTM

Callers: IEAVEPD0

Exit-Normal: Return to the caller.

Exit-Error: Issue 'COD' abend if PURGEDQ not issued by IOSPURGB or IPIB off ASCB is not valid.

Entry-Point: IOSPMPGB

Callers: IOSPURGA

Exit-Normal: Return to the caller.

Exit-Error: To RTM to record and percolate.

External-References: Purge the driver exits.

IOSPURGC – I/O Purge Asynchronous Exits and Branch Entry Interface

Function: IOSPURGC implements I/O prevention, IOQ purging, and includes the asynchronous exits needed to manage the results of HSCH or CSCH requests.

Entry-Point: IOSPURGC

Callers: IOSVPRVT

Exit-Normal: Returns to the caller.

Exit-Error: To RTM to record and percolate.

Entry-Point: IOSPIOQP

Callers: IOSPURGB

Exit-Normal: Returns to the caller.

Entry-Point: IOSPDIE

Callers: IOSVIRBH

Exit-Normal: Returns to the dispatcher.

Entry-Point: HALTTERM

Callers: None

Exit-Normal: Returns to the dispatcher.

Exit-Error: To RTM to record and percolate.

Entry-Point: HALTRMTR

Callers: None

Exit-Normal: Return to the dispatcher

Exit-Error: To RTM to record and percolate.

External-References: IOSPAPCK, IOSPBSP2, IOSVHSCH, IOSVHCRS, IOSVSMGR

IOSPURGD – IOS Purge Quiesce Count Decrement Routine

Function: IOSPURGD decreases the IPIB purge quiesce count. When the count goes to zero, IOSPURGD either posts an ECB or schedules an SRB.

Entry-Point: IECVQCNT

Callers: Drivers' termination procedures, IOSPURGA - IOS purge routine

Exit-Normal: Returns to the caller.

External-References: IEAOPT01, IEAOPT02

IOSRACRW – IOS Asynchronous CRW Processor

Function: To obtain the channel report words (CRWs), either hardware or software generated, and route the CRW to the appropriate processing module for handling. Hardware CRWs are obtained from the channel, software CRWs from the Channel Report Word Queue elements (CRWQs) generated by the caller.

Callers: IOS modules IOSRMCH, IOSRHREC, IOSVFCHP and IOSRACRW FRR routine.

Exit-Normal: Returns to the dispatcher via address in CVTSRBRT.

Exit-Error: To RTM.

Entry-Point ACRWFRR: Recovers from an unexpected error that caused exit to RTM.

Callers: RTM.

Exit-Normal: RTM.

Exit-Error: RTM.

External-References: IOSRCHPR, IOSRSCH.

IOSRCDEV – IOS Clear Device Routine

Function: This routine clears the subchannel(s) associated with a device.

Callers: IOSRCHDR, IOSRFDEV, IOSRHREC or IOSRRRSV.

Exit-Normal: Returns to the caller or the SRB dispatcher.

Entry-Point RESUME: Resumes IOSRCDEV's processing after an asynchronous service has been performed. When the asynchronous service routine completes, it schedules an SRB to invoke this module at entry-point RESUME. RESUME re-establishes this routine's FRR, obtains the UCB lock for the device, and returns to the point within IOSRCDEV where the service routine was invoked.

Callers: Dispatcher.

Exit-Normal: Returns to IOSRCDEV's caller or the SRB dispatcher.

Entry-Point HSCHDIE: Performs the disabled interrupt exit functions when the I/O interrupt is received for a halt subchannel instruction that was invoked by IOSRCDEV. HSCHDIE indicates to IOSVIRBH:

- The action to be taken on the active start (subchannel) request.
- That IOSVIRBH is to schedule IOSRCDEV at entry-point RESUME to continue processing the clear device request.

Callers: IOSVIRBH.

Exit-Normal: Returns to the caller.

Entry-Point CSCHDIE: Performs the disabled interrupt exit functions when the I/O interrupt is received for a clear subchannel instruction that was invoked by IOSRCDEV. CSCHDIE indicates to IOSVIRBH:

- The action to be taken on the active start request.
- That IOSVIRBH is to schedule IOSRCDEV at entry-point RESUME to continue processing the clear device request.

Callers: IOSVIRBH.

Exit-Normal: Returns to the caller.

Entry-Point TIMERDIE: Performs the disabled interrupt exit functions when the external interrupt is received for a SETDIE issued by IOSRCDEV. TIMERDIE schedules IOSRCDEV at entry-point RESUME to continue processing the clear device request.

Callers: Timer SLIH.

Exit-Normal: Returns to the caller.

Entry-Point CDEVFRR: Recovers from an unexpected error in:

- This routine.
- A synchronous routine invoked by this routine that does not have an FRR established or whose FRR percolates the error.

Callers: Recovery/Termination Manager (RTM), on behalf of the SETFRR issued by this routine.

Exit-Normal: Returns to RTM.

Entry-Point FRRRETRY: Performs retry processing after an abnormal termination has occurred. It ensures the UCB lock is held and returns to the mainline to perform a clear subchannel without issuing a store or halt subchannel.

Callers: RTM.

Exit-Normal: Same as entry-point IOSRCDEV.

External-References: IOSRSRBR, IOSVHSCH, IOSVSTSQ, SETDIE.

IOSRCHDR – IOS Device Recovery

Function: This routine performs recovery for a device that has an error on one of its on-line channel paths. There are three sources of errors that cause this routine to be invoked.

1. Hardware errors detected by the channel subsystem.
2. Hot I/O errors detected by IOS hot I/O recovery (IOSRHREC).
3. Force-channel-path-off-line commands from the operator.

Callers: IOSRCHPR.

Exit-Normal: Returns to IOSRCHPR.

Entry-Point RESUME: Resumes device recovery after an asynchronous service has been performed. When the asynchronous service completes, it schedules an SRB to invoke IOSRCHDR at entry-point RESUME. RESUME re-establishes this routine's FRR, obtains the UCB lock for the device being recovered and returns to the point within IOSRCHDR where the asynchronous service was invoked.

Callers: Dispatcher.

Exit-Normal: Returns to the dispatcher.

Entry-Point ASYNMSCH: Frees the area containing the IOSB, SCHIB, and SRB that were used to modify a subchannel asynchronously.

Callers: Dispatcher.

Exit-Normal: Returns to the dispatcher.

Entry-Point CHDRFRR: Recovers from an unexpected error in:

- IOSRCHDR.
- A synchronous routine invoked by IOSRCHDR that does not have an FRR established or whose FRR percolates the error.

Callers: Recovery/Termination Manager (RTM), on behalf of the SETFRR issued by this routine.

Exit-Normal: Returns to RTM.

Entry-Point FRRRETRY: Performs retry processing after IOSRCHDR has abnormally terminated. It ensures that the UCB lock is held and returns to the mainline to complete the device recovery.

Callers: RTM.

Exit-Normal: Same as entry-point IOSRCHDR.

External-References: IECVDPTH, IOSVMSCQ, IOSRCDEV, IOSRCHPR, IOSRDBOX, IOSRRRSV.

IORCHDS – IOS Channel Path Device Scan

Function: This routine initiates channel path device recovery for each device that has recovery path installed.

Entry from: IOSRCHPR

Exit-Normal: Returns to the caller.

Called Routines IOSRCHDR, IOSRCHF, IOSRSCH, SETDIE.

IOSRCHPR – IOS Channel Path Recovery

Function: This module is called by the asynchronous channel report word (CRW) processor (IOSRACRW) to recover the failing channel path whenever the retrieved CRW indicates a channel path is the source of the error. IOSRCHPR is also called to force a channel path offline because of the CONFIG CHP, FORCE operator command or to process some Hot I/O conditions that require channel path processing.

Callers: IOSRACRW.

Exit-Normal: To SRB dispatcher via address in CVTSRBRT.

Exit-Error: To RTM.

Entry-Point IOSRCHPX: Continues channel path recovery after device recovery (IOSRCHDR) completes all asynchronous processing for devices affected by the channel path condition.

Callers: IOSRCHDR.

Exit-Normal: To SRB dispatcher via address in CVTSRBRT.

Exit-Error: To RTM.

Entry-Point CHPRFRR: Recovers from an unexpected error that caused exit to RTM.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: RTM.

External-References: IOSRCHDR, IEEVLDWT, IEAVEQV1, IEEVCHPF.

IOSRDBOX – IOS Device Boxing Routine

Function: Boxes a device that is in error. Boxing prohibits IOS from doing I/O to the device and queuing I/O requests for the device. The device may also be boxed due to an unrecoverable hardware error on the control unit or the channel path.

Callers: IOS modules such as:

IOSRFDEV - Force device off-line.

IOSRHREC - Hot I/O recovery.

IOSRCHDR - Channel path device recovery.

IOSRRRSV - Re-reserve.

IOSRSCH - Subchannel recovery.

IOSVURSV - Unconditional reserve service.

Exit-Normal: Returns to the caller using the address supplied at entry in register 14.

Exit-Error: Returns to the caller using the address supplied at entry in register 14. See also entry-point BOXFRR.

Entry-Point BOXFRR: Recovers from an unexpected error that caused exit to RTM.

Callers: RTM.

Exit-Normal: Returns to RTM specifying retry and the address of a routine in IOSRDBOX.

External-References: IOSVMSCQ, IOSVSCHR, IEFAB4GB, IECVDPTH, IOSRSRBR.

IOSRDPSV – IOS Dynamic Pathing Validation Interface Module

Function: IOSRDPSV performs the service function for callers to request dynamic pathing validation on a device basis. This module schedules an SRB to perform the asynchronous processing required to perform the validation and then returns to the caller. The asynchronous SRB routine interfaces with IECVDPTH to read the Sense-IDs (SNID). The SRB routine then calls IECVDPSV. IECVDPSV evaluates this SNID data.

Entry-Point IOSRDPSV: See function for entry points purpose.

Callers: Callers: who request the DPS validation of a path group.

Exit-Normal: Returns to caller.

Entry-Point RDPSVFRR: The function of this entry point is to recover from an that caused exit to RTM.

Callers: RTM

Exit-Normal: RTM

Exit-Error: RTM

External-References:

- IECVDPTH
- IECVDPSV

IOSRFDEV – IOS Force Device Routine

Function: This routine forces a device off-line.

Callers: IECVFDEV or IOSRHREC.

Exit-Normal: Returns to the caller.

Entry-Point FDEVFRR: Recovers from an unexpected error in:

- This routine.
- A synchronous routine invoked by this routine that does not have an FRR established or whose FRR percolates the error.

Callers: Recovery/Termination Manager (RTM), on behalf of the SETFRR issued by this routine.

Exit-Normal: Returns to RTM.

Entry-Point FRRRETRY: Performs retry processing after an abnormal termination has occurred. It ensures the UCB lock is held and returns to the main routine to retry the force device request.

Callers: RTM.

Exit-Normal: Same as entry-point IOSRFDEV.

External-References: IOSRCDEV, IOSRDBOX, IOSRRRSV, IOSRSRBR.

IOSRHDET – Hot I/O Detection

Function: This module determines if Hot I/O is occurring. Hot I/O is defined as a hardware malfunction which results in the generation of repeated unsolicited I/O interrupts. If a Hot I/O condition is detected, the IOCSLIH field of the IOCOM is updated to point to the Hot I/O SLIH (entry-point IOSRSLIH in this module) rather than IOSVSLIH. When the next interrupt for this device occurs, the Hot I/O SLIH receives control to stop the device from presenting unsolicited interrupts and to schedule Hot I/O recovery (IOSRHREC).

Callers: IOSVIRBU.

Exit-Normal: Returns to the caller.

Entry-Point IOSRSLIH: This entry receives control for all interrupts after a Hot I/O condition has been detected. It stops the Hot device from presenting unsolicited interrupts and it schedules Hot I/O recovery.

Callers: I/O FLIH.

Exit-Normal: If pending status is for a Hot device, then return is made to the I/O FLIH. If pending status is not for a Hot device, then return is made to the I/O SLIH.

Entry-Point IOSRHTPI: This entry receives control from the I/O SLIH (IOSVSLIH) when a new interrupt is brought in via a TPI instruction and the address contained in IOCSLIH is not the I/O SLIH's own entry address. This entry-point is currently the same as IOSRSLIH.

Callers: I/O SLIH.

Exit-Normal: Same as IOSRSLIH entry-point.

Entry-Point HDETFRR: Recovers from an unexpected error that caused exit to RTM. This routine records the error and retries if possible. If unable to retry, it frees resources and percolates the error.

Callers: RTM.

Exit-Normal: Returns to RTM.

External-References: IOSVSLIH, IOSRHREC.

IOSRHREC – Hot I/O Recovery

Function: This module attempts to recover from Hot I/O (a hardware malfunction that results in the generation of repeated interrupts).

Callers: Scheduled by IOSRHDET.

Exit-Normal: Returns to the dispatcher.

Entry-Point HRECFRR: Records the error, reschedules the Hot I/O recovery processing, and, if possible, retries to clean up resources.

Callers: RTM.

Exit-Normal: Returns to RTM.

External-References: IOSRDBOX, IOSRFDEV, IOSRACRW, IEEVLDWT, IOSRCDEV.

IOSRISCE – Interrupt Subclass Update Routine

Function: This routine updates the interrupt subclass mask in control register 6 for the current processor.

Callers: Dispatcher.

Exit-Normal: Returns to the dispatcher via the address in CVTSRBRT.

External-References: None.

IOSRMCH – IOS Machine Check Exit

Function: Checks for and processes I/O machine checks.

Callers: IGFPMSCA, IGFPMAIN, and IEAVTACR.

Exit-Normal: Returns to the caller or exits to IGFPTERM to terminate the system.

Exit-Error: To machine check interrupt handler or to RTM.

External-References: IGFPTERM, IOSRACRW.

IOSRMIHI – Missing Interrupt Handler Initialization Routine

Function: This modules creates the environment needed to scan for possible missing interrupt conditions.

Callers: IEEMB860.

Exit-Normal: Return to RTM to either retry mainline processing or to continue with termination.

External-References: A call is issued to SETDIE, which is pointed to by TPCSDIE, to initially start each timer.

IOSRMIHL – MIH LOGREC Recording Module

Function: Writes records to LOGREC.

Callers: IOSRMIHP, IOSRMIHR.

Exit-Normal: None, this module goes into a Wait for the next POSTing of the LOGREC ECB while MIH termination is not occurring.

Exit-Error: MIH processing is terminating due to the MIHTERM bit being on for a MIH processing problem.

External-References: None.

IOSRMIHM – MIH Message Writing Module

Function: Issues messages for requests queued out of the message stack header in the MIH work area.

Callers: IOSRMIHP, IOSRMIHR.

Exit-Normal: None, this module goes into a Wait for the next POSTing of the message ECB while MIH termination is not occurring.

Exit-Error: MIH processing is terminating due to the MIHTERM bit being on for a MIH processing problem.

External-References: None.

IOSRMIHP – Missing Interrupt Handler Analysis Module

Function: Detects and processes missing interrupts from I/O devices. IOSRMIHP schedules a global SRB for entry into this module at the entry-point appropriate to the type of timer interval that expired. The SRB has been initialized by IOSRMIHT.

Callers: Timer SLIH (IEAVRTI0).

Exit-Normal: Returns to the caller.

Entry-Point IOSRMIH0: Scans non-DASD UCBs assigned to the normal timer interval for missing interrupts.

