

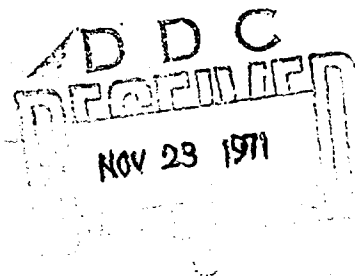
AD732682

COMPUTER MEMORY MANAGEMENT AT RAND

A. C. Shetler
S. Glaseman

May 1971

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
Springfield, Va. 22151



P-4585

Best Available Copy

106

CONTENTS

Section	
I. INTRODUCTION	1
II. INSTALLATION ENVIRONMENT	4
Hardware	4
Software	4
Workload Characteristics	4
Objectives	6
III. FIXED INITIATOR CONTROL OF MEMORY	7
Batch Processing	7
On-Line Processing	7
RJE Processing	11
IV. IMPLEMENTATION OF THE FIXED INITIATOR	16
Externals	16
Internals	16
V. CONCLUSION	23
APPENDIX	24

BLANK PAGE

COMPUTER MEMORY MANAGEMENT AT RAND*

A. C. Shetler

S. Glaseman

The Rand Corporation, Santa Monica, California

I. INTRODUCTION

When the Rand Computation Center (RCC) first put the 360/65 into production status (December 1968), the memory management considerations were confined to approximately 350K¹ of Processor Storage in which our job stream was multiprogrammed.² The problems of core fragmentation and appropriate scheduling algorithms for the efficient utilization of the computer resources were minimal since the environment was dynamic in the sense that any particular unfavorable situation lasted a relatively short period of time. The users in this batch processing environment were usually unaware of memory management problems as long as turnaround was not affected.

Approximately 18 months ago (September 1969) RCC added one million bytes of IBM LCS³ for use in supporting an on-line system for our users. When the LCS was added, there were immediate memory management problems needing resolution.

*Any views expressed in this paper are those of the authors. They should not be interpreted as reflecting the views of The Rand Corporation or the official opinion or policy of any of its government or private research sponsors. Papers are reproduced by The Rand Corporation as a courtesy to members of its staff.

Part of this paper was presented at SHARE XXXVI, March 8-12, 1971, Los Angeles.

¹One K is equivalent to 1024 bytes, or 256 words of system 360 storage; one byte has 4 bits; one word is composed of 4 bytes.

²Multiprogramming is a software mechanism whereby several jobs can execute concurrently in a computer.

³LCS is slower-access memory that can be added to a computer. The difference between the Processor Storage access speed and the LCS access speed is approximately 8 to 1; hence a job would use approximately eight times the number of CPU cycles to execute in LCS as would be required by the same job in Processor Storage.

The batch jobs had to be prevented from executing in LCS. Accidental execution in this slower-speed memory would have created severe accounting problems. It was also necessary to reserve storage for the interactive system when it was not resident. In the case of an ABEND operations would have to quiesce the operating system to restart the interactive system in the desired memory location. This action was certain to affect turnaround for the batch user and delay restart for the on-line user. Though not a primary consideration, the ability to reserve regions with specific memory locations for on-line applications was also found to be desirable. RCC anticipated having several users with interactive programs resident in the machine for several hours; with standard MVT memory management core fragmentation could have become a problem. By restricting these users to predefined regions, core fragmentation could be eliminated for these applications.

Hierarchy Support¹ was rejected as an alternative for managing memory because it did not solve the problem. After the special SYSGEN required, the special link-edits required for tasks using the Hierarchy Storage, and the resident memory required for Hierarchy Support, the problems previously defined still existed. The batch user, by using a JCL manual, could still execute a LCS, though now it would be purposeful. The LCS would still have to be quiesced to guarantee the interactive system storage placement. The core fragmentation from long-running interactive user tasks would still be a part of the Hierarchy Supported environment.

The May 27, 1969 issue of the *IBM Installation Newsletter* described a non-supported, release dependent, modification to the Initiator² which allowed memory locations to be assigned to a particular initiator. This was a relatively easy modification to implement, costing less than 100 bytes of non-resident storage, and appeared to solve our problems. This

¹Hierarchy Support is a SYSGEN option for OS/MVT. It provides for overt selection of the desired memory speed for program execution in slower-speed memory, and default selection of Processor Storage for normal job execution.

²The Initiator is an OS system task which handles the housekeeping activities associated with getting a job started and terminated in the system. An Initiator handles jobs sequentially in the system.

modification to the Initiator provided a mechanism whereby when the initiator is started it retains a region of a fixed size, executing the tasks it attaches in this fixed region.

Section II will describe the current environment of the Rand Computation Center. Section III will detail what kind of memory management is accomplished through fixed initiators. Section IV will describe the externals and internals of Fixed Initiator implementation and Sec. V will summarize the effect of this mechanism for memory management.

II. INSTALLATION ENVIRONMENT

A brief description of the RCC environment should aid in describing the production mode of operation using Fixed Initiators to Fence Core.

HARDWARE

Figure II-1 is a diagram of the hardware configuration as of March 1, 1971. There are multiple manufacturer's equipment attached to the IBM 360/65. A Calcomp 2314 disk device, Potter 7-track tape drives, and 1024K of AMPEX ECM¹ have successfully been running in RCC, at a significant cost savings.

SOFTWARE

The production software for the IBM 360/65 is OS/MVT, multiprogramming with a variable number of tasks. There are several local modifications to this system. Several user SVC's have been included for support of the Video Operating System (VOS), Conventional Programming System (CPS), and user systems. The initiator has been modified to allow fixed and floating initiators. Force Task Switching has been implemented to resolve lock-out problems for our on-line users (also applies to the batch environment). Prior to Release 19.3 of MVT there were Rand modifications to collect EXCP counts and high water mark for memory utilization. Release 19.3 saw the end of the accounting patches and the introduction of SMF² with extensive use of the user exits.

WORKLOAD CHARACTERISTICS

RCC, during March, is converting from Release 17 to Release 19.3 of MVT on a 360/65 with the usual accompaniment of peripheral devices.

¹AMPEX ECM is a faster slow-speed storage with an access speed of 2.8 to 1 compared to Processor Storage; that is, a task would take approximately 2.8 times the CPU to execute in ECM as compared to an execution in Processor Storage.

²System Management Facilities is an OS SYSGEN option to collect and record resource usage by job.

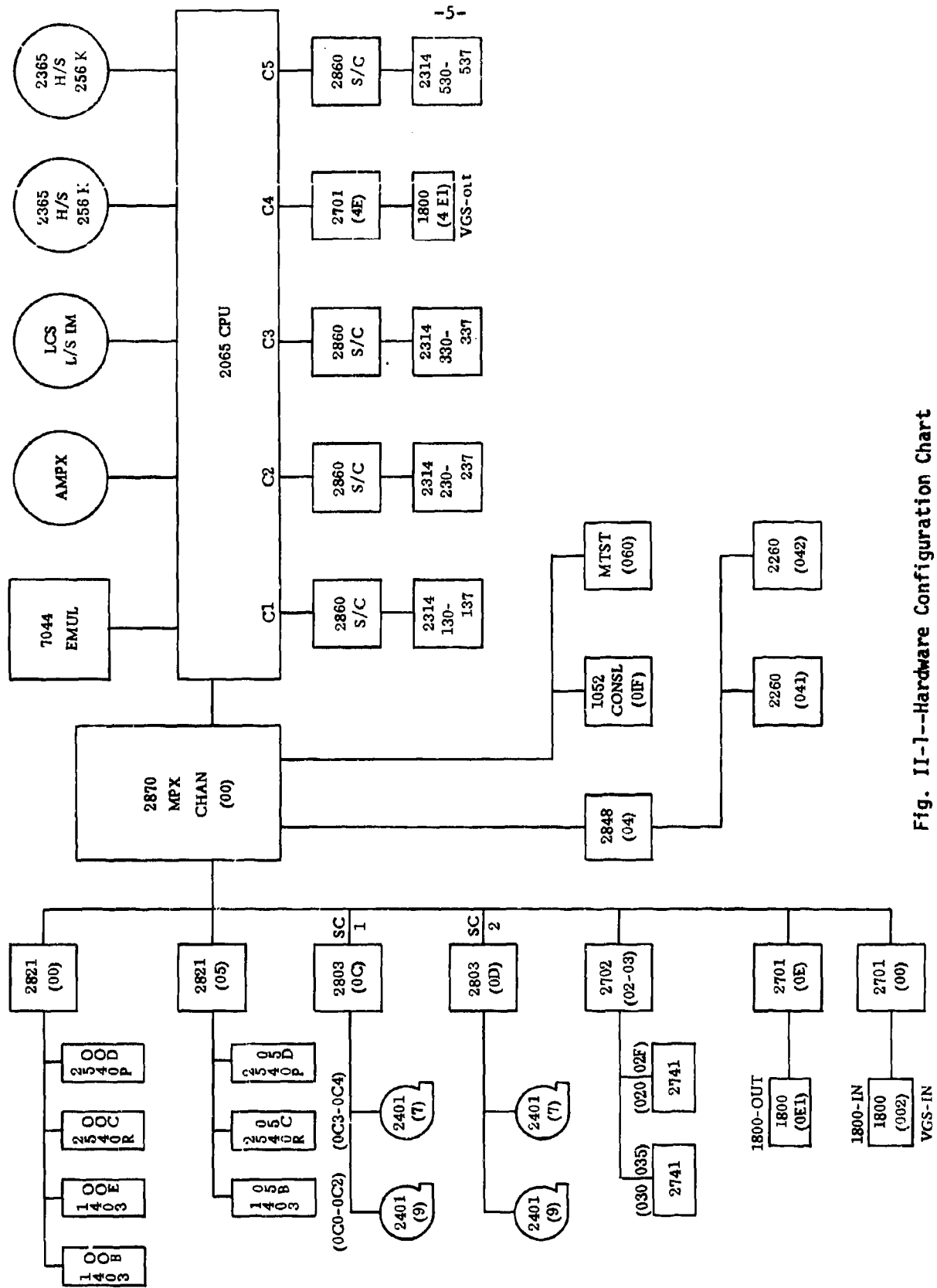


Fig. II-1--Hardware Configuration Chart

Over two hundred batch jobs are processed every day; these are logged in over the counter and returned via the PAL box. Over one hundred RJE jobs are processed every day; these are entered through the two on-line systems. Two on-line systems, having RJE facilities, are available for users between 9:00 a.m. and midnight daily. An IBM/7044 is emulated approximately 30 minutes a night. When measured, the CPU activity is above the 70 percent level.