Callers: Dispatcher.

Exit-Normal: BR 14 returns to the dispatcher.

Entry-Point IOSRMIH1: Scans non-virtual DASD UCBs for missing interrupts.

Callers: Dispatcher.

Exit-Normal: BR 14 returns to the dispatcher.

Entry-Point IOSRMIH2: Scans UCBs assigned to the 3330V timer interval for missing interrupts.

Callers: Dispatcher.

Exit-Normal: BR 14 returns to the dispatcher.

Entry-Point IOSRMIH3: Scans UCBs assigned to an installation-specified timer interval for missing interrupts.

Callers: Dispatcher.

Exit-Normal: BR 14 returns to the dispatcher.

Entry-Point MIHPFRR: Recovers from an error that caused exit to RTM.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: Returns to RTM.

External-References: IOSRMIHR, IOSVSTSQ, IEAQTE00, Device-dependent MIH exit routine pointed to by DDTMIH.

IOSRMIHR — Missing Interrupt Handler Device Recovery

Function: Recovers I/O devices after a missing interrupt. IOSRMIHR initiates device recovery for a missing interrupt.

Callers: IOSRMIHP.

Exit-Normal: Returns to IOSRMIHP.

Entry-Point HALTDIE: Processes the interrupt received on the Halt Subchannel issued by IOSRMIHR.

Callers: IOSVIRBH.

Exit-Normal: Returns to IOSVIRBH.

Entry-Point CLRDIE: Processes the interrupt received on the Clear Subchannel issued by IOSRMIHR.

Callers: IOSVIRBH.

Exit-Normal: Returns to IOSVIRBH.

Entry-Point HCTERM: Initiates the logging function and frees the Halt/Clear Control Block (MHCB).

Callers: Dispatcher.

Exit-Normal: Returns to the dispatcher.

Entry-Point MIHRFRR: Performs recovery for failures in IOSRMIHR mainline code.

Callers: RTM.

Exit-Normal: Returns to RTM.

External-References: IOSVHSCH, IOSVSTSQ, IOSVURVL.

IOSRMIHT — Missing Interrupt Handler Initialization Routine

Function: This module creates the environment needed to scan UCBs for possible missing interrupt conditions.

Callers: Master scheduler.

Exit-Normal: None. This routine enters a wait state until IOSRMIHP determines that MIH processing should be terminated or until the logging subtask or message subtask abnormally terminates.

Entry-Point MIHESTAE: Terminates MIH processing.

Callers: RTM.

Exit-Normal: Returns to RTM to either retry mainline processing or to continue with termination.

Entry-Point ESTIENTM: Performs ESTAI recovery for IOSRMIHM. Takes an SVC dump and continues with termination whenever subtask IOSRMIHM fails.

Callers: RTM.

Exit-Normal: Returns to RTM to continue with termination.

Exit-Error: Returns to RTM. RTM will give control to MIHESTAE.

Entry-Point ESTIENTL: Performs ESTAI recovery for IOSRMIHL. Takes an SVC dump and continues with termination whenever subtask IOSRMIHL fails.

Callers: RTM.

Exit-Normal: Returns to RTM to continue with termination.

Exit-Error: Returns to RTM. RTM will give control to MIHESTAE.

External-References: SETDIE (pointed to by TPCSDIE), DEQUETQE (pointed to by CUTQTD00).

IOSROUTG – IPL/OUTAGE Recorder

Function: This module writes the IPL and outpage records via SVC 76.

Callers: Master scheduler initialization (IEEVIPL).

Exit-Normal: None, this module is always in a write record and wait loop.

External-References: None.

IOSRRRSV – IOS Re-Reserve Module

Function: This module is an IOS service module used during recovery to perform any function associated with re-reserving a device, releasing a reserve on a device, or issuing an unconditional reserve to a device.

Callers:
Unconditional Reserve (IOSVURSV), Channel Path Device Recovery (IOSRCHDR), Force Device (IOSRFDEV).

Exit-Normal: Returns to the caller if no I/O is issued or to the dispatcher if I/O has been issued.

Entry-Point SSCHEXIT: Receives control from IOSVSSCH prior to the issuing of the SSCH instruction in order to save the current UCBLPUM (last path used mask) and to determine if the I/O should still be issued.

Callers: Start Subchannel Routine (IOSVSSCH).

Exit-Normal: Returns to the caller.

Entry-Point DIEPROC: Receives control from IRB analysis module with the UCB lock held after the I/O interrupt for the requested I/O has been received. It performs disabled processing for the interrupt. Recovery for this routine is provided by IOSVIRBA's FRR.

Callers: IRB Analysis Module (IOSVIRBA).

Exit-Normal: Returns to the caller via a BSM.

Entry-Point ABNREXIT: Receives control if the requested I/O completes with error status to perform abnormal exit processing. Recovery for this routine is provided by IECVPST's FRR.

Callers: IOS Post Status Module (IECVPST).

Exit-Normal: Returns to the caller via a BSM.

Entry-Point IOCOMPLT: Receives control upon completion of IOS processing of the I/O request. IOSRRRSV processing continues here. If the I/O was successful or if it was determined that a retry will not help, returns to the caller. Otherwise, this routine determines if I/O should be re-attempted and, if so, sets up and re-issues the I/O. If no reattempt is to be made, returns to the caller.

Callers: IOS Post Status (IECVPST).

Exit-Normal: Returns to the original caller of IOSRRRSV or to the dispatcher if I/O is being re-issued. The IOSRRRSV caller will in turn return to the dispatcher.

Entry-Point RRSVFRR: Handles unexpected errors that occur in entry-points IOSRRRSV and IOCOMPLT.

Callers: RTM.

Exit-Normal: Returns to RTM.

External-References: IOSRCDEV, IEAVDVAL, IOSRDBOX, IOSRSRBR.

IOSRSAIO – Stand-Alone I/O Routine for Disabled Console Communications

Function: :

1. Entry-point IOSRSAIO performs a caller-defined I/O operation to a specified device.
2. Entry-point IOSRSUBC sets or resets the interruption subclass of a specified device.

Callers: Disabled Console Communications Program (IEEVDCCR).

Exit-Normal: Returns to the caller.

Entry-Point IOSRSUBC: Sets or resets the interruption subclass of a designated device.

Callers: Disabled Console Communications Program (IEEVDCCR).

Exit-Normal: Returns to the caller.

External-References: IARUTRV.

IOSRSCH – Subchannel Recovery

Function: This module is scheduled by IOSRACRW (asynchronous CRW processor) when the CRW indicates a subchannel related error in the CRWRSC (reporting source code) field, or from the IOSRSCH FRR routine when the recovery routine determines that retry cannot be performed. IOSRSCH processes the two error-related conditions of available state and installed parameters initialized state.

Callers: IOSRACRW, IOSRSCH.

Exit-Normal: To SRB dispatcher via address in CVTSRBRT.

Exit-Error: To RTM.

Entry-Point RSCHFRR: Recovers from an unexpected error that caused exit to RTM.

Callers: RTM.

Exit-Normal: RTM.

Exit-Error: RTM.

External-References: IOSVSMGR.

IOSRSLH – IOS Subchannel Logout Handler

Function: This module is called by the interruption response block (IRB) analysis modules to process a logout if the IRB indicates that the extended status work (ESW) contains logout information. It records the error, issues the error message, builds the error recovery procedure information block (ERPIB) for subsequent use by an ERP, and interfaces with unconditional reserve validation routine.

Callers: IOSVIRBA and IOSVIRBU.

Exit-Normal: Returns to the caller.

Exit-Error: To RTM.

Entry-Point SLHFRR: Recovers from unexpected errors causing exit to RTM.

Callers: RTM, for SETFRR request from this module.

Exit-Normal: Returns to the caller (RTM).

Exit-Error: RTM.

External-References: IARXMCKS, IARUTRV, Measurement Facility Owner pointed to by the CVTCSOMF field in the CVT, Address-limit Facility Owner pointed to by the CVTCSOAL field in the CVT, IOS Storage Manager pointed to by the IOCSMLG field in the IOCOM, IOSVURVL.

IOSRSRBR – IOS SRB Recovery Routine

Function: This routine is scheduled when an IOS routine that was called in SRB mode and is performing processing asynchronous to its caller's processing cannot recover from an error. This routine returns in SRB mode to the routine that called the failing routine.

Callers: Dispatcher.

Exit-Normal: Returns to the routine that called the failing routine.

External-References: None.

IOSVATTN – IOS Attention Table

Function: This module contains the attention table used by IOS to control processing of attention or unsolicited device end status. A dummy attention routine is also provided.

Callers: None.

Exit-Normal: Not applicable.

Exit-Error: Not applicable.

Entry-Point DUMATTN: Provides a dummy attention routine.

Callers: None

Exit-Normal: Returns to caller.

External-References: Various attention routines.

IOSVCNT – IOQ Counting Service

Function: Provides a count of all inactive I/O requests that are queued for a particular device.

Callers: Any routine running disabled and in supervisor state-key zero.

Exit-Normal: Returns to the caller.

Entry-Point VCNTFRR: Handles error conditions and attempts to recover use of the function.

Callers: RTM.

Exit-Normal: Returns to RTM to complete recovery processing.

External-References: IOSFQVIR.

IOSVCNXL – Store & Modify Subchannel Cancel Routine

Function: This routine dequeues a Store Subchannel or a Modify Subchannel request, freeing any storage obtained by IOS to track the request. If the request is successfully dequeued, this routine schedules the SRB indicated in the IOSB and sets a return code of zero. If the routine cannot find the request on the queue, it sets a return code of four.

Callers: Users of the asynchronous STORE SUBCHANNEL and MODIFY SUBCHANNEL functions.

Exit-Normal: Returns to the caller.

Entry-Point CNXLFRR: Handles error conditions and attempts to recover use of the function.

Callers: RTM.

Exit-Normal: Returns to RTM to complete recovery processing.

External-References: IOSFQVIR, IOSVSMIF.

IOSVDATA – IOS Common Area and Control Blocks

Function: This module defines storage for and initializes the following IOS storage areas:

- IOCOM (I/O Communications area - read only).
- IOCOMW (I/O Communications area - writeable).
- CHRB (Channel Recovery Block).
- ISDT (Interrupt Subcalls Definition Table).
- IOS Synchronization Lock Table.
- Default IRB.
- IOWA (I/O Work area table).
- IOQ staging queues.
- Module work area table.
- IOS SDUMP dump parameters table.
- Dynamic Pathing Host ID table.
- SRB pool for scheduling IOSRACRW.
- MIH tables, MIHA and MIHBs.
- IOQ storage manager page pool table.
- IOS storage manager page pool table.
- SRB used by IOSVIOQM compression routine.
- SRB used by IOSVSMGR compression routine.

Callers: None.

External-References: None.

IOSVDAVV – IOS Direct Access Volume Verification

Function: This module verifies that the correct volume is mounted on a direct access device.

Callers: IOSVSSCH.

Exit-Normal: Returns to the caller.

External-References: IECVPST, IOSVSMGR, IEEVLDWT.

IOSVDEVN – Convert Device Number to EBCDIC Routine

Function: This module converts a device number for the UCB pointed to by register 1 to printable EBCDIC.

Callers: Invokers of the UCBDEVN macro.

Exit-Normal: Return to caller.

External-References: None.

IOSVDPDR – Dynamic Pathing IO Driver Module

Function: This routine is the I/O driver for the dynamic pathing module, IECVDPTH. It builds the CCWs and data to be sent to the device, and it issues the STARTIO macro to process the request.

Callers: IECVDPTH.

Exit-Normal: Returns to the caller.

Entry-Point DPDRNORX (Normal: Exit): Performs additional processing if the STARTIO macro was successful.

Callers: IECVPST.

Exit-Normal: Returns to IECVPST.

Entry-Point DPDRDABX (Abnormal Exit): Handles errors that occurred during requests to disband a device or to add a device to a path group.

Callers: IECVPST.

Exit-Normal: Returns to IECVPST.

Entry-Point DPDRPABX (Abnormal Exit): Handles errors that occurred during requests to remove a path from a path group or to add a path to a path group.

Callers: IECVPST.

Exit-Normal: Returns to IECVPST.

Entry-Point NEXTSSCH: Issues another STARTIO macro after DPDRNORX, DPDRDABX or DPDRPABX has returned to IECVPST with a return code of four in register 15 and this entry-point address in register zero. Redrive return to IECVPST (return code eight) is not used because it could cause an interlock situation. The set path group ID command could be going to a page pack device. A page pack request could be queued at one IOSLEVEL while holding a local or CMS lock. IECVDPTH could be issuing the retry I/O at a higher IOSLEVEL. Retry return to IECVPST causes IECVPST to request the local lock in order to process the retry request. If IECVPST is waiting for the local lock and the page pack I/O is queued waiting for the higher IOSLEVEL I/O to go, the pack would be in an interlock state.

Callers: IECVPST.

Exit-Normal: Returns to IECVPST.

Entry-Point DPDRTRMX (Termination Exit): Resets the IOS level (if this module set the level) and decrements the request count (if necessary) and schedules an SRB to give control to DPDRSRB.

Callers: IECVPST.

Exit-Normal: Returns to the dispatcher.

Entry-Point DPDRSRB: Performs termination processing for this module. This entry point receives control via a SCHEDULE macro after I/O has been completed.

Callers: DPDRTRMX, DPDRPURG.

Exit-Normal: Returns to IECVDPTH.

Entry-Point DPDRPURG: Handles PURGE I/O processing.

Callers: IGC0001F.

Exit-Normal: Returns to IGC0001F.

Entry-Point DPDRSSXA: Terminates request if device is found to be hot.

Callers: IOSVSSCH.

Exit-Normal: Returns to IOSVSSCH.

Entry-Point DPDRFRR: Recovers from an unexpected error that caused an exit to RTM. On first entry, attempts retry. If entered a second time, percolates the error.

Callers: RTM.

Exit-Normal: To RTM.

Exit-Error: To RTM.

External-References: IOSRSRBR.

IOSVDSTF – IOS Device State Transition Flushing

Function: This is an IOS service module which flushes (forces presentation of) all pending device state transition status from I/O devices.

Entry-Point IOSVDSTF: Initiates device state transition flushing by scheduling IOSVDST2.

Callers: IOSVSSCH.

Exit-Normal: Returns to the caller.

Entry-Point IOSVDST2: Issues the I/O request to do device state transition flushing on the first path to the device.

Callers: Dispatcher.

Exit-Normal: Returns to the caller (dispatcher).

Entry-Point DSTFABNX: Processes the abnormal completion of the device state transition flushing I/O request.

Callers: IECVPST.

Exit-Normal: Returns to the caller.

Entry-Point DSTFTRMX: Processes the termination of the device state transition flushing I/O request.

Callers: IECVPST.

Exit-Normal: Returns to the caller.

Entry-Point DSTFFRR: To recover from an unexpected error in this module, or when a module called by IOSVDSTF percolates to IOSVDSTF.

Callers: Recovery/Termination Manger (RTM), on behalf of the SETFRR issued by this routine.

Exit-Normal: Returns to RTM.

External-References: IEAVMAP (Device Path Map Build routine).

IOSVFCHP – IOS Routine to Force a Channel Path Offline

Function: This module schedules IOS channel path recovery modules to force a channel path offline.

Callers: Channel Path Reconfiguration Routine (IEEVCHPR).

Exit-Normal: Returns to the caller.

Exit-Error:

To RTM.

Entry-Point FCHPESTA: Recovers from an unexpected error that caused exit to RTM. On first entry, retries at the exit code, label RETRYPNT. If entered a second time, percolates the error.

Callers: RTM (for ESTAE requests from this module).

Exit-Normal: To RTM.

Exit-Error: To RTM.

External-References: IOSRACRW.

IOSVFMTH – IOS AMDPRDMP User Exit Routine

Function: This module examines the operands specified to AMDPRDMP and formats and prints the appropriate control blocks. This module is also called from the IPCS module BLSRIOSK with the EXCEPTION operand to perform the IOSCHECK command processing.

Callers: AMDPRDMP, BLSRIOSK.

Exit-Normal: Returns to the caller.

Exit-Error: Returns to the caller.

Entry-Point FMTESTAE: Fills in the VRA when an unexpected error occurs.

Callers: RTM.

Exit-Normal: Returns to RTM.

Exit-Error: Returns to RTM.

External-References: IOSVFMTY, AMDPRDMP Service Routines (ACCESS, FORMAT, PRINT).

IOSVFMTI – IOS IOSB Formatter Routine

Function: This module formats an IOSB.

Callers: IOSVFMTY and IPCS.

Exit-Normal: Returns to caller.

External-References: ADPLESRV (IPCS service routine).

IOSVFMTP – IOS Formatter Parse Routine

Function: IOSVFMTP parses the operands on the IOSDATA verb.

Entry-Point IOSVFMTP: See function for entry points purpose.

Callers: IOSVFMTH

Exit-Normal: Returns to caller.

External-References:

- IKJPARS

IOSVFMTR – IOS Recovery Control Block Formatter

Function: IOSVFMTR formats control blocks related to IOS recovery actions. IOSVFMTR also adds symptoms strings.

Entry-Point IOSVFMTR: See function for entry points purpose.

Callers: IOSVFMTH

Exit-Normal: Returns to caller.

External-References:

- IOSVFMTY
- ADPLESRV

IOSVFMTU – IOS UCB Formatter Routine

Function: This module formats a UCB.

Callers: IOSVFMTH (IOS formatter exit), and IPCS.

Exit-Normal: Returns to caller.

External-References: ADPLESRV (IPCS service routine).

IOSVFMTV – IOS Formatter PDE Validation Routine

Function: IOSVFMTV validates the device numbers in a Parse Descriptor Entry (PDE).

Entry-Point IOSVFMTV: See function for entry points purpose.

Callers: IOSVFMTH

Exit-Normal: Returns to caller.

IOSVFMTY – IOS Control Block Dump Formatter

Function: This module formats IOS control blocks requested by the IOS AMDPRDMP user exit routine, IOSVFMTH.

Callers: IOSVFMTH.

Exit-Normal: Returns to the caller.

External-References: AMDPRDMP Service Routines (ACCESS, FORMAT, PRINT).

IOSVHSCH – IOS Halt Subchannel/Clear Subchannel Routine

Function: This module determines whether the requested I/O operation (Halt Subchannel or Clear Subchannel) can be started or not. If it can, this module issues the appropriate instruction (HSCH or CSCH).

Callers: IOS modules such as IGC0001F and IGC0003C.

Exit-Normal: Returns to the caller using the address supplied at entry in register 14.

Exit-Error: Returns to the caller using the address supplied at entry in register 14. See also entry-point HSCHFRR.

Entry-Point HSCHFRR: Recovers from an unexpected error that caused exit to RTM.

Callers: RTM.

Exit-Normal: Returns to RTM specifying retry and the address of a routine in IOSVHSCH.

External-References: IOSVSMGR, IOSFQVIR, System trace via PTRACE macro, GTF trace via HOOK macro.

IOSVIEDN – IPCS EPCDIC Device Number Service Routine

Function: This module returns the EBCDIC device number for a UCB.

Callers: Dump formatter routines running in the IPCS environment.

Exit-Normal: Returns to caller.

External-References: None.

IOSVIPID – IOS I/O Prevention Identifier (IOPIDENT) Services Module

Function: IOSVIPID provides a service function for obtaining and releasing I/O prevention identifiers (IOPIDs).

Entry-Point: IOSVIPID

Callers: Issuers of the IOPIDENT macro.

Exit-Normal: Returns to the caller.

Exit-Error: To RTM.

Entry-Point: VIPIDFRR

Callers: RTM

Exit-Normal: Returns to RTM.

Exit-Error: Returns to RTM.

External-References: None

IOSVIOQC – Scan-IOQ-Chain Service Routine

Function: Scans a UCB's IOQ chain, invoking a subroutine provided by the caller for each IOQ that represents a request to start I/O.

Callers: Any routine holding the UCB lock.

Exit-Normal: Returns to the caller.

Entry-Point MIOQCFRR: Handles error conditions: verifies the IOQ chain, dumps the UCB and associated IOQs, and percolates the error to the caller's FRR.

Callers: RTM.

Exit-Normal: Returns to RTM to complete error processing.

External-References: USERTN (address supplied in register zero), IOSFQVIR.

IOSVIOQM – IOQ Storage Manager

Function: This module manages the IOQ blocks that the IOS component requires in order to provide I/O functions.

Entry-Point IOSVSMIG: Normal: processing for acquiring an IOQ block.

Callers: IOS modules that require an IOQ block.

Exit-Normal: Returns to caller.

Entry-Point IOSVSMIF: Normal: processing for freeing an IOQ block.

Callers: IOS modules that request an IOQ block to be freed.

Exit-Normal: Returns to caller.

Entry-Point IOSVQFRR: Recover from unexpected errors that caused exit to RTM.

Callers: RTM and storage manager callers who provided their own FRR coverage.

Exit-Normal: Return to caller.

Exit-Error: Return to caller.

Entry-Point SMGFREVR: Verify that the storage manager's pool pages are chained correctly. The FRR invokes the system queue verification routine, which invokes this function after doing some preliminary processing.

Callers: IEAVEQV2 (System Queue Verification Routine).

Exit-Normal: Return to caller.

Entry-Point IOSVQMEL: Called by the ENF facility on event code 4 (free SQA) to initialize and schedule the IOQ compress function. When dispatched, the IOSVQCOM entry point is entered to perform the compression.

Callers: ENF routine IEFENFNM.

Exit-Normal: Return to caller.

Entry-Point IOSVQCOM: To compress the IOQ block pages.

Callers: Dispatcher when SRB is scheduled.

Exit-Normal: Return to caller.

External-References: IEAVEQV1.

IOSVIRBA – IOS IRB Analysis Routine

Function: This module analyzes the status of an I/O interrupt, either real or simulated. The status is in the interruption response block (IRB).

Callers: IOSVSLIH and routines simulating interrupts with simulated status.

Exit-Normal: Returns to the caller.

Exit-Error: RTM.

Entry-Point IRBAFRR: Recovers from an unexpected error that caused exit to RTM.

Callers: RTM.

Exit-Normal: Returns to RTM (caller).

Exit-Error: RTM.

External-References: Supervisor Trace Table Routine via PTRACE Macro, GTF via HOOK Macro, IOSRSLH, IOSVIRBD (Device status analysis routine), IOSVIRBN (Deferred CC 3 and N-bit processor), Device Dependent Trap routine, Driver DIE, IOSVIRBH, IOSVIRBU, IOSVSMGR, IOSVSMFR, IGC0001F, IEAVEIOP.

IOSVIRBD – IOS Device Status IRB Analysis Routine

Function: This module is called by the other IRB analysis modules to analyze the device status field of the IRB. The device status bits analyzed are the device end, attention, unit check and unit exception. This module will read the sense data associate with unit checks. This module interfaces with the device-dependent unsolicited and end-of-sense exits. This module also performs the attention and intercept handling.

Callers: IOSVIRBA and IOSVIRBU.

Exit-Normal: Returns to the caller.

Exit-Error: To RTM.

Entry-Point VIRBDFRR: Recovers from an unexpected error that caused exit to RTM.

Callers: RTM.

Exit-Normal: Returns to the caller (RTM).

External-References: GTF, Device-dependent unsolicited exit routine, Device-dependent end-of-sense exit routine, IECVPST, IOSVSMGR (entry-points IOSVSMIG, IOSVSMIF, IOSVSMLG, and IOSVSMLF).

IOSVIRBH – IOS Halt/Clear Function IRB Analysis Routine

Function: This module analyzes an interruption response block (IRB) that is a result of a Halt or Clear Subchannel instruction.

Callers: IOSVIRBA.

Exit-Normal: Returns to the caller.

Exit-Error: To RTM.

Entry-Point IOSVHCRS: Frees resources tied to the associated Start IOSB. The Halt/Clear DIE must call this routine when setting return code eight to ensure that IOS resources are freed.

Callers: Halt/Clear DIE.

Exit-Normal: Returns to the caller.

Exit-Error: To RTM.

Entry-Points VIRBHFRR: Recovers from an unexpected error that caused exit to RTM.

Callers: RTM.

Exit-Normal: RTM.

Exit-Error: RTM.

External-References: GTF, IOSVIRBN, IOSDIE, IOSVSMGR (entry-point IOSVSMIF and entry-point specified in the IOCSMLF field of the IOCOM that returns large blocks).

IOSVIRBN – IOS Deferred CC3 and N-Bit Analysis Routine

Function: This module analyzes the N-bit and the deferred condition code three conditions in the IRB. The caller passes a parameter in register zero to indicate what processing is to be done.

Callers: IOSVIRBA, IOSVIRBH, IOSVIRBU, and IOSVSLIH.

Exit-Normal: Returns to the caller.

Exit-Error: To RTM.

Entry-Point VIRBNFRR: Recovers from an unexpected error that caused exit to RTM.

Callers: RTM.

Exit-Normal: RTM.

Exit-Error: RTM.

External-References: GTF, IOSVURVL, IECVPST, IOSVSMGR (entry-point IOSVSMIG, entry-point IOSVSMIF, entry-point specified in the IOCSMLG field of the IOCOM that obtains large block(s), and entry-point specified in the IOCSMLF field of the IOCOM that returns large block(s)).

IOSVIRBU – IOS Unsolicited Status IRB Analysis Routine

Function: Analyzes an IOS interruption response block (IRB) for unsolicited interruptions.

Callers: IOSVIRBA.

Exit-Normal: Returns to the caller.

Exit-Error: To RTM.

Entry-Point UNSOLFRR: Recovers from an unexpected error that caused exit to RTM.

Callers: RTM.

Exit-Normal: RTM.

Exit-Error: RTM.

External-References: GTF via HOOK Macro, IOSRSLH, IOSRHDET, IOSVIRBD, IOSVSMGR (entry-points IOSVSMIF, IOSVSMFR, and IOCSMLFE).

IOSVLEVL – IOS Level Module

Function: To get, test, set or reset the level associated with a specified function.

Callers: Users of the IOSLEVEL Macro.

Exit-Normal: Returns to the caller.

Entry-Point IOSCHKLV: This routine verifies that the UCBLEVEL and UCBHILVL fields of the UCB correspond to UCBLVMSK and updates these fields if necessary. This routine is for use in recovery situations.

Callers: IOSVLEVL's FRR, IOSVSWAP.

Exit-Normal: Returns to the caller.

Entry-Point LVLFRR: Records error conditions, frees the UCB lock (if obtained by IOSVLEVL) and percolates the error. Also verifies that the UCBLEVEL and UCBHILVL fields of the UCB correspond to the highest level set in the UCBLVMSK field by calling the IOSCHKLV entry-point.

Callers: RTM.

Exit-Normal: Returns to RTM.

External-References: IOSVSCHR.

IOSVLOOK – UCB Look Module

Function: Locates the unit control block (UCB) associated with a specified device number.

Callers: IECVGENA.

Exit-Normal: Returns to the caller.

External-References: None.

IOSVMODI – IOSB Format Models

Function: This module contains format models for the IOSB.

Callers: None.

External-References: None.

IOSVMODU – UCB Format Models

Function: This module contains format models for the UCB.

Callers: None.

External-References: None.

IOSVMOD1 – IOS Format Models

Function: This module contains format models for IOS control blocks.

Callers: None.

External-References: None.

IOSVMOD2 – IOS Format Models

Function: This module contains format models for IOS control blocks.

Callers: None.

External-References: None.

IOSVMSCH – IOS Modify Subchannel Initiation Module

Function: This module issues a Modify Subchannel (MSCH) instruction to update the Subchannel Information Block (SCHIB).

Callers: IOSVMSCQ, IOSVSCHR.

Exit-Normal: Returns to the caller.

Exit-Error: C0D ABEND.

Entry-Point IOSMSCHF: Recovers from unexpected errors.

Callers: RTM.

Exit-Normal: Returns to the caller.

Entry-Point IOSVPDEC: Schedules an SRB to decrease the purge wait count for requests that a purge quiesce has counted.

Callers: IOSVSTSC, IOSVMSCH.

Exit-Normal: Returns to the caller.

Entry-Point IOSVMSRB: Calls the purge decrement count routine.

Callers: Scheduled by IOSVPDEC.

Exit-Normal: Returns to the dispatcher.

External-References: IOSVSMGR, GTF via HOOK macro, System trace, IECVQCNT, IOSFQVIR.

IOSVMSCQ – IOS Modify Subchannel Queue Module

Function: This module processes a modify subchannel request.

Callers: Users of modify subchannel service.

Exit-Normal: Returns to the caller.

Entry-Point IOSMSCQF: Recovers from unexpected errors.

Callers: RTM.

Exit-Normal: Returns to the caller.

External-References: IOSVMSCH, IOSVSMGR.

IOSVNPTH – IOS NIP Path Verification Routine

Function: This routine issues I/O along each physically available channel path to determine the accurate set of logical paths to a device.

Callers: IEAVNP02, IEAVNPM3.

Exit-Normal: Returns to the caller.

External-References: IEA0PT01, IGC0001F, DIEXIT.

IOSVPRVT – I/O Prevention Service Routine

Function: IOSVPRVT receives control from callers that are requesting I/O prevention. IOSVPRVT unallocates the callers I/O prevention identifier (IOPID) and provides the interface for calling the IOS purge function. The IOS purge function satisfies the I/O prevention purge request.

Entry-Point: IOSVPRVT

Callers: Callers: who issue the PREVNTIO macro.

Exit-Normal: Return to the caller.

Entry-Point: VPRVTFRR

Callers: RTM

Exit-Normal: RTM

Exit-Error: RTM

External-References: IOSPURGC, IOSVIPID

IOSVRSTS – IOS Restart Support Routine

Function: This module is scheduled by the restart FLIH extension module to diagnose certain types of system problems and repair them if possible.

Callers: Dispatcher.

Exit-Normal: Returns to the dispatcher.

Exit-Error: To RTM.

Entry-Point RSTSFRR: Recovers from unexpected errors that caused exit to RTM.

Callers: RTM.

Exit-Normal: Returns to the caller (RTM).

Exit-Error: RTM.

External-References: IEEVLDWT.

IOSVRSUM – Resume I/O Service Routine

Function: This routine issues the Resume Subchannel Instruction.

Callers: Users of the resume subchannel instruction.

Exit-Normal: Returns to the caller.

Entry-Point RSUMSRB: Issues the Resume Subchannel Instruction when mainline processing is unable to obtain the UCB lock and, therefore, schedules an SRB for entry at RSUMSRB.

Callers: IOSVRSUM.

Exit-Normal: Returns to the dispatcher.