OBJECTIVES

The Rand Computation Center has specific objectives directing services for our users. First, computing must be provided for corporate and research data processing. Second, hardware and software support must be provided for users developing new systems. And third, interactive services should be provided for the users, and when necessary services do not exist, these services are to be developed. These objectives result in one major constraint on our system. The hardware and software must be maintained in a state that permits easy implementation of new user systems; any system modifications imposing inflexibilities usually cannot be implemented. This restricts the improvements we can make on the system by the effect on the flexibility necessary for the users. And not unlike most computer installations, these objectives and constraints are subject to the overriding requirement for stability and reliability in the system.

III. FIXED INITIATOR CONTROL OF MEMORY

Having discussed the background for and the environment of our Fenced System, let us examine how it is used. Figure III-1 is a memory allocation diagram. There are two and a half million bytes of storage to support our Batch and On-Line/RJE processing. We have 512K of Processor Storage, 1024K of IBM LCS, and 1024K of AMPEX ECM.

BATCH PROCESSING

Figure III-2 describes the Batch Processing environment. In this 512K of Processor Storage, 350K is available for the Batch jobs. This is Rand's MVT environment with 3 or 4 active initiators. The Reader and Writer tasks are not resident in this storage; they reside in the slower-speed memory. Core Fencing controls the batch processing by preventing batch jobs from running in slower-speed storage.

ON-LINE PROCESSING

Figure III-3 describes the On-line Processing environment. There are three user tasks and one system task that make up the on-line systems.

The first of these user tasks is a Rand-developed system, VOS. The Video Operating System is a message switching task which performs communication between active OS tasks and the user at a Video Terminal via a 2701 data line between the IBM 1800 and 360/65. Through VOS a user at a Video Terminal can communicate with any or all on-line systems active in OS. And since VOS can provide message paths between the Video Terminal and any tasks in the system, a single user can be connected with several tasks at one time. There are 24 Video Terminals available for the users.

The next user task is Simultaneous Graphics System (SGS), also a Rand-developed system. It is a text editor with RJE facilities. Up to 10 users can be supported in a 60K region. Communications between SGS and the users is through the VOS system.

The third user task is CPS, the class A supported Conversational Programming System with Rand modifications. The PL/1 subset, CPS, is