Entry-Point RSUMFRR: Cleans up resources and records an error that occurred in IOSVRSUM.

Callers: RTM.

Exit-Normal: Returns to RTM.

External-References: System trace, GTF trace.

IOSVSCHR – Subchannel Redrive Routine

Function: This module redrives subchannel requests queued on the caller-specified UCB.

Callers: IOSVSLIH, IECVGENA, IOSRDBOX.

Exit-Normal: Returns to the caller.

Exit-Error: To RTM.

Entry-Point SCHRFRR: Recovers from an error that caused exit to RTM.

Callers: RTM, for SETFRR request from IOSVSCHR or IOSVSSCH.

Exit-Normal: Returns to RTM.

Exit-Error: RTM.

External-References: IOSVSSCH, IOSSSCHF, IOSFQVIR.

IOSVSHUP – Shared UP Service Routine

Function: IOSVSHUP determines whether a device that was SYSGENed with the SHAREDUP option should have the shared indicator in its UCB set or reset (turned off).

Callers: All modules that desire this function.

Exit-Normal: Returns to the caller.

External-References: None.

IOSVSLIH – IOS Second-Level Interruption Handler

Function: This module obtains and processes the status of a subchannel that caused an I/O interruption.

Callers: IEAVEIO, IOSRHDET.

Exit-Normal: Returns using the address supplied at entry in register 14.

Exit-Error: Returns using the address supplied at entry in register 14. See also entry-point SLIHFRR.

Entry-Point SLIHFRR: Recovers from an unexpected error that caused exit to RTM.

Callers: RTM.

Exit-Normal: Returns to RTM specifying retry and the address of a routine in IOSVSLIH.
This page left blank.

Entry-Point IOSHTSCH: Makes location of TSCH instruction known externally. Contains no executable code.

Entry-Point IOSHTPI: Makes location of TPI instruction known externally. Contains no executable code.

External-References: IOSVIRBA, IOSVSCHR, ILRCMP, IOSRHDET, IOSFQVIR, IOSVIRBN, system trace via PTRACE macro, GTF trace via HOOK macro, ASM notification exit via VIONOTE.

IOSVSMGR – IOS Storage Manager

Function: This module controls and manages the three control block sizes (IOQ, RQE (medium) and large blocks) that the IOS and EXCP components require in order to provide I/O functions.

Entry-Point IOSVSMIG: Normal processing for acquiring an IOQ (small) block.

Callers: IOS modules that require an IOQ block.

Exit-Normal: Returns to the caller.

Entry-Point IOSVSMIF: Normal processing for freeing an IOQ block.

Callers: IOS modules that request an IOQ block to be freed.

Exit-Normal: Returns to the caller.

Entry-Point IOSVSMLG: Normal processing for acquiring large block(s).

Callers: IOS modules requesting large block(s).

Exit-Normal: Returns to the caller.

Entry-Point IOSVSMLF: Normal processing for returning (freeing) large block(s).

Callers: IOS modules freeing large blocks(s).

Exit-Normal: Returns to the caller.

Entry-Point IECVSMMG: Normal processing for acquiring an RQE block (medium block).

Callers: IECVEXCP.

Exit-Normal: Returns to the caller.

Entry-Point IECVSMMF: Normal processing for freeing an RQE (medium) block.

Callers: IECVEXCP.

Exit-Normal: Returns to the caller.

Entry-Point IECVSMEG: EXCP processing for acquiring large block(s).

Callers: IECVEXCP.

Exit-Normal: Returns to the caller.

Entry-Point IECVSMEF: EXCP processing for freeing large block(s).

Callers: IECVEXCP.

Exit-Normal: Returns to the caller.

Entry-Point IOSVSMPF: Normal processing for purging/freeing large and RQE (medium) blocks in the caller's address space. This function is called by purge (SVC 16) when it receives an address space halt request (usually as a result of an out-of-storage ABEND or address space termination.

Callers: IGC0001F.

Exit-Normal: Returns to the caller.

Entry-Point IOSVCPRM: For IOS storage initialization:

- Entered from NIP to initialize the large and medium block storage pools, and to initialize the IOWA and other processor-related work areas.
- Entered from CONFIG CPU, on-line to initialize the IOWA and other processor-related work areas.

These two callers must provide serialization to ensure that processor initialization is done serially and not concurrently, since this function is only serially reusable.

Callers: NIP module IEAVNP00 and CONFIG module IEECVCPR.

Exit-Normal: Returns to the caller.

Entry-Point IOSVSMFR: Recovers from unexpected errors that caused exit to RTM.

Callers: RTM and storage manager callers who provided their own FRR coverage.

Exit-Normal: Returns to the caller.

Exit-Error: Returns to the caller.

Entry-Point SMGFREVR: Verifies that the storage manager's pool pages are chained correctly. The FRR invokes the system queue verification routine, which invokes this function after doing some preliminary processing.

Callers: IEAVEQV2 (System Queue Verification Routine).

Exit-Normal: Returns to the caller.

Entry-Point IOSVSMEI: Establishes an IOS storage manager ENF listener environment on event code four,(free SQA). Entry-point IOSVSMEL in the storage manager is established as the listening routine to be called on the occurrence of event four.

Callers: IOSVIOSI.

Exit-Normal: Returns to the caller.

Entry-Point IOSVSMEL: Called by the ENF facility on event code four (free SQA) to determine if a compress of the medium and large blocks would be beneficial. If the compress function is to be performed, the compress SRB is initialized and scheduled. When dispatched, IOSVSCOM entry-point is entered to perform the compress.

Callers: ENF routine IEFENFNM.

Exit-Normal: Returns to the caller.

Entry-Point IOSVSCOM: Compresses both medium and large block pages.

Callers: Dispatcher when SRB is scheduled.

Exit-Normal: Returns to the caller (Dispatcher).

External-References: IEAVEQV1.

IOSVSSCH – IOS Start Subchannel Initiation Module

Function: This module determines if an SSCH request can be started. If it can, this module invokes the device-dependent SIO or sense exit to make further checks and to cause the Start Subchannel instruction (which initiates the I/O request) to be issued. This module also handles the condition code set by the Start Subchannel instruction.

Callers: IOSVSSCQ, IOSVSCHR.

Exit-Normal: Returns to the caller.

Exit-Error: X'C0D' ABEND.

Entry-Point IOSVSENS: Issues a Start Subchannel for a sense request.

Callers: IOSVSCHR.

Exit-Normal: Returns to the caller.

Entry-Point IOSVSCP: Issues the Start Subchannel instruction requested by the device-dependent exit.

Callers: Device-dependent SIO exit.

Exit-Normal: Returns to the caller.

Entry-Point IOSSSCHF: Recovers from errors that caused exit to RTM.

Callers: RTM.

Exit-Normal: Returns to RTM specifying retry and the address of a retry routine.

External-References: Device-dependent SIO exits, Device-dependent sense exits, Start subchannel exit, System trace, GTF via HOOK macro, IOSVSMGR, IOSVIRBA, IOSVDAVV.

IOSVSSCQ – Start Subchannel Queueing Routine

Function: Queues a START I/O request onto the appropriate UCB and calls the start subchannel initiation routine to attempt to start the request.

Callers: Any I/O driver or the STARTIO Compatibility Interface Routine (IECVSTIO).

Exit-Normal: BSM to the caller's return address.

Entry-Point SSCQFRR: Handles error conditions and attempts to recover use of the function.

Callers: RTM.

Exit-Normal: Returns to RTM to complete recovery processing.

Exit-Error: Returns to RTM to percolate the error to the caller.

Entry-Point SSCQFRR2: Performs second-level recovery and tries to recover use of the function.

Callers: RTM.

Exit-Normal: Returns to RTM to complete recovery processing.

External-References: IOSVSMGR (entry-points IOSVSMIG, IOSVSMIF, and IOSVSMFR), IOSVSSCH (entry-points IOSVSSCH and IOSSSCHF).

IOSVSTSC – IOS Store Subchannel Initialization Routine

Function: This modules determines whether a Store Subchannel request can be started. If it can, this module issues the Store Subchannel instruction (STSCH).

Callers: IOSVSCHR, IOSVSTSQ.

Exit-Normal: Returns to the caller using the address supplied at entry in register 14.

Exit-Error: Returns to the caller using the address supplied at entry in register 14. See also entry-point STSCFRR.

Entry-Point STSCFRR: Recovers from an unexpected error that caused exit to RTM.

Callers: RTM.

Exit-Normal: Returns to RTM specifying retry and the address of a routine in IOSVSTSC.

External-References: IOSVSMGR, IOSFQVIR.

IOSVSTSQ – IOS Store Subchannel Queue Routine

Function: Queues a Store Subchannel request on the appropriate UCB and calls the Store Subchannel initiation routine (IOSVSTSC) to execute the request.

Callers: Users of Store Subchannel service.

Exit-Normal: Returns to the caller using the address and addressing mode supplied at entry in register 14.

Exit-Error: Returns to the caller using the address and addressing mode supplied at entry in register 14. See also entry-point STSQFRR.

Entry-Point STSQFRR: Recovers from an unexpected error that caused exit to RTM.

Callers: RTM.

Exit-Normal: Returns to RTM specifying retry and the address of a routine in IOSVSTSQ.

External-References: IOSVSMGR, IOSVSTSC, IOSFQVIR.

IOSVSUCB – UCB Scan Service

Function: Provides a scan of the UCB chain for all UCB's or all UCB's within a specified device class by being repetitively invoked by the caller.

Callers: Routines that need to obtain device lists.

Exit-Normal: Returns to the caller.

External-References: None.

IOSVSWAP – Swap Service Routine

Function: This routine swaps the contents of two UCBs for dynamic device reconfiguration (DDR) and the MSS ERP.

Callers: Dynamic Device Reconfiguration (DDR) and the MSS ERP.

Exit-Normal: Returns to the caller.

Exit-Error: Returns to the caller.

Entry-Point SWAPFRR: Cleans up resources and records an error that occurred in the swap service routine.

Callers: RTM.

Exit-Normal: Returns to RTM.

Entry-Point SRBFTUCB: Notifies IOSVSWAP of the completion of an asynchronous Modify Subchannel request to enable the UCB. This is a general routine for either the from-UCB or the to-UCB.

Callers: Dispatcher.

Exit-Normal: Returns to the dispatcher.

Entry-Point SWAPSRB: Calls the Modify Subchannel service from the FRR to update the interrupt parameters in the UCB.

Callers: Dispatcher.

Exit-Normal: Returns to the dispatcher.

External-References:

- IEFAB4GA, IOSVMSCQ, IOSLEVEL, IOSCHKLV (entry-point in IOSLEVEL).

IOSVUCBS – UCB Find Routine

Function: Find the unit control block (UCB) corresponding to a specific device address. Build an equate symbol record (macro BLSRESSY) to describe the control block.

Callers: None.

Exit-Normal: Return to caller with return code 0.

Exit-Normal: Return to caller with standard IPCS return codes.

External-References: See "FLOWS" block comment before PROCEDURE statement.

IOSVUCBV – UCB Validity Check Routine

Function: Validity check the unit control block (UCB).

Callers: None.

Exit-Normal: Return to caller with return code 0.

Exit-Normal: Return to caller with standard IPCS return codes.

External-References:

- BLSRACCQ
- BLSRSAG
- BLSRSAPC
- BLSUALLO
- BLSUFRE1
- BLSUMPKN
- BLSUPUTV

IOSVURDT – Unconditional Reserve Detection Routine

Function: This routine is called by IECVPST to complete the unconditional reserve processing. On first entry, this module determines whether unconditional reserve processing will continue. It checks the status of the I/O request initiated by IOSVURVL to determine whether the error was caused by the loss of a device path or control unit. If either of these conditions exist, unconditional reserve processing continues with a call to IECVDURP. If the error is considered temporary, this module calls IOSLEVEL to reset the unconditional reserve level and redrive the device. On all subsequent entries, this module calls IECVDURP.

Callers: IECVPST.

Exit-Normal: Returns to the caller.

Entry-Point IOSVURSV: This routine is called by IECVDURP to perform recovery for a device that has a device path or control unit failure. If recovery is requested, it calls IOSRRRSV to issue the unconditional reserve command. If the device is not reserved, a release is also issued. If the request was to box the device, or the unconditional reserve failed and IECVDURP requests the device be boxed, it obtains the UCB lock and calls IOSRDBOX to box the device. IOSRDBOX issues a message to identify the device that was boxed. On return from IOSRDBOX, the UCB lock is released. If requested, the routine redrives the device via IOSLEVEL.

Callers: IECVDURP.

Exit-Normal: Returns to the caller.

Entry-Point URDTFRR: Recovers from an unexpected error that caused exit to RTM. This FRR provides recovery for two entry-points in this module, IOSVURDT and IOSVURSV, and IECVDURP processing. The first time this procedure is entered from either IOSVURDT or IOSVURSV, it attempts to retry. If entered a second time from the same procedure or the first time from IECVDURP, this FRR schedules an SRB to give control to IOSRSRBR. IOSRSRBR will return to IECVPST at the normal return point from IOSVURDT.

Callers: RTM.

Exit-Normal: To RTM.

Exit-Error: To RTM.

External-References:

- IECVDURP, IOSRRRSV, IOSRDBOX, IOSRSRBR.

IOSVURVL – Unconditional Reserve Validation Routine

Function: This routine is called by IOSRSLH to determine if unconditional reserve processing should be performed for channel check errors or by IOSVIRBN for deferred condition code three errors. This routine also can be called from the CLEAR DIE of IOSRMIHR to handle simulated interrupts and start pending conditions.

Callers: IOSRSLH, IOSVIRBN, IOSRMIHR.

Exit-Normal: Returns to the caller.

Entry-Point URVLFRR: Recovers from an unexpected error that caused exit to RTM. On first entry, attempts retry. If entered a second time, percolates the error.

Callers: RTM.

Exit-Normal: To RTM.

Exit-Error: To RTM.

External-References:

- IOSFQVIR, IEAVDVAL, IOSVSMGR.

IOSVVARY – IOS Vary Service

Function: This service routine updates the path indicators (both software and hardware) for a path that is being varied on-line or off-line. For off-line requests, this module also ensures that any previously active I/O for the requested path is complete. Because the off-line request is processed asynchronously, a cancel option is also provided.

Callers: Reconfiguration (VARY path and VARY channel path.

Note: Vary Path is the only user of the on-line function).

Exit-Normal: Returns to the caller.

Entry-Point VARYSRB: This entry-point receives control when the modify subchannel service routine completes for the vary off-line request. The dynamic pathing module (IECVDPTH) is called to resign the path from the path group. This entry-point also posts the caller to indicate the vary has completed.

Callers: Modify Subchannel Service (IOSVMSCH).

Exit-Normal: Returns to the caller.

Entry-Point VARYRMTR: Entered when PURGEDQ is issued to free resources that were obtained by this module.

Callers: PURGEDQ.

Exit-Normal: Returns to the caller.

Entry-Point VARYFRR: Records the error and, if possible, recovers from the error by retrying. If a retry is not possible, frees resources obtained by the module.

Callers: RTM.

Exit-Normal: Returns to the caller.

External-References: IECVDPTH, IOSVMSCQ, IOSVCNXL.

IRARMANL – SRM Syntax Analyzer

Function: This module uses a syntax table (IPSYNTAB, ICSYNTAB or OPSYNTAB) to analyze the syntax of input data. Depending on the analysis, it transfers control to the required option initialization routine.

Entry from: IRARMIPS, IRARMICS, IRARMOPT, and IRARMANL (recursive).

Exit to: Caller.

IRARMCHM – SRM Channel Measurement Routine

Function: This module consists of several individually invoked subroutines that collect measurements associated with channels.

Entry from: For entry IRARMUPD (a timed algorithm): IRARMCTL Time Queue Processor. For entry IRARMLPI: SYSEVENT Processor. For entry IRARMMSP: SYSEVENT Processor, IRARMCHM (IRARMUPD), IRARMIOM, IRARMERR.

Exit to: Caller.

IRARMCNS – SRM Constants Module

Function: Provides the pre-assembled tables used within the SRM for non-refreshable data.

Entry from: Not applicable.

Exit to: Not applicable.

IRARMCPM – SRM CPU Management

Function: See the entry points for the function and purpose description.

Entry-Point IRARMAP1: Automatic priority group recorder processing.

Callers: IRARMCTL and time queue processor.

Entry-Point IRARMAP2: ASCBCHAP standard interface routine.

Callers: IRARMCTL and Algorithm request processor.

Entry-Point IRARMACT: Activate dispatching priority control algorithms.

Callers: None

Entry-Point IRARMEQ1: ENQHOLD monitoring.

Callers: None

Entry-Point IRARMCL0: CPU load balancing system profile adjustment.

Callers: IRARMEVT and QSCE start SYSEVENT.

Entry-Point IRARMCL1: CPU utilization monitoring

Callers: IRARMCTL and Time queue processor.

Entry-Point IRARMCL3: CPU load balancing user evaluation.

Callers: IRARMCTL and partial analysis processor.

Exit-Normal: BR 14, no return code.

External-References:

- IRARMSRV
- IRARMI02
- IEAVELK
- IEAVLKSE

IRARMCPU – SRM CPU Adjustment Array

Function: This module contains model-dependent constants used by SRM to adjust some of its parameters.

Entry from: Not applicable (non-executable code).

Exit to: Not applicable (non-executable code).

IRARMCTL – SRM Control Algorithm

Function: This routine routes control to various SRM routines both as a result of explicit requests, and as a result of the expiration of internally scheduled periodic intervals. The routine defers the routing of control to certain requested routines until concurrently executing SRM processing completes. SRM control includes the routines that analyze the system users to determine which should be swapped in or out.

Entry from: IRARMEVT when a SYSEVENT has been received, from IRARMFIP when a timed algorithm is due to be run, and from the dispatcher for the execution of an SRM SRB.

Exit to: IRARMEVT or IRARMINT.

IRARMERR – SRM Functional Recovery

Function: This module provides functional error recovery for SRM.

Entry from: Recovery termination manager.

Exit to: Recovery termination manager.

IRARMEVT – SRM SYSEVENT Routers and Processors

Function: This module translates information describing changes in the status of an address space or in the system environment from an external representation to an internal representation that can be acted upon by SRM. Also performs a limited number of specific services for other components.

Entry from: IRARMINT.

Exit to: IRARMINT or IRARMCTL.

IRARMEV2 – SRM Expansion Module

Function: This module performs the SRM device allocation function by selecting a single DASD or tape unit from an input list of candidate devices.

Entry from: For entry IRARME28: SYSEVENT Processor.

Exit to: IRARMCTL or IRARMI01.

IRARMFIP – SRM Fast Interface Path

Function: This module performs the processing for those SYSEVENTs whose execution frequency or short path length makes the generalized path of IRARMINT unnecessary.

Entry from: Any issuer of a SYSEVENT predefined as a candidate for a fast path event.

Exit to: The SYSEVENT issuer, IRARMCTL, or IRARMINT.

IRARMFMT – SRMDATA Print Dump Format Exit

Function: If the SRMDATA format control verb is specified in the print dump input, this module formats the three SRM swap queues of user control blocks (OUCBs). They are the WAIT, OUT, and IN queues. It also formats the domain table (DMDT), which controls the system's multiprogramming level (MPL).

Entry from: AMDPRUIM (an AMDPRDMP utility module).

Exit to: AMDPRUIM.

IRARMICS – System Resources Manager Installation Control Specification

Function: Scans the list in the IEAICSxx member of parmlib, checks its syntax, and builds a set of contiguous control blocks in subpool 245. The control blocks control the assignment of performance group numbers to MVS users according to any of these factors: subsystem name, transaction name, userid, transaction class, or account number.

Entry-Point: IRARMICS

Callers: IEEMB812 via CALL

Exit-Normal: Return to caller via BR 14

Exit-Error: None

External-References:

Routines:

- IRARMANL - to analyze the syntax of the member
- IRARMRDR (in caller's module) - to read and record from parmlib
- IRARMMWR - to write a message
- IRARMFMA - to free storage

Data Areas

- SIZDATD field in IRARMANL - length of IRARMANL work area
- Message fields in IRARMIPM - to issue syntax error messages to the system log

Tables

- ICSYNTAB - syntax table for the ICS
- PGNARRAY - array of selected PGNS
- RPGNARAY - array of selected RPGNS

IRARMINT – SRM Interface Program

Function: This module performs the necessary processing for most branch entry and SVC SYSEVENTS to permit information pertaining to individual address spaces or system resources to be passed to and retrieved from SRM.

Entry from: Most SYSEVENT issuers (branch type SYSEVENT), SVC processors (SVC type SYSEVENT), IRARMEVT, or IRARMCTL.

Exit to: Most SYSEVENT issuers (branch type SYSEVENT), type 1 SVC exit routine (SVC type SYSEVENT), or the dispatcher (if an SVC entry and an SRB must be scheduled immediately).

IRARMIOM – SRM I/O Management

Function: This module consists of a set of routines that monitor the I/O logical path usage of certain address spaces. These routines recommend address spaces for swapping based upon the extent to which the swap-in or swap-out of the address space would correct a detected I/O imbalance.

Entry from: IRARMCTL.

Exit to: IRARMCTL.

IRARMIPM – IPS/Installation Control Specification/OPT Message Module

Function: This module contains the text for syntax error messages issued by IRARMIPS, IRARMICS, and IRARMOPT.

Entry from: Not applicable.

Exit to: Not applicable.

IRARMIPS – SRM IPS Processor

Function: This module scans the installation performance specification (IPS) list in the IEAIPSxx member of SYS1.PARMLIB and builds control blocks for IPS information.

Entry from: IEEMB812.

Exit to: IEEMB812.

IRARMMSG – SRM Message Module

Function: This module contains the WTO list macro forms of SRM messages.

Entry from: Not applicable.

Exit to: Not applicable.

IRARMOPT – SRM OPT Processor

Function: This module scans the OPT list in the IEAOPTxx member of SYS1.PARMLIB and builds a parameter list containing the new values for the OPT variables.

Entry from: IEEMB812.

Exit to: IEEMB812.

IRARMRMR – SRM Resource Monitor

Function: This module accumulates several system resource contention indicators, computes the average utilization of these resources and based on the level of utilization, raises or lowers the system multiprogramming level (MPL). It also adjusts system maximum think time for logical swapping.

Entry from: IRARMCTL.

Exit to: IRARMCTL.

IRARMSET – SRM Non-Resident Set to New-PARMLIB Member

Function: This module replaces the internal SRM constants and control blocks with information from the appropriate PARMLIB member (IEAIPSxx, IEAICSxx, or IEAOPTxx) in response to SET IPS, SET ICS, or SET OPT.

Entry from: IRARMEVT.

Exit to: IRARMEVT.

IRARMSRV – SRM Supervisor Service Request Routine

Function: This module requests the invocation of specific supervisor services for SRM routines. The supervisor services requested include reordering the ASCB chain, stealing page frames, and obtaining or freeing storage in the SRM storage pool.

Entry from: An SRM routine.

Exit to: The invoking routine.

IRARMSTM – SRM Storage Management

Function: Consists of a set of routines that control the use of main storage by all address spaces. For non-swappable users, the mechanism of page stealing is used for storage management control; for swappable users, both page stealing and swapping provide the requisite control.

Entry from: IRARMCTL and IRARMEVT.

Exit to: IRARMCTL and IRARMEVT.

IRARMST2 – SRM Storage Management Subroutine Module

Function: See the function and purpose description for each entry point.

Entry Point IRARMMS2 Real page shortage prevention.

Callers: IRARMCTL and algorithm request processor.

Entry-Point IRARMMS4: Threshold adjusting algorithm for storage load balancing and pageable storage shortage processing.

Callers: IRARMCTL and algorithm request processor.

Entry-Point IRARMMS5: Fixed frame threshold calculation.

Callers: IRARMCTL and time queue processor.

Entry-Point IRARMMS6:

Callers: IRARMCTL and time queue processor.

Entry-Point IRARMSQA: IRARMCTL, algorithm request processor, and time queue processor.

Exit-Normal: BR 14, no return code.

External-References:

- IRARMCTL
- IRARMCRL
- IRARMCSO
- IRARMRPS
- IRARMSRV
- IRARMI09
- IRARMI20
- IRARMWL4
- IRARMWM3
- IARYRPMG

IRARMSWP – Swap Algorithm

Function: This module includes the routine which determines which users in which domains should be swapped in or out.

Entry-Point IRARMCAP: This routine attempts to keep the multiprogramming level (MPL) per domain at its target level. The target MPL per domain is the desired number of memories belonging to that domain to be kept in core. IRARMCAP attempts to maintain the MPS per domain at its target by performing swaps. The swaps which are performed are deferred swap-ins, unilateral swap-outs, express swap ins, unilateral swap-ins and exchange swaps. Domain table initialization which is also performed in the swap analysis routine will only be done if any of 3 conditions exist:

1. If the domain table reinitialization bit (RMCTDMTI) is set.
2. If the redrive count (RMCAREDR) is greater than zero.
3. If at least 30 real seconds have elapsed from the last initialization.

Callers: IRARME13, IRARME15, IRARME24, SWINFL (SYSEVENTS), IRARMCPO, IRARMCPI (IRARMSWP), IRARMMS2 (IRARMST2), IRARMHIT (IRARMWL4), and IRARMCEL (IRARMCTL).

Exit-Normal: Branch on Register 14.

Entry-Point IRARMSA7: The Deferred Migration Swap Algorithm redrives the deferred migration users which have been physically swapped out and on the wait queue. It attempts to swap in these users if they fit in storage and if they do not, raises the thresholds so they will fit.

Callers: IRARME17 (SYSEVENT), IRARMSA7 (IRARMSWP), and IRARMCEL (IRARMCTL).

Exit-Normal: Branch on Register 14

Entry-Point IRARMCPI: IRARMCPI is the routine which scans the OUT queue and selects candidates to be swapped in. It chooses the 2 best candidates with the highest swap recommendation value within a domain.

Callers: IRARMCAP and IRARMHIT.

Exit-Normal: Branch on Register 14 and OUCB address is non-zero.

Exit-Error: OUCB address is zero.

Entry-Point IRARMCPI: IRARMCPO is the routine which scans the IN queue and selects a candidate to be swapped out. It chooses the best candidate with the lowest swap recommendation value within a domain.

Callers: IRARMCAP (IRARMSWP).

Exit-Normal: Branch on Register 14. OUCB address is non-zero.

Exit-Error: OUCB address is zero.

Entry-Point IRARMCVL: IRARMCVL computes a numerical value representing the recommendation of a user to be swapped in, swapped out or to remain in core. The recommendation value is the composite sum of the user's normalized workload level and the recommendation of the I/O and CPU resource Managers.

Callers: IRARMCIP and IRARRMCPO (IRARMSWP).

Exit-Normal: Branch on Register 14.

Entry-Point IRARMRPT: IRARMRPT is called when there is not enough room for an address space to be brought into processor storage. In order to make room RPT raises the reals and/or expanded thresholds to accumulate more free frames to try to swap in this to swap this address space on the next attempt.

Callers: IRARME17, IRARMCAP and IRARMSA7.

Exit-Normal: Branch on Register 14.

Entry Point IRARMLPT: IRARMLPT lowers the real and expanded storage shortage thresholds by the amount by which they were raised in RPT on behalf of a user.

Callers: SWINFL, IRARME16, IRARMPR5, IRARMMS2 and DEFSWPIN.

Exit-Normal: Branch on Register 14.

Entry-Point IRARMCSI: IRARMCSI accepts a request that an address space should be swapped in or migrated. If the address space is logically swapped, then quiesce is posted and the swap is turned around. If the address space is to be swapped in, then an OUXB is allocated for the address space and the swap in is initiated. If the address space is to be migrated, then the migration is scheduled.

Callers: IRARME68, IRARMCAP, IRARMSA7, DEFSWPIN and IRARMHIT.

Exit-Normal: Branch on Register 14. Register 15 contains zero meaning swap-in initiated.

Exit-Error: Branch on Register 14. Register 15 contains eight meaning the address space was already swapped in.

Entry-Point IRARMCSO: IRARMCSO accepts a request for an address space to be swapped out. If the address space is already swapped out then return a return code. If the address space is logically swapped, then initiate the physical swap-out. If the address space is to be swapped out of core, then quiesce is posted and the address space is marked as going out.

Callers: IRARME02, IRARME03, IRARME08, IRARME18, IRARME19, EVTCSOD, IRARMASM.

Exit-Normal: Branch on Register 14. Register 15 contains zero meaning swap-out initiated.

Exit-Error: Branch on Register 14. Register 15 contains four meaning an error occurred while trying to post quiesce for address space. Register 15 contains eight meaning the address space was already swapped out.

External-References:

- IRARMWM3
- IRARMCL3
- IRARMIL3
- IRARMMS3
- IRARMI02
- IRARMI04
- IRARMI07
- IRARMI09
- IRARMI16
- IRARMMS2
- IRARMMGD
- IRARMMGI

IRARMWAR – SRM Workload Activity Recording Routine

Function: Collects workload activity data when so requested by RMF. Terminates the collection of workload activity data when so requested by RMF, or when the IPS or installation control specification is changed.

Entry from: IRARMEVT or IRARMWLM.

Exit to: IRARMEVT or IRARMWLM.

IRARMWLM – SRM Workload Manager Algorithm Module - Part 1

Function: Consists of a set of routines that update user transaction statistics, and monitor the workload level at which individual users are receiving service.

Entry from: IRARMCTL and IRARMEVT.

Exit to: IRARMCTL and IRARMEVT.

IRARMWL3 – SRM Workload Manager Algorithm Module Three

Function: This module consists of a collection of individually invoked subroutines. The logic for when these subroutines are called is mainly contained in various SYSEVENT processing routines within module IRARMEVT. A description of the function of each subroutine is provided in a separate introductory comment for that subroutine. The collective function provided by these subroutines is to provide services to the workload manager function.

Callers: None

Exit-Normal: Branch on Register 14 (for all subroutines).

Exit-Error: None

External-References:

- IRARMI04
- IRARMFPG
- IRARMWR9

IRARMWL4 – SRM Workload Manager Algorithm Module Four

Function: This module consists of a collection of individually invoked subroutines. The logic for when these subroutines are called is mainly contained in various SYSEVENT processing routines within module IRARMEVT. A description of the function of each subroutine is provided in a separate introductory comment for that subroutine. The collective function provided by these subroutines is to provide services to the workload manager function.

Callers: None

Exit-Normal: Branch on Register 14 (for all subroutines).