MEMORY ALLOCATION

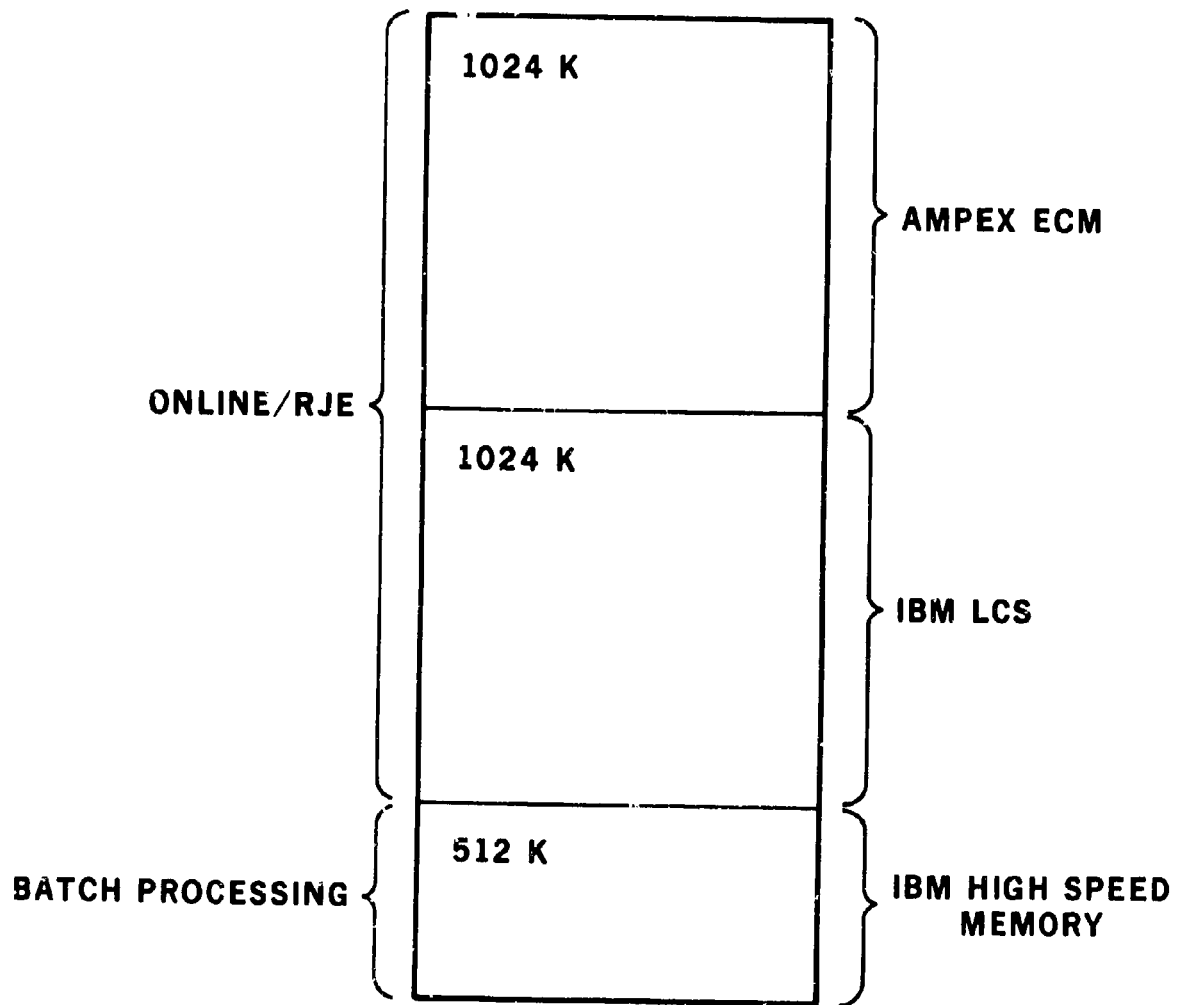


Fig. III-1--Memory Allocation

BATCH PROCESSING

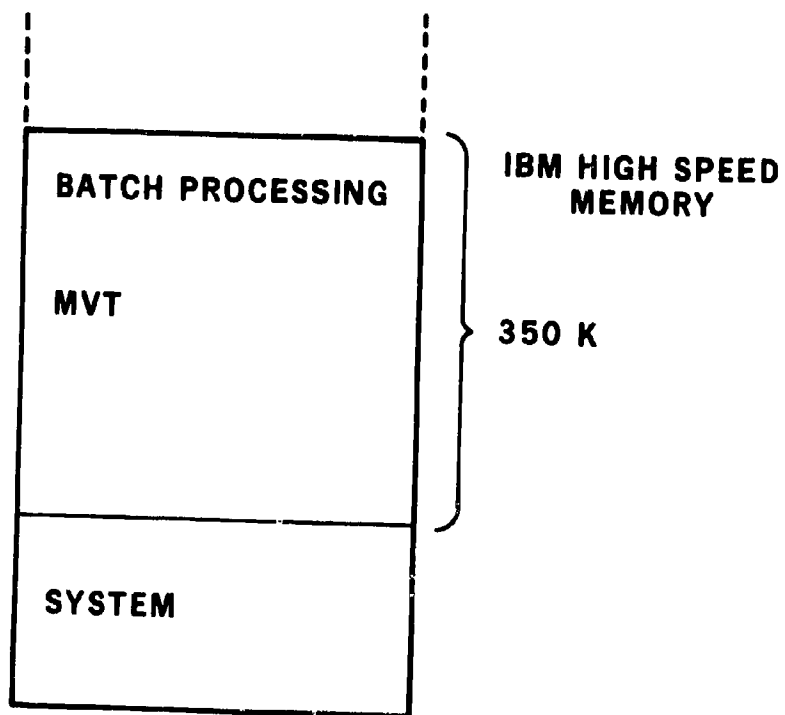


Fig. III-2--Batch Processing

ONLINE SYSTEMS

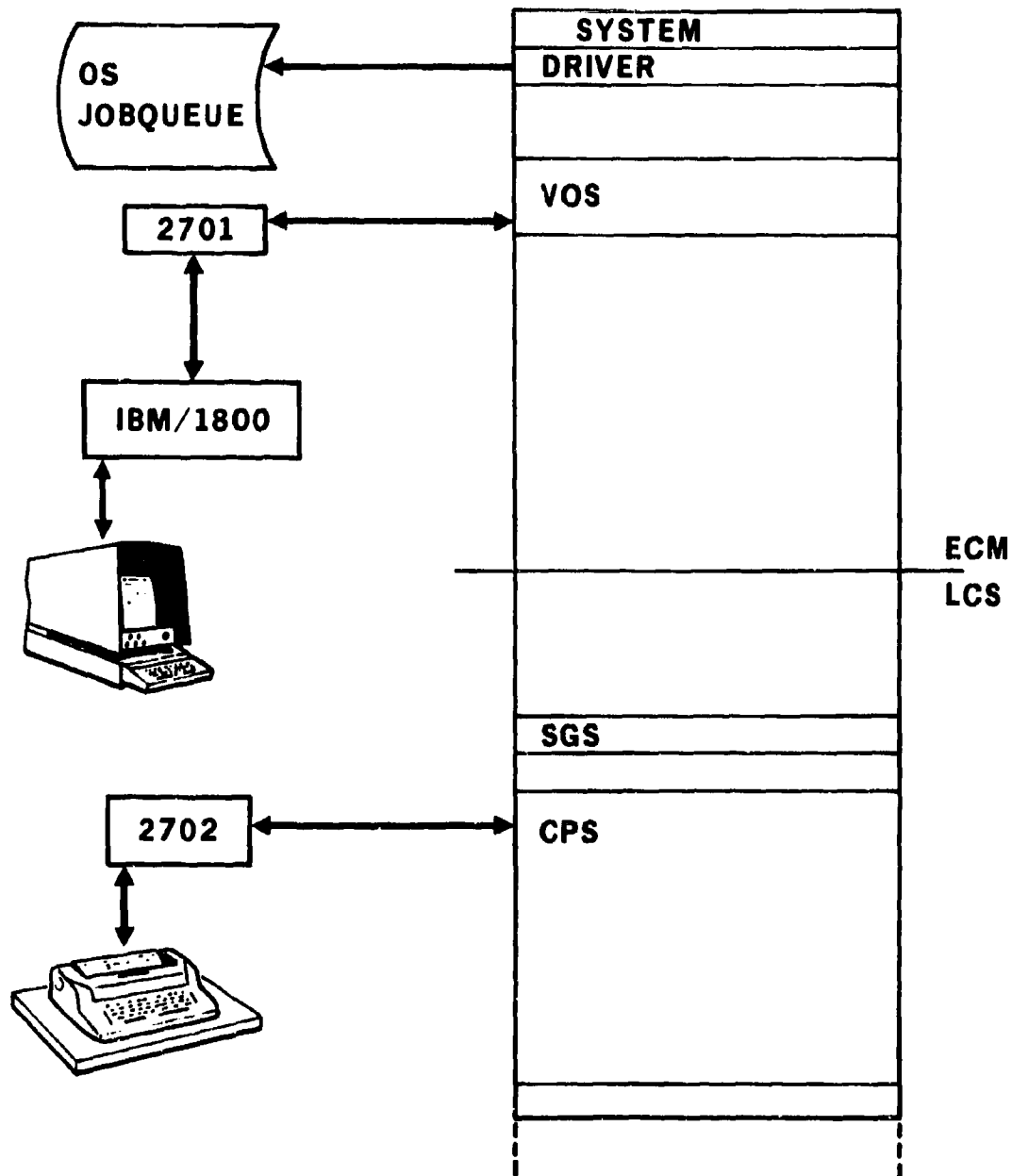


Fig. III-3--On-Line Systems

provided, as is the text editing with RJE facilities. Communication with CPS is through the 2701 to the 2741 typewriter terminals, or through the VOS to the Video Terminals. The Video Communication with CPS is one of the Rand modifications to CPS.

At the top of Fig. III-3 is a System task called DRIVER. DRIVER is a resident reader facility which sequentially processes RJE jobs passed from either SGS or CPS, and enters these jobs into the standard OS job queue. CPS or SGS can pass RJE jobs to DRIVER. Since DRIVER is a system task it must be active to reserve memory. The three user tasks, VOS, SGS, and CPS have Fixed Initiators which reserve the Storage for them when they are not resident.

RJE PROCESSING

In Fig. III-4, which describes RJE Processing, VOS is included because the video facilities are also available to the RJE processing regions. There are three 228K regions, three 104K regions, and one 64K region. The jobs executed in these regions are entered into the system via RJE through DRIVER. These jobs or tasks can include any processing that is normally done in the batch mode, with limitations on tape and disk mounts. That is, normal batch processing can be executed in these regions if the user is willing to pay for the slower-access memory. The purpose of these regions is to permit user-developed on-line systems nonfragmented storage for debugging and production processing. These regions are also used in another way. They can be reserved for projects which require rapid access to the system. An example of this is our Pseudo-50.

A bit of an explanation is necessary for the Pseudo-50. A 360/50 with 256K of Processor Storage, a 5-drive 2314, a printer, a card reader/punch, and a sign-in sheet sat in one corner of the Computer Room, where the users ran in a hands-on mode. These users also had access to the Video system; they IPLed the 360/50 and ran their jobs hands-on.

In order to save money the reader/punch and the printer were connected to the 360/65 and an OS Reader and Writer were provided for these devices, as indicated in Fig. III-5. A Fixed Initiator provided a