Exit-Error: None

External-References:

- IRARMWM4
- IRARMWM5
- IRARMRPS
- IRARMCAP
- IRARMCPI

IRARMWTO – WTO and DOM Message Handler

Function: Issues messages and deletes messages when requested.

Callers: IEEMB860.

Exit-Normal: None.

External-References: None.

ISGBCI – Global Resource Serialization Ring Processing Routine

Function: This module:

- Sends or receives commands and data to or from other systems in the global resource serialization complex.
- Adds or removes a system to or from the global resource serialization ring.
- Starts a one-system ring.
- Provides status information about the systems and CTCs within the global resource serialization complex.

Entry from: ISGBTC, ISGCMDR, ISGCQMRG, ISGCPRG, ISGCQSC, ISGCRST, ISGJFE, and ISGNGRSP.

Exit to: Caller via branch.

ISGBDR – Global Resource Serialization Timer Management Routine

Function: This module converts time intervals to timer format, handles timer interrupts for global resource sharing complexes, and calls timer management routines to check for overdue arrival of the main-ring RSA and to schedule an SRB when expiration of the main-ring RSA residence interval occurs.

Entry from:

For entry-point ISGBDR: ISGBTC and ISGBSM.

For entry-points ISGBDRM and ISGBDRS: The timer SLIH

For entry-point ISGBDRC: ISGBTC, ISGBCI, and ISGBSM.

For entry-point ISGBDRFR: RTM.

Exit to: Caller via branch.

Called routines: TPCSDIE, CVTQTE00.

ISGBERCV – Global Resource Serialization Ring Processing Recovery Routine

Function: This module is the ESTAE routine that provides recovery for ISGBTC and ISGBCI. ISGBERCV initializes the SDWA with the appropriate module information, verifies the GVT and GVTX control blocks, and takes an SVC dump. Exit for ISGBTC recovery is a retry to one of the retry addresses provided in the ESTAE parameter list. Percolation is done only when there is a verification failure of the GVT or GVTX control block. Exit for ISGBCI recovery is percolation. Retry is done only when an error occurred within ISGBTCIR.

Entry from:

For entry-point ISGBERCV: RTM to perform error recovery.

For entry-point ISGVFRSL: IEAVEQV1 to validate queue elements.

Exit to: Caller via BALR.

Called routines: ISGJFE.

ISGBFRCV – Global Resource Serialization Ring System Authority (RSA) Send/Receive Functional Recovery Routine (FRR)

Function: This module is a functional recovery routine (FRR) that provides recovery for ISGBSR. ISGBFRCV performs SYS1.LOGREC recording, verifies the GVT, GVTX, and RSV control blocks, takes an SVC dump, marks the RSL (CTC) as disabled, and/or causes a main-ring failure, and posts ISGBTC to write operator messages.

Entry from: RTM.

Exit to: Caller via branch.

ISGBPG – Global Resource Serialization Ring Permutator

Function: This module generates possible main-ring configurations when adding or removing a system from the global resource complex.

Entry from: For entry-points ISGBPGI and ISGBPGN: ISGBSR.

Exit to: Caller via branch.

ISGBRF

Function: The diagnosis information for this module is in *MVS/ESA Component diagnosis: Global Resource Serialization.*

ISGBSM

Function: The diagnosis information for this module is in *MVS/ESA Component diagnosis: Global Resource Serialization.*

ISGBSR – Global Resource Serialization Send/Receive Routine

Function: This module receives, processes, and sends the RSA and RSAIRCD messages to systems active in the global resource serialization complex. It also converts a numeric SYSID to SYSNAME by searching for a matching SYSID in the RVENTY table. When a match is found, the corresponding SYSNAME is passed to ISGQSCAN.

Entry from:

For entry-point ISGBSR: CTC driver.

For entry-point ISGBSRME: ISGBDR.

For entry-point ISGBSRIN: ISGQSCAN.

For entry-point ISGBSRNI: ISGCPRG, ISGQSCAN.

For entry-points ISGBBE and ISGBSRRI: ISGCQMRG, ISGBCI.

Exit to:

For entry-points ISGBSRIN, ISGBSRNI, and ISGBBE: Caller.

For entry-points ISGBSR, ISGBSRME, ISGBSRRI, ISGBSRR, ISGBSRSR: Dispatcher.

Called routines: CVTOPT01, ISGBDR, ISGBDRC, ISGBFRCV, ISGBPGI, ISGBPGN, ISGGQWB1, ISGJGVBF, ISGJFE, ISGJSNBF, and ISGSALC.

ISGBTC – Global Resource Serialization Ring Processing Task Mode Controller

Function: This module performs task mode services for ring processing. These services include ring initialization, handling unusual events, initiating a one-system ring, and sending a message or immediate CCW via a CTC adapter.

Entry from:

For entry-point ISGBTC: ISGNASIM.

For entry-point ISGBTCR1: ISGBCI.

For entry-point ISGBTCIR: ISGBCI, ISGBTC.

Exit to: Caller via branch.

Called routines: Dispatcher, ISGBCI, ISGBDR, ISGJFE, ISGJENF0, ISGMSG00.

ISGCDSP – Global Resource Serialization DISPLAY GRS Request Processor

Function: This module processes global resource serialization DISPLAY status requests and produces the ISG020I message. The global resource serialization command router (ISGCMDR) attaches ISGCDSP when it finds a command request block (CRB) for a DISPLAY request on the global resource serialization command request queue. If communication with ISGCMDR is not possible ISGCMDI attaches ISGCDSP.

Entry from: ISGCMDI and ISGCMDR.

Entry-Point: ISGCDS02.

Exit to: Caller.

Error exit X'09A' ABEND.

Called routines: IEECB808 at entry MSGSERV.

ISGCMDE – DISPLAY GRS Command Parser Exit

Function: As IEEMB887 parses each element of the DISPLAY GRS command, this module records positive identification (the keyword found) in a command request block (CRB) or error in a message in the extended save area (XSA).

Entry from: IEEMB877.

Exit to: Caller.

Called routines: None.

ISGCMDI – Global Resource Serialization Command Interface

Function: This module performs authority checking for the VARY GRS command and syntax checking for the VARY GRS and DISPLAY GRS commands.

Entry from: IEECB808.

Entry-point IEECB921: Processes the VARY command.

Exit to: Caller.

Error exit X'09A' ABEND.

Entry-point IEECB922: Processes the DISPLAY command.

Exit to: Caller.

Error exit X'09A' ABEND.

Called routines: ISGCDSP, ISGMSG00, IEE0503D, IEEMB887.

ISGCMDR – Global Resource Serialization Command Router

Function: This module attaches the message module and the restart, quiesce, purge, and display processors to process the VARY GRS and DISPLAY GRS commands. This module is also called at entry-point ISGCTXR1 to detach the command processor and release any storage it obtained.

Entry from: ISGNASIM.

Entry-points: ISGCTXR1, ISGCDRRV.

Exit to: Caller.

Error exit X'09A' ABEND.

Called routines: ISGCDSP, ISGCQSC, ISGCPRG, ISGCRST, ISGMSG00, ISGCBCI.

ISGCPRG — Global Resource Serialization VARY GRS PURGE Request Processor

Function: This module processes the PURGE parameter of the VARY GRS command. The PURGE parameter removes a system from the global resource serialization complex. ISGCPRG receives control from the command router (ISGCMDR) when a command request block (CRB) for a purge request is found on the global resource serialization command work queue.

Entry from: ISGCMDR.

Exit to: Caller.

Error exit X'09A' ABEND.

Called routines: ISGBCI, ISGGFRR1, ISGGQWB0, ISGMSG00, ISGSDAL.

ISGCQMRG — Global Resource Serialization Queue Merge

Function: This module ensures that the global resource queues for a system joining a global resource serialization complex are identical to those of the other active systems in the complex.

Entry from: ISGNGRSP, ISGCRST.

Exit to: Caller.

Called routines: IEA0PT01, ISGBSR, ISGBCI, ISGGQWB0, ISGGQSRV, ISGGRP00.

Error entry ISGCQMRR.

Error exit X'09A' ABEND.

ISGCQSC — Global Resource Serialization VARY GRS QUIESCE Request Processor

Function: This module processes the QUIESCE parameter of the VARY GRS command. The QUIESCE parameter removes a system from a global resource serialization ring. ISGCQSC receives control from the command router (ISGCMDR) when a command request block (CRB) for quiesce processing is found on the global resource serialization command work queue.

Entry from: ISGCMDR.

Exit to: Caller.

Error exit X'09A' ABEND.

Called routines: ISGBCI, ISGMSG00.

ISGCRCV — Global Resource Serialization Command Recovery Routine

Function: This module is the ESTAE/I routine used by the following global resource serialization command processing and initialization routines:

- ISGCDSP — DISPLAY GRS.
- ISGCMDI — Command interface.
- ISGCMDR — Command router.
- ISGCPRG — VARY GRS PURGE.
- ISGCQMRG — Queue merge.
- ISGCQSC — VARY GRS QUIESCE.
- ISGCRST — VARY GRS RESTART.
- ISGMSG00 — Message processing.
- ISGNASIM — Address space initialization.
- ISGNGRSP — GRS= Option processing.

ISGCRCV requests RTM to perform SYS1.LOGREC recording, takes SVC dumps, routes control to special exit routines, and releases storage for the failing module. This module then indicates to RTM whether a retry should be attempted or termination continued.

Entry from: RTM.

Entry-points ISGCRCVE, ISGCRCVC, ISGCRCVD, ISGCRCVR, ISGCRCVY.

Exit to: RTM.

Called routine IEAVEADV.

ISGCRET1 — Global Resource Serialization Router ERRET Routine

Function: This routine performs error processing when the cross memory POST for the routine that placed the CRB/MRB on the command request queue (CRQ) fails.

> For entry: ISGCRET1 — Its function is to take an SVC dump of the failing ECB and the LSQA of the requestor's address space.
>
> For entry: ISGCR1RV — The functional recovery routine for ISGCRET1. It performs the following tasks: releases any obtained storage, indicates to RTM to release the local lock and record the error, and returns to RTM to continue with termination.

Entry from:

> For entry ISGCRET1: POST.
> For entry ISGCR1RV: RTM.

Exit to:

For entry ISGCRET1: Caller.
For entry ISGCR1RV: Caller.

ISGCRETO – Global Resource Serialization Command Interface ERRET Routine

Function: This module performs error processing when the cross memory POST for ISGCMDR fails.

For entry: ISGCRET0 – Its function is to deactivate the Command Request Queue and take an SVC dump of the failing ECB, global resource serialization LSQA, and the global vector table (GVT).

For entry: ISGCRORV – The functional recovery routine for ISGCRET0. It performs the following tasks: deactivates the Command Request Queue, releases any obtained storage, indicates to RTM to release the local lock and record the error, and returns to RTM to continue with termination.

Entry from:

For entry ISGCRET0: POST.
For entry ISGCRORV: RTM.

Exit to:

For entry ISGCRET0: Caller.
For entry ISGCRORV: Caller.

ISGCRET1 – Global Resource Serialization Router ERRET Routine

Function: This module performs error processing when the cross memory POST for the routine that placed the CRB/MRB on the global resource serialization Command Request Queue fails.

For entry: ISGCRET1 – Its function is to take an SVC dump of the failing ECB and the LSQA of the requestor's address space.

For entry: ISGCR1RV – The functional recovery routine for ISGCRET1. It performs the following tasks: releases any obtained storage, indicates to RTM to release the local lock and record the error, and returns to RTM to continue with termination.

Entry from:

For entry ISGCRET1: POST.
For entry ISGCR1RV: RTM.

Exit to:

For entry ISGCRET1: Caller.
For entry ISGCR1RV: Caller.

ISGCRST – Global Resource Serialization VARY GRS RESTART Request Processor

Function: This module processes the RESTART parameter of the VARY GRS command. The RESTART parameter performs the following functions:

- Bring a new system into the global resource serialization complex.
- Restart global resource serialization on a quiesced system.
- Restart global resource serialization processing on one or more systems after a failure in the ring.

ISGCRST receives control from the command router (ISGCMDR) when a command request block (CRB) for a restart request is found on the global resource serialization command work queue.

Entry from: ISGCMDR.

Entry-Point: ISGCRS02.

Exit to: Caller.

Called routines: ISGBCI, ISGCQMRG, ISGMSG00.

ISGDGCB0 – Dump Global Resource Serialization Control Blocks

Function: This module moves the GVT, global resource serialization ASCB, major global resource serialization control blocks (GVTX, LQHT, LRPT, GQHT, GRPT, SAHT), RSV, and the active global resource serialization resource queue area pages for PQCBs, QCBs, QELs, and QXBs to the SDUMP output buffer.

Entry from: ISGDSDMP.

Exit to: Caller.

Called routines: None.

ISGDCONT – Global Resource Serialization ENQ Contention Analysis Exit

Function: The module provides ENQ contention data gathered from Queue Control Blocks (QCBs), Queue Elements (QELs) and Queue Extension Blocks (QXBs) contained in the Global Resource Serialization to the Contention Queue Element (CQE).

Entry from: BLSACGTH.

Exit to: Caller.

Called routines: AMDMEMAR, ADPLESRV and BLSUMPKN.

ISGDPDMP – Global Resource Serialization Print Dump Exit

Function: This module formats and prints information about the local and global resources known by this system and the requestors of those resources.

Entry from: AMDPRDMP.

Exit to: Caller.

Called routines: ISGDSORT, AMDMEMAR, AMDWRITR, and ADPLESRV.

ISGDQDMP

Function: The diagnosis information for this module is in *MVS/ESA Component diagnosis: Global Resource Serialization.*

ISGDRSAQ

Function: The diagnosis information for this module is in *MVS/ESA Component diagnosis: Global Resource Serialization.*

ISGDSDMP – Global Resource Serialization SVC Dump Exit

Function: This module writes the GVT, global resource serialization ASCB, major global resource serialization control blocks (GVTX, LQHT, LRPT, GQHT, GRPT, SAHT), the RSV, and the active global resource serialization resource queue area (RQA) pages for PQCBs, QCBs, QELs, and QXBs to the dump data set.

Entry from:

For entry-point ISGDSDMP: IEAVTSDU.

For entry-point ISGDSDR1 (a retry entry-point): RTM to continue with the dump.

For entry-point ISGDSDRV (the ESTAE routine for ISGDSDMP and ISGDGCB0): RTM to attempt recovery.

Exit to:

For entry-point ISGDSDMP: IEAVTSDU.

For entry-point ISGDSDR1: IEAVTSDU.

For entry-point ISGDSDRV: RTM specifying that a retry is to be performed to entry-point ISGDSDR1 or percolation is to occur.

Called routines: IEAVTSE0 and ISGDGCB0.

ISGDSNAP – Global Resource Serialization SNAP Dump

Function: This module formats and prints information about the local and global resources owned or requested by tasks in the current job step.

Entry from:

For entry-point ISGDSNAP: IEAVAD01.

For entry-point ISGDSNR1 (a retry entry-point): RTM when global resources have not yet been processed.

For entry-point ISGDSNR2 (a retry entry-point): RTM to clean up storage when local and global resources have been partially or completely processed.

For entry-point ISGDSNRV (the ESTAE routine for ISGDSNAP): RTM to attempt recovery.

Exit to:

For entry-point ISGDSNAP: IEAVAD01.

For entry-point ISGDSNR1: IEAVAD01.

For entry-point ISGDSNR2: IEAVAD01.