RJE PROCESSING

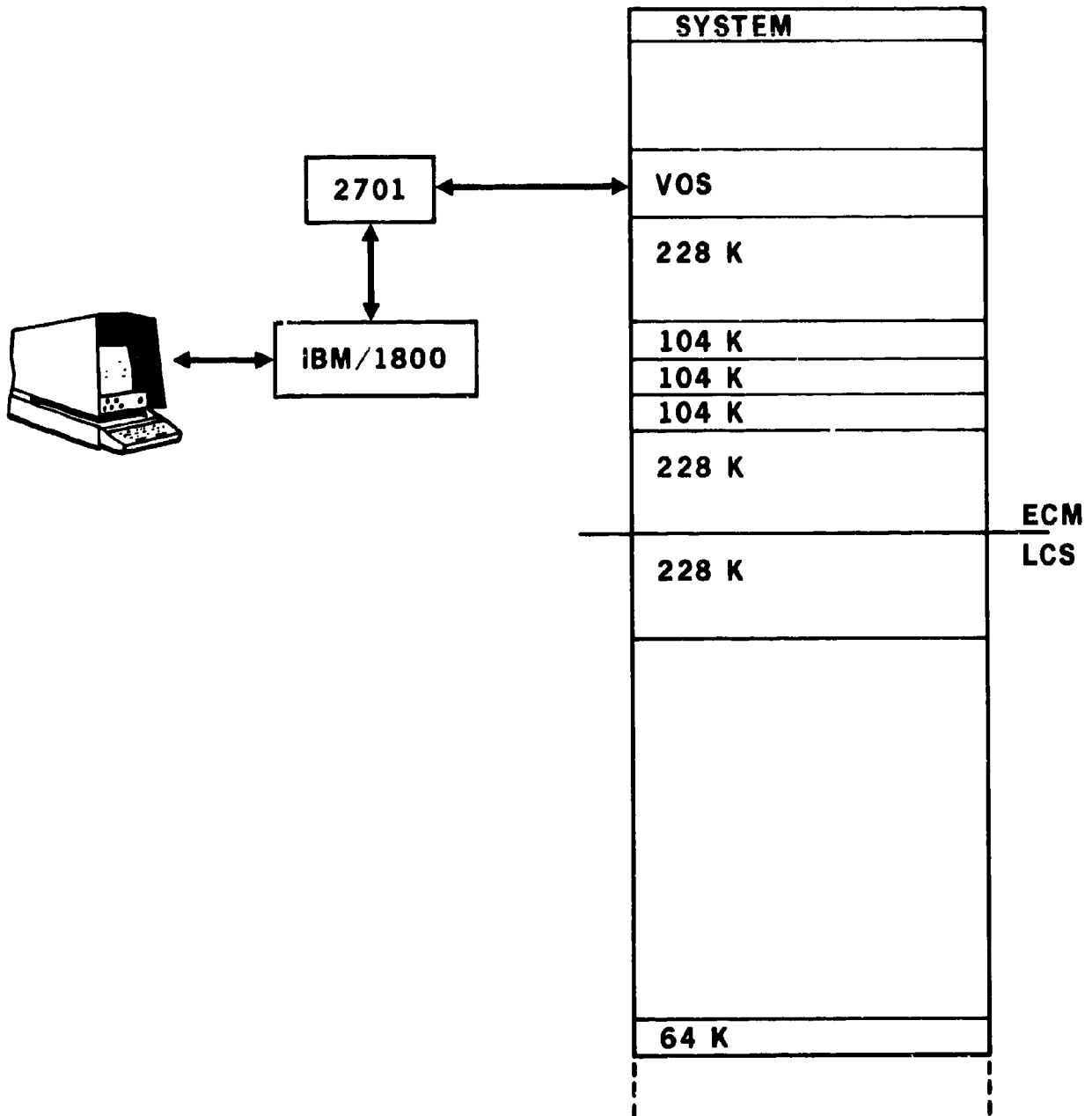


Fig. III-4--RJE Processing

PSEUDO-50

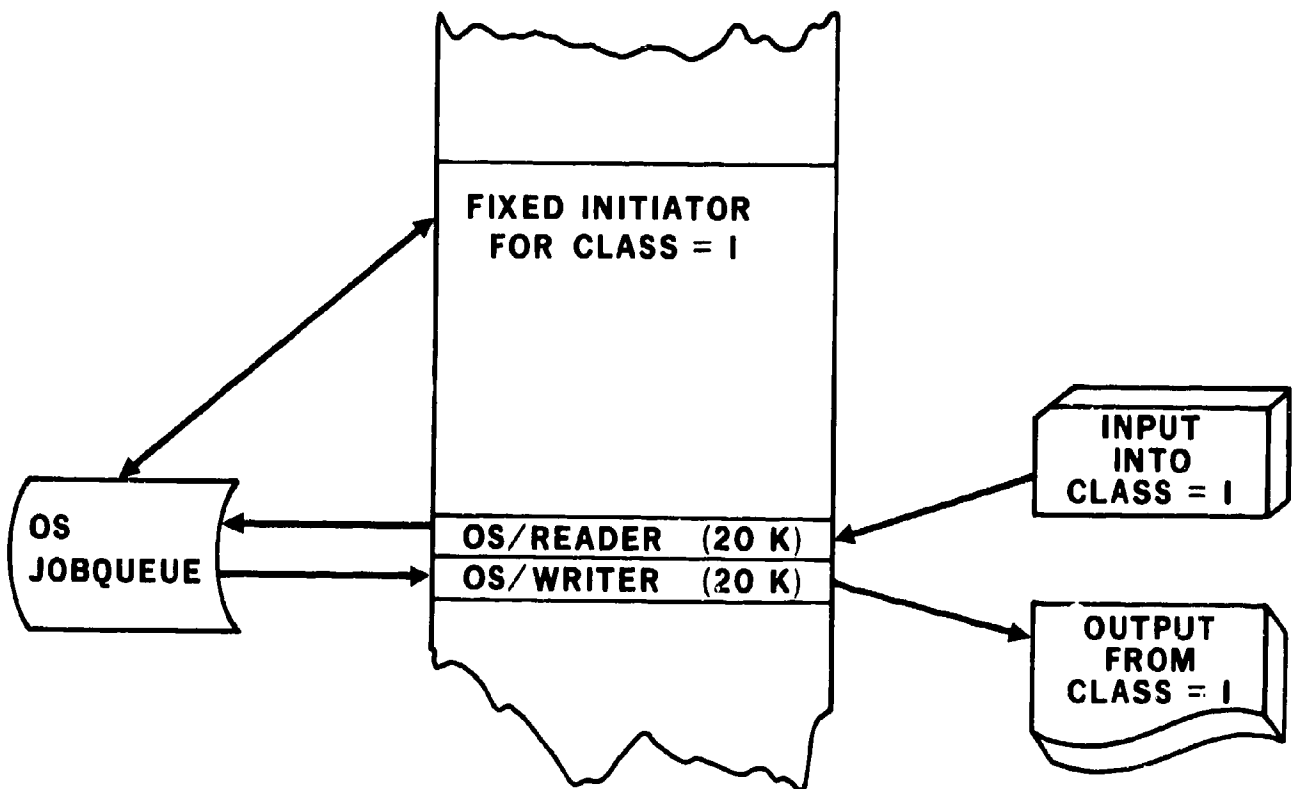


Fig. III-5--Pseudo-50

300K Fenced Region for our Pseudo-50. The sign-in log was maintained and three 2314 drives on the 360/65 were made available to these users. The 360/50 CPU, 256K Processor Storage, and the 5-drive 2314 were returned. The Pseudo-50 user could still enter his cards through the reader, sending the job into the standard OS job queue on the 360/65. The job would be picked up by the Fixed Initiator assigned to the appropriate job class, and when the job completed, the printer output was directed to the printer or card punch at the user location; hence the Pseudo-50. This is an example of how approximately \$20,000 per month of equipment rental was saved, and the user had effectively the same environment as the hands-on machine. Without Core Fencing through Fixed Initiators the idea of a Pseudo-50 could not have been accomplished.

Figure III-6 describes the On-line/RJE Processing System with its Fixed Region/Fenced Core. There is purposeful fragmentation; some storage is not fenced (i.e., between DRIVER and VOS). These unfenced regions are reserved for Operation Aids and the Batch Reader and Writers. Batch User jobs are not able to be initiated in these fragments because they are smaller in size than the minimum region request for a user task.

ONLINE/RJE SYSTEM

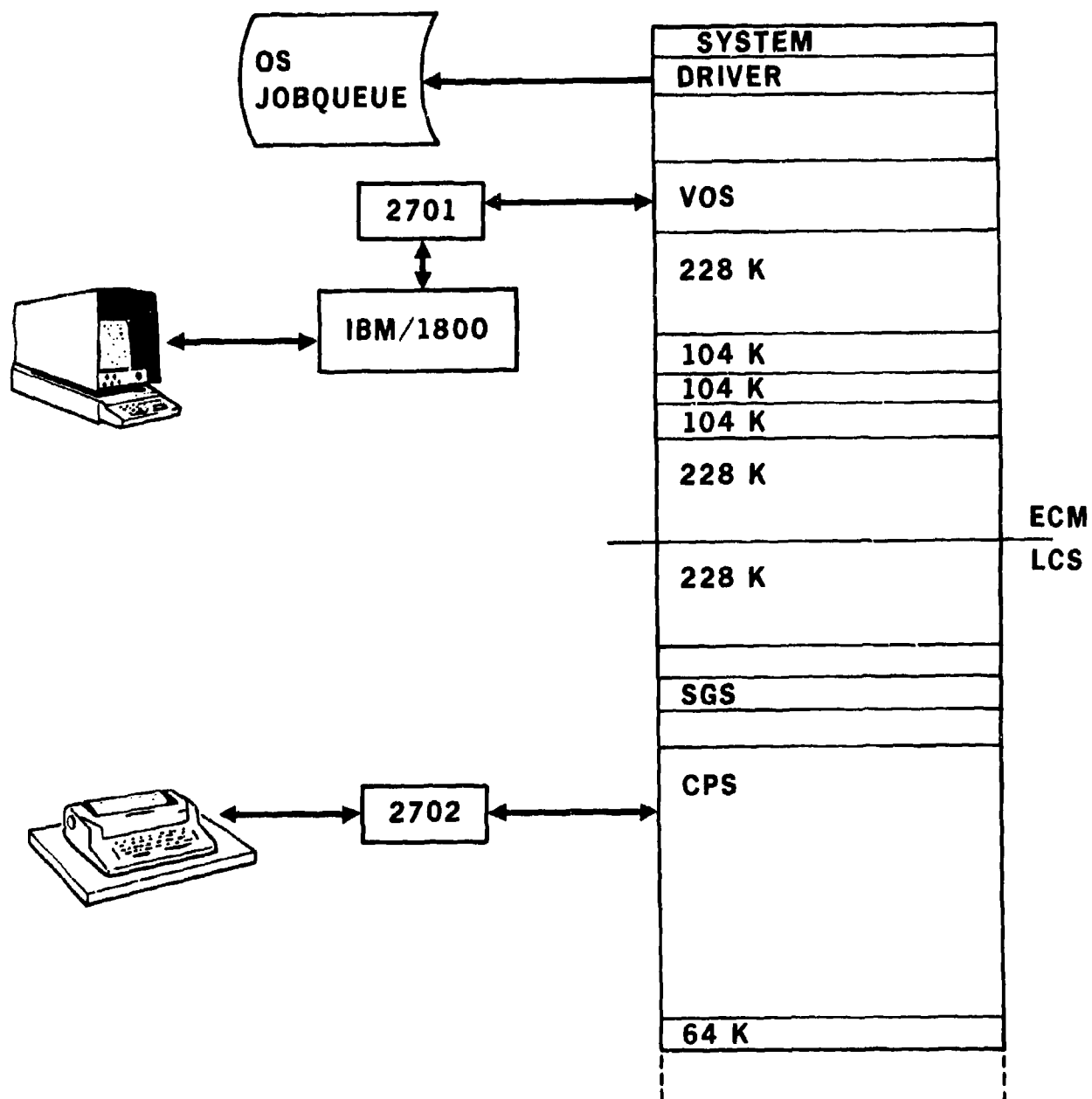


Fig. III-6--On-Line/RJE System

IV. IMPLEMENTATION OF THE FIXED INITIATOR

The facility for managing on-line storage in a multiprogramming environment had the requirement of easy implementation. This is expanded below in the Externals; the OS modifications necessary for implementation are expanded in the Internals.

EXTERNALS

The implementation of the Fixed Initiator modification is relatively straightforward. The Procedure names for the Initiator to be Fixed must start with a common alpha; we chose F for fixed. The region parameter on the execute statement for these initiators specify the fixed region required by the initiator. The set-up for the normal memory configuration takes about 20 minutes for a Warm Start. Fig. IV-1 illustrates the sequence of commands currently required to configure the system. These commands are executed via a reader procedure which points to the command data set as its input stream.

INTERNALS

The modifications for Fixed Initiators have changed since Release 17. The Initiator modules have been rewritten and Fixed Initiators now require modifications to four modules in LINKLIB. This represents less than 400 bytes of nonresident code in these four modules.

In what follows, attention is called to IBM's practice of identical naming of load modules and control sections (CSECTs). Beware of confusing the two. Microfiche names appear on the accompanying charts, as do load module names. Note that CSECT names are *not* represented on Figs. IV-2 or IV-3 (pp. 21-22).

I. Standard Initiation - MVT

The initiation task in an MVT environment has several steady-state conditions. Currently, at Rand, the conditions possible are:

- a. Waiting for work--in this case the initiation procedure is represented by one load module in the link pack area. None of available core storage is assigned to the initiation task.

FIGURE IV-1

STANDARD WARMSTART

<u>COREMAP IMAGE</u>		<u>"IPL" READER COMMANDS *</u>		
MASTER SCHEDULER	80A	H	Q	10
IPL (DRIVER)	22A	D	JOB NAMES, T	20
COREMAP	16A	S	COREMAP	30
LOVE	16A	S	LOVE	40
FVOS	122A	S	FVOS	50
FNIH	228A	S	FNIH	60
FHG104	104A	S	FHG104	70
F2HG104	104A	S	F2HG104	80
FGH104	104A	S	FGH104	90
FJ228	228A	S	FJ228	110
Ampex LCS/Boundary		S	WTRE	120
WTRE	20L	S	FF228	130
FF228	228L	S	WTRB	140
WTRB	20L	S	FSGS	150
FSGS	52L	S	WTRD	160
WTRD	20L	S	FCPS	170
FCPS	596L	S	TWRITER	185
TWRITER	24L	S	FM64	190
FM64	64L	S	ONLINE	200
LCS Boundary		S	NORMAL	230
FREE CORE	342H	S	SLOPPY	240
SQS & NUCLEUS	170H	S	SLOW	250
NORMAL	WAITING	S	SPEEDY	260
SLOPPY	WAITING	D	A	270
SLOW	WAITING			
SPEEDY	WAITING			

*This sequence of commands is activated by the operator issued command:

S IPL

- b. Job/step initiation processing--in this case the initiator is represented by several load modules. Some of these reside in, and are executed from, the link pack area. The others must be furnished with a region of available core storage. This allocation of core is carried out dynamically, under standard region management conventions. That is, the region is freed when the initiator modules relinquish control (e.g., to the task being initiated).
- c. Limbo--a name for the situation in which a user job step has been attached by the initiator. In this case the initiator is represented by a load module in the link pack area which will receive control when the job step terminates. The initiator proper has no dynamic storage assigned it at this time; but, the attached job step is running in a region obtained for it by the initiator prior to the attach.
- d. Job/step termination processing--this condition is similar to (b), above. When termination processing is complete the dynamically allocated core assigned to initiator load modules is released and the situation reverts to condition (a) or (b), above.

II. Fixed Initiation - MVT

An IBM Installation Newsletter, issued while Rand was on Release 17 of MVT, outlined what seemed a relatively easy way to obtain for MVT some of the benefits of the fixed partition system of MFT. Specifically, the modification involved changes to the steady-state conditions described above. Instead of releasing the region assigned to the initiator load modules when the initiator is waiting for work, or has attached a user job step, the region is kept fixed in core. Setting the region size requested in the initiator procedures to appropriate values enables the system to more efficiently manage our complex mixture of permanently resident on-line systems and transient batch job processing.

III. Implementation of Fixed Initiation

- a. For Release 17 of MVT--Implementation involved changes to only two control sections within one load module, IEFSD061. The single further requirement is that the procedure names for the fixed initiators begin with the letter F (e.g., FGH104, FF228). Changes to control sections are described as follows:
 - 1. CSECT IEFSD061: Code was inserted to test for a fixed initiator. If yes, and the initiator must wait for work, the wait is performed in-line *instead of* transferring control to load module IEFSD105 which normally frees the initiator's region, and then issues the wait from the link pack area.

2. CSECT IEFSD062: Code was inserted to test for a fixed initiator. If yes, and the job step being processed requires enqueueing on data sets, enqueueing is performed in-line. If yes, and no enqueueing is required, the in-line enqueue is bypassed. In either case, control then passes directly to the allocation interface module IEFSD062 *instead of* transferring control to module IEFSD102 which normally performs a region swap, in addition to checking for the need to enqueue on data sets.

With the exception of the above changes for initiators whose names begin with F, job step initiation proceeds in the standard manner. Figure III-2 (for Release 17) summarizes where the changes are located with respect to the entire procedure.

- b. For Release 19 of MVT--implementation was complicated by several rather basic changes to the job/step initiating procedure. In Release 19 some parts of the system task control routine were merged into the initiator. A more detailed explanation of these changes may be found in the MVT JOB MANAGEMENT PLM (GY28-6660). These modifications required a reworking of the system changes which would allow fixed initiation. The following text identifies, by CSECT within load module, the changes necessary to allow fixed initiation with Release 19.

1. Load Module IEFSD061

- o CSECT IEFSD062--this module retained the changes necessary under Release 17. Code was added to implement a "first time through" flag. This procedure allows a starting fixed initiator to obtain its core through a standard region swap in module IEFSD102. After this "first time through" the flag allows a fixed initiator no further access to module IEFSD102.
- o CSECT IEFSD061--this module retained the changes necessary under Release 17. Code was added to remotely reset the flag in CSECT IEFSD062 whenever load module IEFSD061 was loaded via an XCTL from modules IEFSD060 or IEFSD062.
- o CSECT IEFSD104--code was added to remotely reset the flag in CSECT IEFSD062 whenever load module IEFSD061 was loaded via an XCTL from module IEFSD263.
- o CSECT IEFW42SD--code was added to remotely reset the flag in CSECT IEFSD062 whenever load module IEFSD061 was loaded via an XCTL to entry points IEFW42SD or IEEV4221 within the block labeled "TERMINATION" (see Fig. IV-3).

- o CSECT IEFSD065--code was added to remotely reset the flag in CSECT IEFSD062 whenever load module IEFSD061 was loaded via an XCTL from module IEFSD062.
2. Load Module IEFSD062
- o CSECT IEFSD062--code was added to test for a fixed initiator. (As before, the fixed initiator's job-name must begin with F.) If yes, a branch is enabled which causes the region of core containing the Getpart Work Table (GWT) to be released. The absence of GWT is used in subsequent code as an indication that a region swap at attach-time is not desired.
3. Load Module IEFSD263
- o CSECT IEFSD263--code was added to test for a fixed initiator. If yes, code is bypassed which normally swaps regions just prior to job step attachment. Another fixed initiator test later in this CSECT prevents a region swap upon return to this CSECT when the attached job step gives up control.

Figure IV-3 (for Release 19) summarizes the above modifications, and locates them within the initiator control flow.