For entry-point ISGDSNRV: RTM specifying that retry is to be performed at entry-point ISGDSNR1 or ISGDSNR2, or percolation is to occur.

Error exit If the GQSCAN service routine returns a failing return code of 12 or 16, ISGDSNAP issues an X'90A' ABEND.

Called routines: ISGDSORT and IEAVAD01.

ISGDSORT – Global Resource Serialization Dump Sort Routine

Function: This module sorts resource information blocks (RIBs) in ascending order according to QNAMEs/RNAMEs and chains them together. ISGDSORT starts by assuming that the first RIB is the first sorted RIB. Each subsequent RIB is placed in the correct position on the previously processed chain of RIBs which are chained via the RIBCHAIN field.

Entry from: ERBGSENQ, ERBGENQR, ISGDPDMP, and ISGDSNAP.

Exit to: Caller.

Called routines: None.

ISGGDEQP – DEQ Purge Processing

Function: This module purges the resources associated with a terminating task or ASID. ISGGDEQP identifies the resources to be purged, then branches to ISGGNQDQ (at entry-point ISGGDQ00) where the purge is done.

Entry from: ISGGTRM1 and ISGGRP00.

Exit to: Caller.

Called routines: ISGGNQDQ, ISGGPGRG, and ISGSDAL.

ISGGEST0 – ENQ/DEQ/RESERVE Cleanup For Requestors In WAIT

Function: This module is the cleanup routine for requestors who were put in an RB wait by ENQ/DEQ/RESERVE processing. In error cases, this module terminates the request for resources, frees any resources obtained, and percolates.

Entry from: Any user of ENQ/DEQ/RESERVE services through ISGGWAIT.

Exit to: Caller.

Called routines: ISGGNQDQ, ISGGQWB0, ISGSALC, ISGSDAL.

ISGGFDEQ

Function: The diagnosis information for this module is in *MVS/ESA Component diagnosis: Global Resource Serialization.*

ISGGFRR0 – FRR Recovery Routine for ENQ/DEQ/RESERVE

Function: This module determines which process failed, saves the error information, issues an SDUMP, repairs the resource queues, and, if possible, retries.

Entry from:

For entry-point ISGGFRR0: RTM to perform error recovery.

For entry-point ISGGFRR1: ISGCPRG, ISGGQWBR, or ISGQSCNR to validate and repair storage manager resources.

For entry-point ISGGFRRV: IEAVEQV1, IEAVEQV2, IEAVEQV3 to validate queue elements.

Exit to: Caller.

Called routines: IEAVEQV1, IEAVEQV2, IEAVEQV3, IEA0PT01, ISGSALC, and ISGSDAL.

Error entry None.

ISGGLUEX – Global Resource Serialization Addressing Mode Interface Module

Function: This module returns to the caller of ISGGREX0 with the correct addressing mode (AMODE).

Entry from:

For entry-point ISGGLUE1: ISGGREX0 to return to ISGGQWBI with the correct AMODE.

For entry-point ISGGLUE2: ISGGREX0 to return to ISGLNQDQ with the correct AMODE.

Exit to: Caller of ISGGREX0 (ISGGQWBI or ISGLNQDQ).

Called routines: None.

ISGGNQDQ – ENQ/DEQ/RESERVE Processing

Function: This routine processes all ENQ, DEQ, and RESERVE requests that can not be handled by the ENQ/DEQ Fast Path Routine (ISGLNQDQ). ISGGNQDQ checks that the request is valid and if it is processes it. An invalid request results in an ABEND.

Entry from:

For entry-point IGC048: Any DEQ user.

For entry-point IGC056: Any ENQ user.

For entry-point ISGGNQ00: ISGGRP00 for a global ENQ request.

For entry-point ISGGDQ00: ISGGRP00 for a global DEQ request.

For entry-point ISGGDQ00: ISGGDEQP for a DEQ purge request.

Exit to: Caller.

Called routines: ISGGPGRP, ISGGQWBI, ISGSALC, ISGGQWB0, ISGGWAIT, ISGSDAL, ISGSSAH, ISGGQWBC, RMF (via CVTRMFPT), IEAVEPD0, IGC07902, IRARMI00, IEAVGM00, and IEA0PT01.

ISGGPGRP – QEL Group Processing

Function: This module processes QEL groups for task requests.

Entry from: ISGGDEQP, ISSGGNQDQ.

Exit to: Caller.

ISGGQCSR

Function: The diagnosis information for this module is in *MVS/ESA Component diagnosis: Global Resource Serialization.*

ISGGQSRV – Global Queue Services

Function: This module is called in three situations: when global QCBs need to be processed by ISGCQMRG, when a global QCB has been processed by ISGCQMRG, and when global QCBs are to be dequeued by ISGCQMRG.

Entry from: ISGCQMRG at entry-points ISGGQS01, ISGGQS02, and ISGGQS03.

Exit to: Caller.

Called routines: ISGGQWB0, ISGBBE, ISGGRP00, IEAVEADV, and ISGSDAL.

Error entry For entry-points ISGGRECV and ISGGRTRY: RTM.

ISGGQWBC – QWB Copy Routine

Function: This module copies SQA queue work blocks (QWBs) from the SQA into global resource serialization address space's private area QWBs. It then places the copied QWBs on ISGGNQDQ's internal hold queue for further processing.

Entry from: ISGGNQDQ.

Exit to: Caller.

Called routines: ISGSDAL.

ISGGQWBI – Global Resource Serialization QWB Initialization

Function: This module moves the requestor's parameter element list (PEL) to the SQA queue work block (QWB) and establishes addressability to the global resource serialization address space.

Entry from: ISGGNQDQ.

Exit to: Caller.

Called routines: ISGGRCEX, ISGGSEEX, ISGGSIEX (entry-points in ISGGREX0) and ISGSALC.

ISGGQWB0

Function: The diagnosis information for this module is in *MVS/ESA Component diagnosis: Global Resource Serialization.*

ISGGREX0 – Global Resource Serialization Resource Exit Routine

Function: This module searches a specified resource name list (RNL) for a given resource name. If a match is found, ISGGREX0 sets a return code of zero, otherwise, it sets a return code of 4. ISGGWBI or ISGLNQDQ calls this module to determine if a resource is to be included in global resource serialization, excluded from global resource serialization, or converted from a hardware RESERVE to a global ENQ.

Entry from:

For entry-point ISGGSIEX: ISGGQWBI or ISGLNQDQ to scan the SYSTEM (Scope) inclusion RNL to determine if the specified resource name is to be included in global resource serialization.

For entry-point ISGGRCEX: ISGGQWBI to scan the RESERVE conversion RNL to determine if the RESERVE request is to be converted to a global ENQ request.

For entry-point ISGGSEEX: ISGGQWBI to scan the SYSTEMS (Scope) exclusion RNL to determine if the specified resource name is to be excluded from global resource serialization.

Exit to: Caller via ISGGLUEX.

Called routines: None.

Error entry None.

ISGGRP00

Function: The diagnosis information for this module is in *MVS/ESA Component diagnosis: Global Resource Serialization.*

ISGGSRVI — Global Resource Serialization Mainline Service Routine Initialization

Function: This module initializes the entry-points for all the global resource serialization service routines used to provide global resource serialization services to the system.

Entry from: ISGNASIM.

Exit to: Caller.

Called routines: None.

Error entry None.

ISGGTRM0 — ENQ/DEQ/RESERVE Termination Resource Manager - Stage 1

Function: This module is called by RTM whenever a task or address space is terminated. This module handles the setup and cleanup processing for ISGGTRM1. (ISGGTRM1 does the actual purge of all local and global resources acquired by the terminating task or address space.)

Entry from: RTM at entry-point ISGGTRM0 (alias IEAVENQ2).

Exit to: Caller via a BR14.

Called routines: ISGGTRM1.

Error entry None.

ISGGTRM1 — ENQ/DEQ/RESERVE Termination Resource Manager - Stage 2

Function: This routine purges all local and global resources acquired by the terminating task or address space. The control blocks are returned to the appropriate storage management pools.

Entry from:

For entry-point ISGGTRM1: ISGGTRM0 by a PC instruction.

For entry-point ISGGTRMR: ISGGFRR0 via RTM.

Exit to: ISGGTRM0 by a PT instruction.

Called routines: ISGGDEQP, ISGGQWB0, ISGSALC, IEA0PT01, ISGSDAL.

Error entry ISGGTRMR.

ISGGWAIT — Global Resource Serialization Wait Routine

Function: This module is an interface to IEAVWAIT for ISGGQWB0, ISGLNQDQ, and ISGGNQDQ, which cannot invoke IEAVWAIT directly because of addressability restrictions. ISGGWAIT:

- Saves the caller's registers in the TCB and the resume PSW in the RB.
- Establishes addressability to the home address space.
- Establishes an ESTAE (pointing to ISGGEST0) to cover the WAIT if called at ISGGWTO2.
- Calls the wait service routine (IEAVWAIT) to put the current RB in a wait state.

Entry from:

For entry-point ISGGWT01: ISGGQWB0.

For entry-point ISGGWT02: ISGGNQDQ and ISGLNQDQ.

For entry-point ISGGWT03: RTM to recover from an error in ISGGNQDQ or ISGLNQDQ.

Exit to: Caller.

Called routines: IEAVWAIT and RESTAE.

ISGJDI – Global Resource Serialization CTC Driver DIE

Function: This module contains disabled interrupt exits (DIEs) and error-handling exits. They receive control from the I/O supervisor (IOS) to complete the processing for channel programs on a CTC dedicated to global resource serialization.

Entry-Point DI1000: This DIE performs processing for I/O that uses the sense IOSB.

Entry from: IOSVIRBA using BASSM.

Exit to: Caller.

Called routines: IOSVLEVL.

Entry-Point DI2000: This DIE processes I/O completions for write channel programs, which use the write IOSB.

Entry from: IOSVIRBA using BASSM.

Exit to: Caller.

Called routines: IOSVLEVL.

Entry-Point DI3000: This DIE processes I/O completions for read channel programs, which use the read IOSB.

Entry from: IOSVIRBA using BASSM.

Exit to: Caller.

Called routines: IOSVLEVL.

Entry-Point ABN0000: This exit handles abnormal I/O completions that do not indicate incorrect length. It determines if LOGREC recording should be done.

Entry from: IECVPST using BASSM.

Exit to: Caller.

Called routines: IOSVLEVL.

Entry-Point NRM0000: This exit handles I/O completions that indicate incorrect length.

Entry from: IECVPST using BASSM.

Exit to: Caller.

Called routines: IOSVLEVL.

Entry-Point PGAD000: This exit allows the global resource serialization CTC driver to free the IOSB.

Entry from: IECVPST using BASSM or the dispatcher.

Exit to: Caller.

Called routines: IOSVLEVL.

ISGJENF0 – Global Resource Serialization Event Notification Exits

Function: This module establishes three event notification "listen" exits (entry-points ISGJENF1, ISGJENF2, and ISGJENF3) to allow global resource serialization to process a global resource serialization CTC that is enabled or disabled through the use of the VARY ONLINE/OFFLINE operator command.

Entry from: ISGBTC.

Exit to: Caller.

Called routines: IEFENFFX.

Entry-Point ISGJENF1: Sets the UCB to allow a VARY *device*, OFFLINE (where *device* is or is not a global resource serialization CTC) to complete.

Entry from: IEFENFNM (event notification facility).

Exit to: Caller.

Called routines: ISGMSG00.

Entry-Point ISGJENF2: Notifies global resource serialization that a VARY *device*, OFFLINE for a global resource serialization CTC has completed.

Entry from: IEFENFNM (event notification facility).

Exit to: Caller.

Called routines: ISGMSG00.

Entry-Point ISGJENF3: Notifies global resource serialization that a VARY *device*, ONLINE request for a global resource serialization CTC has completed.

Entry from: IEFENFNM (event notification facility).

Exit to: Caller.

Called routines: ISGMSG00.

Entry-Point ISGJENFR: Is an ESTAE that provides recovery for ISGJENF0. ISGJENFR records any error that occurs in ISGJENF0 in SYS1.LOGREC, dumps the storage associated with the error, issues an error message notifying the operator of the error, and retries ISGJENF0 at entry-point ISGJRTRY.

Entry from: RTM.

Exit to: Caller.

Called routines: SVC dump, ISGMSG00.

Entry-Point ISGJRTRY: Is the retry routine for ISGJENF0; the retry routine deletes any existing recovery environment and returns to the caller.

Entry from: RTM.

Exit to: Caller

Called routines: None.

ISGJFE – Global Resource Serialization CTC Driver Front End

Function: This module contains entry-points that initialize global resource serialization control blocks and provide services, related to the CTC driver, for other global resource serialization modules.

Entry-Point ISGJFE: This entry determines the amount of storage needed for the global resource serialization CTC driver queueing element (GCQ) and the global resource serialization CTC driver vector table (GCV).

Entry from: ISGBTC using BALR.

Exit to: Caller.

Called routines: None.

Entry-Point ISGJIDEN: This entry initializes the global resource serialization CTC driver vector table (GCV).

Entry from: ISGBTC using BALR.

Exit to: Caller.

Called routines: None.

Entry-Point ISGJCNCT: This entry initializes the global resource serialization CTC driver link control block (GCL) for the specified CTC. This entry verifies the paths to the CTC and marks its UCB as online and allocated.

Entry from: ISGBTC using BALR.

Exit to: Caller.

Called routines: IECVIOPM and IOSVLEVL.

Entry-Point ISGJMGCL: This entry marks a GCL as the main ring CTC, as a non-main ring CTC, or as an inoperative CTC when a system error occurs.

Entry from: ISGBTC using BALR.

Exit to: Caller.

Called routines: None.

Entry-Point ISGJGVSR: This entry makes an SRB available and schedules it if an unusual event has occurred.

Entry from: ISGBTC using BALR.

Exit to: Caller.

Called routines: None.

Entry-Point ISGJGTUE: This entry checks the global resource serialization CTC driver link control block (GCL) for an unusual event to be reported.

Entry from: ISGBTC using BALR.

Exit to: Caller.

Called routines: None.

Entry-Point ISGJGVBF: This entry places a buffer on the GCL read queue for a CTC. This buffer is used to receive a message from the system at the other end of the CTC.

Entry from: ISGBSR and ISGBTC using BALR.

Exit to: Caller.

Called routines: None.

Entry-Point ISGJGTUE: Reports on unusual events that can occur during ring processing.

Entry from: ISGBTC.

Exit to: Caller.

Called routines: None.

Entry-Point ISGJMGCL: Indicates whether the GCL should be defined as a main ring CTC or not.

Entry from: ISGBTC.

Exit to: Caller.

Called routines: None.

Entry-Point ISGJSRBX: This entry sets up the input registers (for ISGBSR) with values obtained from the global resource serialization CTC driver queueing element (GCQ).

Entry from: SRB dispatcher (IEAVEDS0) using a branch.

Exit to: ISGBSR.

Called routines: None.

Entry-Point ISGJSNBF: This entry obtains and initializes the write IOSB. It starts the I/O to write to the system at the other end of the CTC.

Entry from: ISGBSR using BALR.

Exit to: Caller.

Called routines: None.

Entry-Point ISGJTKBF: This entry removes a global resource serialization CTC driver queueing element (GCQ), and its associated buffer, from the GCL queue. ISGJTKBF makes the GCQ and buffer available to the caller.

Entry from: ISGBCI and ISGBTC using BALR.

Exit to: Caller.