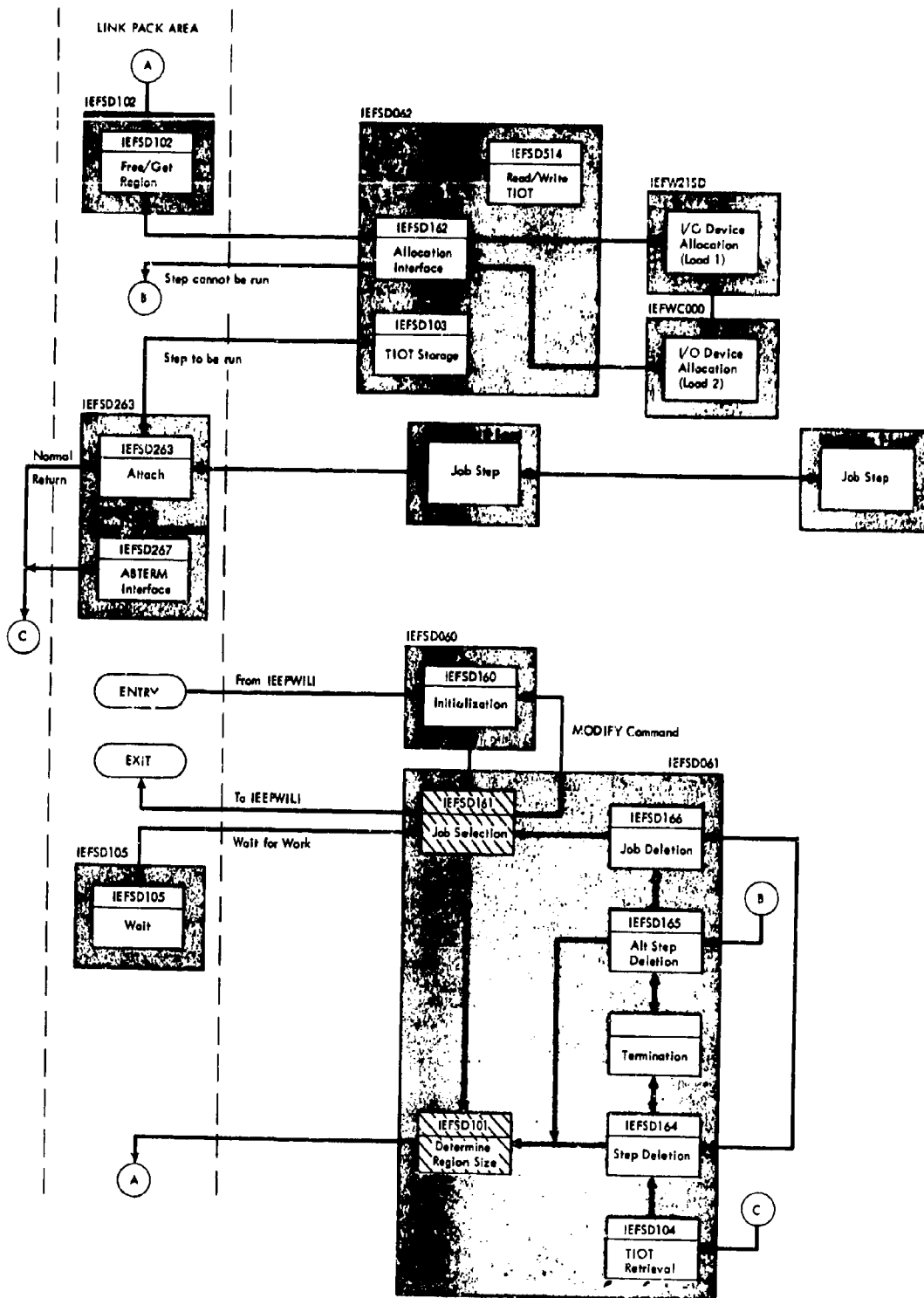


Fig. IV-2--Release 17

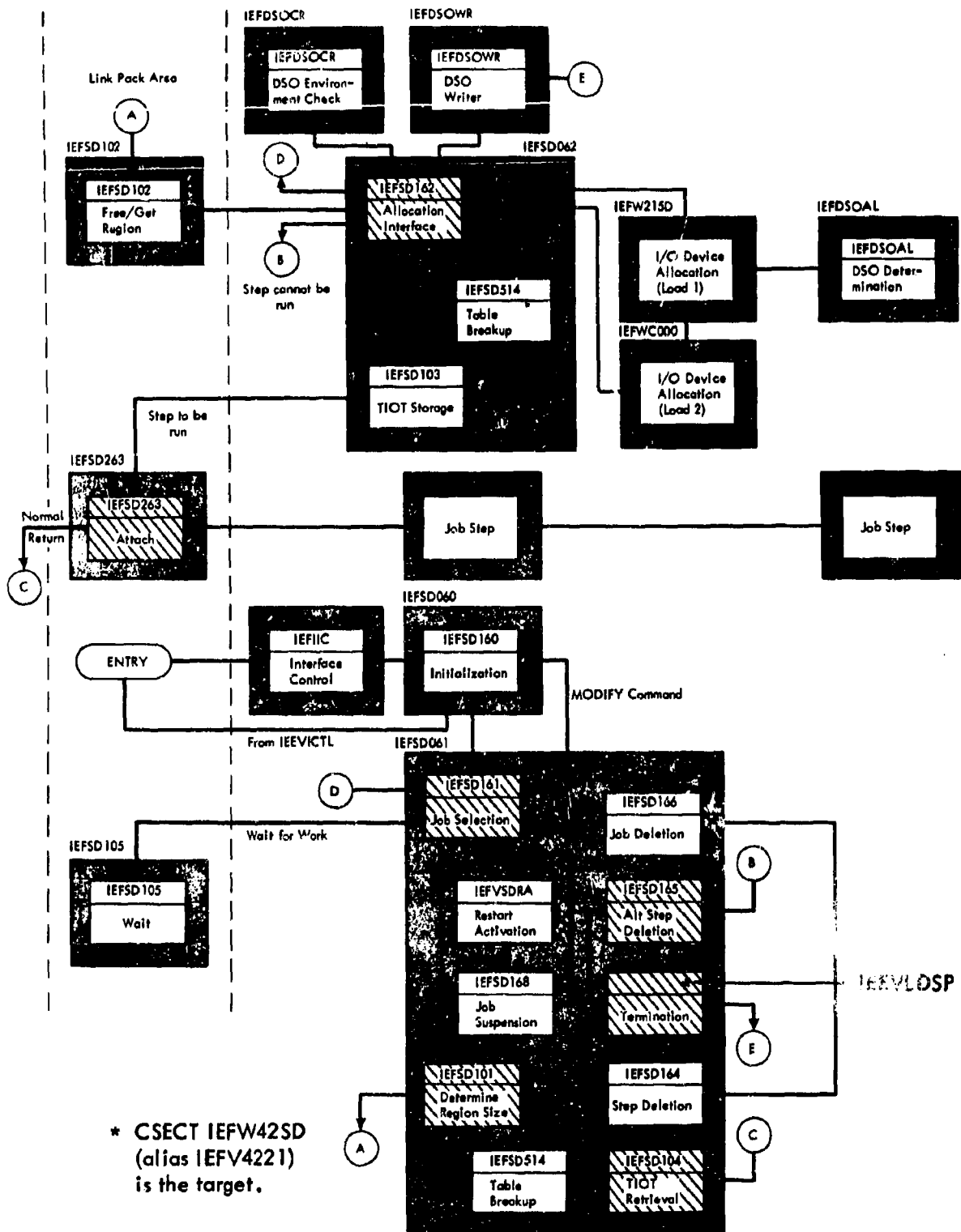


Fig. IV-3--Release 19

V. CONCLUSION

The Fixed Initiator facility is a stable and reliable modification to the MVT system at Rand which has accomplished our objectives and exceeded our expectations. Fixed Initiators can reserve storage for on-line systems, reserve storage for guaranteed access applications, and protect the batch processing from accidental execution in slower-speed storage. As a facility for memory management it has reduced the problems associated with controlling three access speeds of memory.

APPENDIX

The following listings reflect the control sections which have been modified in order to implement fixed initiation on OS/MVT Release 19.3. The patches are enclosed by rectangles.

```

TITLE 'IEFSD101-MS/1 PARTITION REPLACE INTERFACE'
IEFSD0062 CSECT
*
*2833 013800
*6092
*015361-015396,015700-016960,022200
      BALR R12,0
*
*****
*
*       IEFSD101-MS/1 PARTITION REPLACE INTERFACE
*
*****
* STATUS - CHANGE LEVEL 000
*
* FUNCTION TO DETERMIN THE PARTITION SIZES NEEDED FOR ALLOCATION 1272 00250019
*       FOR THE JOB STEP AND FOR TERRMINATION. 1272 00260019
*
* ENTRY POINT: IEFSD062 FROM IEFSD161 AND IEFSD164 AND IEFSD166
*
* INPUT: REGISTER ONE CONTAINS THE LCT ADDRESS
*
* OUTPUT: LCTPARM3 IN LCT CONTAINS SUBPOOL NO. AND PARTITION SIZE
*
* EXTERNAL REFERENCES - ABSOLUTE LOCATION 16 IS USED TO PICK UP THE
*       ADDRESS OF THE CVT.
*
* EXIT: IEFSD102 - PARTITION REPLACE MODULE
*
* TABLES/WORKAREAS - LCT, SCT, MASTER SCHEDULER RESIDENT CORE AREA
*
* ATTRIBUTES - REENTRANT
*****
*
      USING *,R12
      USING IEFLNT,R10
      USING IEFSC7,R9
      EJECT
*****
*
* THE FOLLOWING SECTION OF CODE SERVES ONLY TO IDENTIFY THIS MODULE
* IN A MEMORY DUMP
*
      B FIRST
      DC X'CCCC'
      DC C'IEFSD101'
      DC X'06081966'
      DC X'CCCC'
FIRST DS OH
*
*****
      EJECT
      LR R10,R1          LCT ADDRESS
      LR R4,R13
      GETMAIN R,LV=72
SMF1 01002018
SMF1 01004018

```

IR	R13,R1		SMF1	01006018
USING	CVTDSFCT,R1	CVT ADDRESSABILITY	M3849	01006219
L	R1,CVTPTR	ADDRESS OF CVT PTR	M3849	01006419
L	R1,CVTTCHP	GET TCH PTRS	M3849	01006619
DRDP	R1		M3849	01006819
L	R1,D4(R1)	GET OUR TCH FOR SMF & ALLUC	M3849	01007019
ST	R1,LCTTCHAD	PUT OUR TCH ADDR IN LCT	M3849	01007219
LR	R1,R10		SMF1	01008018
L	R15,VCONSMF	GO TO SMF INIT EXIT	SMF1	01010018
HALR	R14,R15	PASS LCT IN R1	SMF1	01012018
FRFEMA	IN R, LV=72, A=(R13)		SMF1	01014018
LR	R13,R4		SMF1	01016018
L	R2,OMCAP	LOAD PTR TO TRACK STACK INFO		01020000
L	R2,0(R2)	LOAD PTR TO STACK		01040000
LA	R2,0(R2)	GET RID OF UPPER BYTE		01060000
LTR	R2,R2	IF ADDR IS ZERO, NO STACK- SO		01080000
HC	R,SD101005	BRANCH AROUND		01100000
LR	R5,R13	ADDR IS NOT ZERO - SAVE REG		01120000
LR	R13,R2	POINT TO SAVE AREA IN STACK		01140000
LA	R1,OMGR1	LOAD PTR TO OMPA		01160000
XC	OMPOP(1),OMPOP	CLEAR OP CODE FOR PURGE RTN		01180000
		NOT TO CONFUSE IT WITH DELETE		01200000
L	R15,IFFSDSTP	ADDR OF STACK PURGE PGM		01220000
HALR	R14,R15	GO TO PURGE AND FREE THE STACK		01240000
LR	R7,R15	SAVE THE RETURN CODE		01260000
LR	R1,R13	LOAD PTR OF RETURNED SAVE AREA		01280000
FRFEMA	IN R, LV=72, A=(1)	AND FREE IT		01300000
LR	R13,R5	RESTORE REGISTER		01320000
LTR	R7,R7	TEST RETURN CODE -- IF IT 0,		01340000
HC	R,SD101005	RETURN WAS NORMAL, BRANCH		01360000
LA	R1,X'0000'	I/O ERROR IN TRACK	20768	01370018
SLA	R1,12	STACKING	20768	01380018
ABEND	(R1),DUMP	OKO ABEND	20768	01390018
SD101005	EQU *			01400000
XC	LCTINTSW(D1),LCTINTSW	ZERO INTERNAL SWITCHES	1272	01410019
L	R9,LCTSCCTAD	GET SCT ADDRESS		01420000
L	R4,16	LOAD CVT ADDR		01460000
USING	CVTDSFCT,R4			01480000
L	R5,CVTMSER	LOAD MASTER RES CORE ADDR		01500000
USING	HASF,R5			01520000
LH	R7,RAMINPAR	LOAD MINIMUM PARTITION SIZE		01520717
DRDP	R4			01521417
L	R4,LCTJCTAD	LOAD JCT ADDRESS		01522117
USING	JCT,R4			01522817
TM	JCTJSTAT,JCTABEND	HAS ABEND OCCURED WHICH MIGHT		01523517
		ALLOW A FOLLOWING STEP TO		01524217
		EXECUTE		01524917
RZ	SD101006	NO, GO TEST JOB FAIL AND JOB		01525617
		FLUSH BITS		01526317
TM	SCTABEND,SCTEVEN+SCTONLY			01527017
		YES, IS THIS STEP TO RUN AFTER		01527717
		THE ABEND		01528417
RZ	SD101007	NO, GO SET UP FOR MINPART		01529117
CLI	SCTABEND+2,0	WERE EIGHT CONDITIONS PRESENT		01529817
		FOR THIS STEP		01530517
RNE	SD101007	YES, NOT EVEN/ONLY, GO SET UP		01531217
		TO GET MINPART		01531917

-27-

	R	SD101008	THIS STEP WILL BE EXECUTED- GO	01532617
*			GET REQUESTED REGION	01533317
SD101006	EQU	*		01534017
	TM	JCTJSTAT,X'44'	IS JOB FAIL OR FLUSH HIT ON	CR17 01534717
	HZ	SD101008	NO, GO GET REQUESTED REGION	01535417
SD101007	DI	LCTINTSW,LCTMINRG	JOB FAIL INDICATE MINPAR	1272 01536419
	XC	SCTMSADR(D8),SCTMSADR	ZERO ADDRESSES	1272 01537419
SD101008	EQU	*		01540317
*			THIS ROUTINE BUILDS THE GETPART WORK TABLE. THIS TABLE IS TO BE	LS17 01543017
*			USED AS A PARAMETER LIST FOR LIST FORM OF GETMAIN TO OBTAIN A	LS17 01546017
*			REGION	LS17 01549017
	LA	R0,GPMINPAR+4-GETPTWT	LOAD LENGTH OF GWT	LS17 01552017
	LA	R8,TCORF	GET CORE FROM SUPERVISOR CORE	LS17 01555017
	SLL	R8,24		LS17 01558017
	OR	R0,R8	PLACE SUBPOOL + LENGTH IN R0	LS17 01561017
	GETMAIN	R, LV=(0)	GET CORE FOR GETPART WK TALE	LS17 01564017
	LR	R8,R1		LS17 01567017
	USING	GETPTWT,R8		LS17 01570017
	MVC	GPADDOH(8),SCTMSADR	MOVE HIERARCHY ADDRESSES INTO	CR17 01573017
*			WORK TABLE	CR17 01576017
	SLL	R7,M10	MULTIPLY BY 1024	1272 01577019
	MVI	GPCODE,UNCONDGM		1272 01578019
	MVI	GPSURP,POOL247	INDICATE SUBPOOL 247	1272 01579019
	ST	R7,GPMINPAR	SET UP FOR REGISTER FORM	1272 01580019
	MVI	GPMINPAR,POOL247	OF GETPART IF ERROR	1272 01581019
	MVC	GPADDOH(D4),GPADDOH	SAVE ORIGINAL ADDRESS	1272 01582019
	LH	R11,SCTMSSZE	H0 SIZE FROM SCT	1272 01583019
	SLL	R11,M10	MULTIPLY BY 1024	1272 01584019
	ST	R11,GPSIZEH0	SIZE INTO GWT	1272 01585019
	LH	R3,SCTLCSSZ	H1 SIZE FROM SCT	1272 01586019
	SLL	R3,M10	MULTIPLY BY 1024	1272 01587019
	ST	R3,GPSIZEH1	SIZE INTO GWT	1272 01588019
	MVI	GPSIZEH1,ENDLIST	INDICATE END OF LIST	1272 01589019
	TM	LCTOPSW2,LCTINTH0	TEST FOR HIER ZERO	1272 01590019
	R0	SD10110	BRANCH IF HIER ZERO	1272 01591019
	TM	LCTOPSW2,LCTINTH1	TEST FOR HIER ONE	1272 01592019
	HZ	SD10109	BRANCH NOT HIER ONE	1272 01593019
	DI	LCTINTSW,LCTIHIER	INIT RUN IN HIER ONE	1272 01594019
	AR	R3,R11	ADD REGION REQUESTS	1272 01595019
	ST	R3,GPSIZEH1	STORE IN H1 SIZE	1272 01596019
	MVI	GPSIZEH1,ENDLIST	INDICATE END OF LIST	1272 01597019
	XC	GPSIZEH0(D4),GPSIZEH0	ZERO H0 SIZE	1272 01598019
	B	SD10113		1272 01599019
SD10109	LTR	R11,R11	TEST FOR H0 SIZE	1272 01600019
	BNE	SD101105		1272 01601019
	LTR	R3,R3	TEST FOR HIER ONE	1272 01602019
	BE	SD101105	BRANCH DEFAULT TO ZERO	1272 01603019
	DI	LCTINTSW,LCTIHIER	INIT RUN IN HIER ONE	1272 01604019
	B	SD10113		1272 01605019
SD10110	AR	R11,R3	ADD REGION REQUESTS	1272 01606019
	ST	R11,GPSIZEH0	STORE IN H0 SIZE	1272 01607019
	XC	GPSIZEH1(D4),GPSIZEH1	ZERO H1 SIZE	1272 01608019
	SR	R3,R3	ZERO H1 SIZE IN REGISTER	1272 01609019
SD101105	LA	R14,GPADDOH	SET UP PTR TO ADDRESS	1272 01610019
	ST	R14,GPADDOH	LIST OF HIERARCHIES	1272 01611019
	LA	R14,GPSIZEH0	SET UP PTR TO REGION	1272 01612019
	ST	R14,GPSIZEA	SIZE LIST OF HIERARCHIES	1272 01613019

	LTR	R3,R3	TEST FOR HIER ONE	1272	01614019
	RNE	SD10114	BRANCH IF THERE IS H1	1272	01615019
	MVI	GPSIZEH0,ENDLIST	SET H0 SIZE AS END OF LIST	1272	01616019
	R	SD10114		1272	01617019
SD10113	LA	R14,GPADDH1	SET UP PTR TO ADDRESS LIST	1272	01618019
	ST	R14,GPADDLT	FOR HIERARCHY ONE	1272	01619019
	LA	R14,GPsizeH1	SET UP PTR TO REGION SIZE	1272	01620019
	ST	R14,GPsizeA	STORE PTR	1272	01621019
	CLR	R3,R7	COMPARE REQUEST WITH MINPAR	1272	01622019
	HNL	SD10124	BRANCH REQUEST GREATER	1272	01623019
	MVC	GPSIZEH1+D1(D3),GPMINPAR+D1	MOVE IN MINPAR SIZE	1272	01624019
	LR	R6,R3	REQUEST SIZE IN R6	1272	01625019
	R	SD10115		1272	01626019
SD10114	CLR	R11,R7	COMPARE REQUEST WITH MINPAR	1272	01627019
	HNL	SD10124	BRANCH REQUEST GREATER	1272	01628019
	MVC	GPSIZEH0+D1(D3),GPMINPAR+D1	MOVE IN MINPAR SIZE	1272	01629019
	LR	R6,R11	REQUEST IN R6	1272	01630019
SD10115	TM	LCTINTSW,LCTMINRG	TEST FOR JOB FAIL	1272	01631019
	RZ	SD10116	BRANCH JOB NOT FAILED	1272	01632019
SD101155	TM	LCTINTSW,LCTIHIER	TEST FOR HIERARCHY	1272	01633019
	RZ	SD101165	BRANCH HIERARCHY ZERO	1272	01634019
	NC	GPSIZEH0(D4),GPSIZEH0	ZERO HIERARCHY ZERO SIZE	1272	01635019
	MVC	GPSIZEH1+D1(D3),GPMINPAR+D1	MOVE MINPAR SIZE	1272	01636019
	R	SD10125		1272	01637019
SD101165	XC	GPSIZEH1(D4),GPSIZEH1	ZERO H1 SIZE	1272	01638019
	MVI	GPSIZEH0,ENDLIST	INDICATE END OF LIST	1272	01639019
	MVC	GPSIZEH0+D1(D3),GPMINPAR+D1	MOVE IN MINPAR SIZE	1272	01640019
	R	SD10125		1272	01641019
SD10116	OC	HAMIPAR2(D2),HAMIPAR2	IS TERM IN LPA	1272	01642019
	BE	SD10120	BRANCH NOT IN LPA	1272	01643019
	TM	LCTINTSW,LCTIHIER	INIT IN HIERARCHY ONE	1272	01644019
	RO	SD101166	BRANCH HIERARCHY ONE	1272	01645019
	OC	GPADDH0+D1(D3),GPADDH0+D1	SPECIFIC REQUEST	1272	01646019
	BE	SD10117	BRANCH NOT SPECIFIC	1272	01647019
	SR	R7,R11	R7=(MINPAR-S)	1272	01648019
	L	R2,GPADDH0	GET REQUESTED ADDRESS	1272	01649019
	SR	R2,R7	R2=A-(MINPAR-S)	1272	01650019
	ST	R2,GPADDH0	NEW ADDRESS IN PARM	1272	01651019
	R	SD10117		1272	01652019
SD101166	OC	GPADDH1+D1(D3),GPADDH1+D1	SPECIFIC REQUEST	1272	01653019
	BE	SD10117	BRANCH NOT SPECIFIC	1272	01654019
	SR	R7,R3	R7=(MINPAR-S)	1272	01655019
	L	R2,GPADDH1	GET REQUESTED ADDRESS	1272	01656019
	LA	R2,0(R2)	CLEAR HIGH ORDER BYTE	1272	01657019
	SR	R2,R7	R2=(A-(MINPAR-S)	1272	01658019
	ST	R2,GPADDH1	NEW ADDRESS IN PARM	1272	01659019
	MVI	GPADDH1,HEX01	INDICATE HIERARCHY ONE	1272	01660019
SD10117	LH	R7,HAMIPAR2	LOAD MIN-MIN SIZE	1272	01660519
	SLL	R7,M10	MULTIPLY BY 1024	1272	01661019
	CLR	R6,R7	IS REQUEST GREATER THEN MIN-MIN	1272	01661519
	RNH	SD10118	BRANCH MIN-MINPAR GREATER	1272	01662019
	ST	R6,GPsizePP	USE REQUEST FOR P/P	1272	01663019
	R	SD10126		1272	01664019
SD10118	EQU	*		1272	01665019
	TM	LCTOPSW1,LCTMINPF	SMALL REGION FLAG	1272	01667019
	RO	SD10119	BRANCH NO SMALL REGION	1272	01668019
	ST	R7,GPsizePP	USE MIN-MINPAR FOR P/P	1272	01669019