Called routines: None.

Entry-Point ISGJDCNC: This entry halts any I/O in progress for the specified CTC.

Entry from: ISBGCI using BALR.

Exit to: Caller.

Called routines: None.

Entry-Point ISGJRJFE: This entry frees the write IOSB if recovery processing got control while entry-point ISGJSNBF was executing.

Entry from: ISGBFRCV.

Exit to: Caller.

Called routines: IEAVEADV.

ISGJRCV – Global Resource Serialization CTC Driver DIE Functional Recovery Routine

Function: This module is the functional recovery routine (FRR) that gets control if an error occurs during ISGJDI processing. ISGJRCV validates control blocks, records error information in the SDWA, provides a dump, and frees the IOSB.

Entry from: RTM.

Exit to: Caller.

Called routines: IEAVEADV.

ISGLNQDQ – ENQ/DEQ Fast Path Routine

Function: This routine provides a fast path for some local ENQ/DEQ requests. Requests that do not meet ISGLNQDQ's criteria are passed on to ISGGNQDQ. If any errors occur in processing, or if ISGGFRR0 is attempting a retry, the request is passed on to ISGGNQDQ.

Entry from:

For entry-point ISG048FP: A DEQ user.
For entry-point ISG056FP: An ENQ user.

Exit to: Caller.

Called routines: ISGGNQDQ, ISGGREXO, RMF (via CVTRMFPT), IRARMI00, IEAVEPD0, and IEA0PT01.

ISGMSG00 – Global Resource Serialization Message Processor

Function: This module provides the global resource serialization services with the ability to communicate with the operator. ISGMSG00 issues both informational (WTO/MLWTO) and reply (WTOR) messages.

Entry from:

For entry-point ISGMSG00: ISGBCI, ISGBTC, ISGCMDR, ISGNASIM, ISGNGRSP, ISGCMDI, ISGCRST, ISGCQSC, ISGCPRG, ISGJENF0.

For entry-point ISGMSGRT: RTM.

Exit to: Caller.

Error exit: X'09A' ABEND.

ISGNBRSE The diagnosis information for this module is in *MVS/ESA Component diagnosis: Global Resource Serialization.*

ISGNFINT The diagnosis information for this module is in *MVS/ESA Component diagnosis: Global Resource Serialization.*

ISGQSCAN – Global Resource Serialization Queue Scanning Service Module

Function: This module searches the resource queues for information about resources and requestors of those resources.

Entry from: Issuers of the GQSCAN macro instruction.

Exit to: Caller.

Error exit X'09A' ABEND.

Called routines: ISGBSR at entry-points ISGBSRIN and ISGBSRNI, ISGSALC, ISGSDAL, and ISGSHASH at entry-point ISGSGLH.

ISGQSCNR – Global Resource Serialization Queue Scanning Service FRR Routine

Function: If an error occurs while ISGGSCAN (the global resource serialization queue scanning service routine) is processing, ISGQSCNR receives control to release any control blocks or locks obtained by ISGQSCAN. It returns the control blocks to the storage manager. For unexpected errors, ISGQSCNR requests an SVC dump and records the error in SYS1.LOGREC.

Entry from: RTM.

Exit to: Caller.

Called routine IEAVEADV, ISGSDAL, and ISGGFRR1.

ISGSALC – Global Resource Serialization Storage Management Allocation Routine

Function: This module allocates storage cells (control blocks) from pool extent blocks (PEXBs) in the resource queue area (RQA) for QWB, MRB, CRB, TWKA and HWKA cell type and in the extended resource queue area (ERQA) for QCB, QEL, QXB and PQCB cell type, of the global resource serialization address space. Global resource serialization modules use this routine to obtain storage for global resource serialization control blocks.

Entry from: ISGBSR, ISGBSM, ISGGDEQP, ISGGEST0, ISGGFRR0, ISGGNQDQ, ISGGQWB0, ISGGRP00, ISGGTRM1, ISGLNQDQ, ISGNASIM, ISGQSCAN, or ISGSMI.

Exit to: Caller.

Error exit X'09A' ABEND.

ISGSDAL – Global Resource Serialization Storage Management Deallocation Routine

Function: This module deallocates cells (control blocks) in pool extent blocks (PEXBs) in the resource queue area (RQA) for QWB, MRB, CRB, TWKA and HWKA cell type in the extended resource queue area (ERQA) for QCB, QEL, QXB and PQCB cell type, of the global resource serialization address space. When all the cells in a predetermined number of PEXBs have been deallocated, entry-point ISGSPRLS frees the frames containing those PEXBs.

Entry from:

For entry-point ISGSDAL: ISGBCI, ISGCMDR, ISGCQMRG, ISGCPRG, ISGGEST0, ISGGFRR0, ISGGNQDQ, ISGGQSRV, ISGGQWBC, ISGGQWBD, ISGGRP00, ISGGTRM1, ISGLNQDQ, ISGQSCAN, ISGQSCNR, or ISGSMI.

For entry-point ISGSPRLS: the dispatcher.

Exit to: Caller.

ISGSHASH – Global Resource Serialization Hashing Routines

Function: This module hashes a name (QNAME, RNAME) to derive an entry address in the global queue hash table (GQHT) or in the local queue hash table (LQHT). ISGSHASH also hashes a SYSID/ASID to derive an entry address in the SYSID/ASID hash table.

Entry from:

For entry-point ISGSGLH: ISGGNQDQ, ISGLNQDQ, ISGNASIM, or ISGQSCAN.

For entry-point ISGSSAH: ISGGNQDQ, ISGGRP00, or ISGGFRR0.

Exit to: Caller.

ISGSMI – Global Resource Serialization Storage Management Interface Routine

Function: This module provides a cross memory interface for a module not executing in the global resource serialization address space to either ISGSALC or ISGSDAL, which execute in the global resource serialization address space.

Entry from: ISGCMDI.

Exit to: Caller.

Called routines: ISGSALC and ISGSDAL.

ITRF000B – System Trace I/O TTE Format Routine

Function: ITRF000B is part of the system trace formatter. ITRF000B filters out (skips) or formats an input/output (I/O) trace table entry (TTE).

Callers: IEAVETEF.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

External-References: IEAVETFA, IEAVETPB entry-point in IEAVETFA.

ITFR000F — System Trace Dispatcher TTE Format Routine

Function: ITRF000F is part of the system trace formatter. ITRF000F filters out (skips) or formats a task dispatch (DSP), initial SRB dispatch (SRB), suspended SRB dispatch (SSRB), or WAIT dispatch (WAIT) trace table entry (TTE).

Callers: IEAVETEF.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

External-References: IEAVETFA, IEAVETPB entry-point in IEAVETFA.

ITRF0001 — System Trace Subchannel TTE Format Routine

Function: ITRF0001 is part of the system trace formatter. ITRF0001 filters out (skips) or formats a start subchannel (SSCH), modify subchannel (MSCH), halt subchannel (HSCH), clear subchannel (CSCH), or resume subchannel (RSCH) trace table entry (TTE).

Callers: IEAVETEF.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

External-References: IEAVETFA, IEAVETPB entry-point in IEAVETFA.

ITRF0003 — System Trace External Interrupt TTE Format Routine

Function: ITRF0003 is part of the system trace formatter. ITRF0003 filters out (skips) or formats an external interrupt (EXT), emergency signal external interrupt (EMS), service signal external interrupt (SS), external call external interrupt (CALL) or clock comparator external interrupt (CLKC) trace table entry (TTE).

Callers: IEAVETEF.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

External-References: IEAVETFA, IEAVETPB entry-point in IEAVETFA.

ITRF0005 — System Trace SVC TTE Format Routine

Function: ITRF0005 is part of the system trace formatter. ITRF0005 filters out (skips) or formats an SVC interrupt (SVC), SVC return (SVCR) or SVC error (SVCE) trace table entry (TTE).

Callers: IEAVETEF.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

External-References: IEAVETFA, IEAVETPB entry-point in IEAVETFA.

ITRF0007 — System Trace Program Interrupt TTE Format Routine

Function: ITRF0007 is part of the system trace formatter. ITRF0007 filters out (skips) or formats a program check (PGM) trace table entry (TTE).

Callers: IEAVETEF.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

External-References: IEAVETFA, IEAVETPB entry-point in IEAVETFA.

ITRF0009 — System Trace Event SLIP/PER TTE Format Routine

Function: ITRF0009 is part of the system trace formatter. The system trace table entry filter/formatter routine, IEAVETEF, calls ITRF0009 when IEAVETEF encounters an explicit trace table entry (TTE) with an explicit entry id (TTEXPID) of X'09' ITRF0009 filters out (skips) or formats a SLIP/PER (SPER) TTE.

Callers: IEAVETEF.

Exit-Normal: Return to the caller.

Exit-Error: To RTM via an ABEND SVC.

External-References: IEAVETFA, IEAVETPB.

ITRF0013 — System Trace Machine Check TTE Format Routine

Function: ITRF0013 is part of the system trace formatter. ITRF0013 filters out (skips) or formats a machine check (MCH) trace table entry (TTE).

Callers: IEAVETEF.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

External-References: IEAVETFA, IEAVETPB entry-point in IEAVETFA.

ITRF0015 – System Trace Restart TTE Format Routine

Function: ITRF0015 is part of the system trace formatter. ITRF0015 filters out (skips) or formats a restart (RST) trace table entry (TTE).

Callers: IEAVETEF.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

External-References: IEAVETFA, IEAVETPB entry-point in IEAVETFA.

ITRF0017 – System Trace ACR TTE Format Routine

Function: ITRF0017 is part of the system trace formatter. ITRTF0017 filters out (skips) or formats an automatic CPU recovery (ACR) trace table entry (TTE).

Callers: IEAVETEF.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

External-References: IEAVETFA, IEAVETPB entry-point in IEAVETFA.

ITRF0019 – System Trace Suspension TTE Format Routine

Function: ITRF0019 is part of the system trace formatter. ITRF0019 filters out (skips) or formats a suspension (SUSP) trace table entry (TTE).

Callers: IEAVETEF.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

External-References: IEAVETFA, IEAVETPB entry-point in IEAVETFA.

ITRF001B – System Trace Alter Trace TTE Format Routine

Function: ITRF001B is part of the system trace formatter. ITRF001B filters out (skips) or formats an alter trace (ALTR) trace table entry (TTE).

Callers: IEAVETEF.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

External-References: IEAVETFA, IEAVETPB entry-point in IEAVETFA.

ITRF007F – System Trace Default USR0 TTE Format Routine

Function: ITRF007F is part of the system trace formatter. ITRF007F filters out or formats a VSRn trace table entry (TTE) by invoking the default USRn TTE formatting routine (ITRFDEFU). ITRF007F is designed to be replaced by a user-written module if a format other than the default USRn format is desired.

Callers: IEAVETEF.

Exit-Normal: Returns to the caller.

External-References: ITFRDEFU.

ITRFBR – System Trace Branch and Address Space TTE Formatter Routines

Function: ITRFBR is part of the system trace formatter. ITRFBR formats branch and address space trace table entries (TTEs).

- The ITRFBR entry-point formats one or more consecutive 31-bit or 24-bit branch (BR) TTEs contained within a single trace work buffer. ITRFBR formats up to seven consecutive branch TTEs as a single output record.
- The ITRFPC entry-point formats a PC TTE.
- The ITFRPT entry-point formats a PT TTE.
- The ITRFSSAR entry-point formats a SSAR TTE.

Callers: IEAVETEF.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

Entry-Point ITRFPC: Formats a program call (PC) TTE.

Callers: IEAVETEF.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

Entry-Point ITRFPT: Formats a program transfer (PT) TTE.

Callers: IEAVETEF.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

Entry-Point ITRFSSAR: Formats a set-secondary-address-register (SSAR) TTE.

Callers: IEAVETEF.

Exit-Normal: Returns to the caller via BR 14.

Exit-Error: To RTM via an ABEND SVC.

External-References: IEAVETPB entry-point in IEAVETFA.

ITRFHEX – System Trace Default Hex/Explicit/USRn TTE Formatter Routines

Function: ITRFHEX contains three formatters:

- Entry-point ITRFHEX formats the remainder of the current trace buffer in the default hex format.
- Entry-point ITRFEXP formats the current TTE in the default explicit TTE format.
- Entry-point ITRFDEFU formats the current USRn TTE in the default USRn TTE format. The IBM-supplied USRn TTE formatting routines, ITRF0n7F (where n = 0-F), use this entry-point to perform their function.

Callers: IEAVETFC.

Exit-Normal: Returns to the caller.

Exit-Error: To RTM via an ABEND.

Entry-Point ITRFEXP: Formats the current TTE in default explicit TTE format. The trace entry formatter, IEAVETEF, calls this entry-point when it encounters an explicit TTE of an unknown type (therefore, the TTE does not have a standard formatter routine), or when the type is known but the formatter routine for the TTE determines that the TTE cannot be formatted normally because it has an invalid subtype or unexpected length.

Callers: IEAVETEF.

Exit-Normal: Returns to the caller.

Exit-Error: To RTM via an ABEND.

Entry-Point ITRFDEFU: Formats a USRn TTE on behalf of one of the IBM-supplied USRn TTE formatters (ITRF007F-ITRF0F7F).

Callers: ITRF0n7F (n = 0-F).

Exit-Normal: Returns to the caller.

Exit-Error: To RTM via an ABEND.

External-References: IEAVETFA, IEAVETPB.

ITT – Component Trace

Function: All modules and macros with this prefix are part of the component TRACE component. The diagnosis information for this component is in *MVS/ESA Component Diagnosis: Component Trace.*

ITV – Data-in-Virtual

Function: All modules and macros with this prefix are part of the data-in-virtual component. The diagnosis information for this component is in *MVS/ESA Component Diagnosis: Data-in-Virtual.*

MVS/ESA
Component Diagnosis
and Logic:
Module Descriptions

READER'S
COMMENT
FORM

LY28-1420-2

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note: ***Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.***

Possible topics for comment are:

Clarity   Accuracy   Completeness   Organization   Coding   Retrieval   Legibility

If you wish a reply, give your name, company, mailing address, and date:

____

____

____

____

What is your occupation? ____________________________________

How do you use this publication? ____________________________________

Number of latest Newsletter associated with this publication: ____________________

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

MVS/ESA Component Diagnosis and Logic: Module Descriptions

LY28-1420-2

S370-37

cut along this line

Reader's Comment Form

Fold and Tape

Please Do Not Staple

Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department D58, Building 921-2
PO Box 950
Poughkeepsie, New York 12602-9935

Fold and Tape

Please Do Not Staple

Fold and Tape

Printed in U.S.A.

IBM®

MVS/ESA
Component Diagnosis
and Logic:
Module Descriptions

READER'S
COMMENT
FORM

LY28-1420-2

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note: ***Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.***

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

____

____

____

____

What is your occupation? ______________________________

How do you use this publication? ______________________________

Number of latest Newsletter associated with this publication: ____________________

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

MVS/ESA Component Diagnosis and Logic: Module Descriptions

LY28-1420-2 S370-37

cut along this line

Reader's Comment Form

Fold and Tape Please Do Not Staple Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department D58, Building 921-2
PO Box 950
Poughkeepsie, New York 12602-9935

Fold and Tape Please Do Not Staple Fold and Tape

Printed in U.S.A.

IBM®

IBM®

Program Number
5685-001
5685-002

File Number
S370-37

LY28-1420-02

Printed in U.S.A.