-29-

SD10119	R	SD10126		1272	01670019
	ST	R6,GPSIZEPP	USE REQUEST FOR P/P	1272	01671019
	ST	R7,GPSIZET	USE MIN-MINPAR FOR TERM	1272	01672019
	B	SD10127		1272	01673019
SD10120	TM	LCTOPSW1,LCTMINPF	SMALL PARTITION OPTION	1272	01674019
	AZ	SD10125	BRANCH NO	1272	01675019
	ST	R6,GPSIZEPP	USE REQUEST FOR P/P	1272	01676019
	ST	R7,GPSIZET	USE MIN/MINPAR FOR TERM	1272	01677019
	R	SD10127		1272	01678019
SD10124	TM	LCTINTSW,LCTMINRG	TEST FOR JOB FAIL	1272	01679019
	RO	SD101155	BRANCH IF JOB FAILED	1272	01680019
SD10125	XC	GPSIZEPP(D4),GPSIZEPP		1272	01681019
SD10126	XC	GPSIZET(D4),GPSIZET	ZERO TERMINATE SIZE	1272	01682019
SD10127	EQU	*		1272	01683019
SD101C4	LA	R8,0(R8)	ZERO HIGH ORDER BYTE	LS17	01699017
	ST	R8,LCTPARM3	GETMAIN PARAM FOR IEFS0102	LS17	01702017
	L	R1,LCTQDRTY	ADDRESS OF CSCH	M4176	01702619
	USING	IEFCSCB,R1		M4176	01703219
	TM	CHSWT,CHSYS	IS THIS A SYSTEM TASK	M4176	01703819
	RO	SKIPCHAP	BRANCH SYSTEM TASK - SKIP CHAP	M4176	01704419
	DROP	R1		M4176	01705019
	SR	R1,R1	CLEAR REGISTER	1241	01706018
	LA	RO,15	PUT 15 IN REG	1241	01710018
	IC	R1,LCTQDRTY	LOAD JOB SCHEDULING PRIORITY	1241	01714018
	SR	RO,R1	GET 15-PRTY	1241	01718018
	LNR	RO,RO	NEGATE REGISTER	1241	01722018
	SR	R1,R1	CLEAR REGISTER	1241	01726018
	CHAP	(0),(1)	CHAP DOWN BY 15-PRTY	1241	01730018
SKIPCHAP	EQU	*		M4176	01735019
	LR	R1,R10	RESTORE LCT ADDRESS		01740000
	L	15,16	MOTHER CVT		0120
	L	15,0(0,15)	DOUBLEWORD		0121
	L	15,4(0,15)	THE TCB		0122
	L	15,12(0,15)	OUT TIOT		0123
	CLI	0(15),C'F'	FIXED PARTITION?		0124
	BNE	NORMAL	NO - PROCEED NORMALLY		0125
	CLI	FTTF,X'FF'	CHECK FIRST TIME THRU FLAG.		
	BE	HITIT	ALLOW FIXED INITIATION.		
	MVI	FTTF,X'FF'	SET FTTF TO ALLOW FI NEXT TIME.		
	B	NORMAL	LET IT GO THIS TIME.		
FTTF	DC	X'00'	FIRST TIME THRU FLAG.		
HITIT	L	R1,LCTPARM1	GET DS END TABLE.		
	LTR	R1,R1	IS IT ZERO?		0127
	BZ	LACTATE	YEAH		0128
	LA	R1,8(0,R1)	END ON THE		0129
	SVC	56	JOB'S DATASETS		0130
	L	RO,LCTPARM2	FREE UP DS END		0131
	L	R1,LCTPARM1	TABLE CORF		0132
	SVC	10			0133
LACTATE	LR	R1,R10	RESTORE LCT POINTER		0134
	XCTL	EP=IEFS0062	GO TO MEMBER IEFS0062		0135
NDRMAL	DS	OH			0136
	XCTL	EP=IEFS0102,MF=(E,(1))			01760000
IEFS0STP	DC	V(IEFS0112)	ADDR OF TRACK STACK PURGE PGM		01780000
VCONSMF	DC	V(IEFSMFIE)	*	SMF1	01790018
RO	EQU	0			01800000
R1	EQU	1			01820000

R2	EQU	2		01840000
R3	EQU	3		01860000
R4	EQU	4		01880000
R5	EQU	5		01900000
R6	EQU	6		01920000
R7	EQU	7		01940000
R8	EQU	8		01960000
R9	EQU	9		01980000
R10	EQU	10		02000000
R11	EQU	11		02020000
R12	EQU	12		02040000
R13	EQU	13		02060000
R14	EQU	14		02080000
R15	EQU	15		02100000
TCORE	EQU	253		LS17 02110017
CVTDSECT	DSECT			02120000
UNCONDGM	EQU	X'80'		1272 02121019
M10	EQU	10		1272 02122019
ENDLIST	EQU	X'80'		1272 02123019
D1	EQU	1		1272 02124019
D2	EQU	2		1272 02125019
D3	EQU	3		1272 02126019
D4	EQU	4		1272 02127019
DA	EQU	8		1272 02128019
POOL247	EQU	247		1272 02129019
HEX01	EQU	X'01'		1272 02130019
	CVT			02140000
IEERASEA				
IEFLOT	DSECT			02180000
IEFALLCT				02200000
	DS	3F		1272 02204019
LCTINTSW	DS	0C	INITIATOR INTERNAL SWITCHES	1272 02208019
LCTIHIER	EQU	128		1272 02212019
LCTSTOP	EQU	4	INITIATOR INTERNAL STOP	1272 02216019
LCTABEND	EQU	2	P/P ABHENDED	1272 02220019
LCTMINRG	EQU	16		1272 02224019
LCTSPLP	DS	F		02228019
	DS	42F		1272 02232019
QMGR1	DS	9F	MAIN QMGR PARAMETER AREA	02240000
QMGR2	DS	9F	'ALTERNATE' QMGR PARAMETER AREA	02260000
TRSTKINF	DS	2F	NEEDED FOR TRACK STACKING AND QUEUE	02280000
*			BREAK INFORMATION. 1ST BYTE TO CON-	02300000
*			TAIN NUMBER OF BUFFERS, NEXT 3 BYTES	02320000
*			FOR STACK ADDR, NEXT 4 BYTES FOR	02340000
*			QUEUE BREAK INFORMATION	02360000
QMCAP	EQU	QMGR1+24	PTR IN QMPA TO TRK STACK INFO	02380000
QMPDP	EQU	QMGR1+8	QMGR OP CODE IN QMPA AREA	02400000
	DS	6F		1272 02400219
* THESE FIELDS ARE NEEDED FOR L-SHAPE/INIT MERGE			*	1272 02400419
LCTOPSW1	DS	0C	INITIATOR OPTIONS BYTE 1	1272 02400619
LCTPKYF	EQU	128	DONT GET PROTECT KEY	1272 02400819
LCTWFF	EQU	64	DONT PROCESS DEDICATED WORKF	1272 02401019
LCTSTMDP	EQU	32	DONT PROCESS STOP/MODIFY	1272 02401219
LCTMINPF	EQU	16	GET REGION SIZE SPECIFIED	1272 02401419
LCTCANF	EQU	8	ALLOW CANCEL ONLY AT ALLOC	1272 02401619
LCTONEJF	EQU	4	PROCFSS ONLY ONE JOB	1272 02401819
LCTICMDF	EQU	2	DONT PROCESS INITIATOR CMDS	1272 02402019

-31-

LCTEXIT	DS	F	ADDRESS INITIATOR EXIT LIST	1272	02402219
LCTDPSW2	DS	0C	INITIATOR OPTIONS BYTE 2	1272	02402419
LCTTIMEF	EQU	128	DONT TIME THIS JOB	1272	02402619
LCTGRF	EQU	64	DONT ALLOW CHECKPT/RESTART	1272	02402819
LCTDSOF	EQU	32	DONT PROCESS DSO	1272	02403019
LCTINTHO	EQU	16	H0 SPECIFIED ON START COMMAND	1272	02403219
LCTINTH1	EQU	8	H1 SPECIFIED ON START COMMAND	1272	02403419
LCTCONM	DS	F	COMMUNICATIONS PARAM AREA PTR	1272	02403619
					02404017
JCT	DSFCT				02408017
	IEFAJCTR				02412017
	EJECT				02413019
IEFCSCR	DSECT				02414019
	IEECHAIN				02415019
	EJECT				02416017
IEFSCT	DSECT				02420000
	IEFASCTH				02440000
GETPTWT	DSECT				02441017
GPSIZEA	DS	1F	CONTAIN THE ADDRESS OF THE REGION SIZE LI	LS17	02442017
GPADDLT	DS	1F	CONTAIN THE ADDR OF ADDR LIST FOR THE HIE	LS17	02443017
GPCODE	DS	X'01	CONTAIN 80 TO INDICATE UNCONDITIONAL REGU	LS17	02444017
GPSURP	DS	X'01	CONTAIN SUBPOOL 247 OR 246	LS17	02445017
	DS	H'01		LS17	02446017
GPHIARCO	DS	0F	CONTAINS 00	LS17	02447017
GPADDHO	DS	1F	CONTAINS ADDRESS(SPECIFIC) ,ZERO(NON-SPEC	LS17	02448017
GPHIARC1	DS	0F	CONTAINS 01	LS17	02449017
GPADDH1	DS	1F	CONTAINS ADDRESS(SPECIFIC) ,ZERO(NON-SPEC	LS17	02450017
GPSIZEH0	DS	1F	SIZE OF H0	LS17	02451017
GPSIZEH1	DS	1F	SIZE OF H1	LS17	02452017
GPSIZEPP	DS	1F	REGION SIZE FOR P/P	1272	02452319
GPSIZET	DS	1F	REGION SIZE FOR TERMINATE	1272	02452619
GPADDHOR	DS	1F	ORIGINAL ADDRESS	CR17	02453017
GPMINPAR	DS	1F	247 + MINIMUM REGION SIZE	CR17	02454017
	END				02460000

TITLE 'IEFSD161 VMS JOB SELECT MODULE'	00020000
IEFSD061 CSECT	00040000
*	A21364 00045019
*037880	M0429 00050018
*	00051019
* RELEASE 19 - - -	00052019
* 037700-037970,124053-124062	A27668 00052519
* 033800-034400,059400-061000,089600-091000,107800-108400	099 00053019
* 124000-127000	099 00054019
*	00055019
*	00056019
*	00057019
*	00058019
*	00059019
*****	00060000
*	* 00080000
JOB SELECT MODULE	* 00100000
*	* 00120000
*****	* 00140000
*	* 00160000
*	00180000
*STATUS: CHANGE LEVEL 000	00200000
* FUNCTION: TO GET THE HIGHEST PRIORITY JOB FROM THE QUEUE, READ THE	* 00220000
* APPROPRIATE SCT FOR THE JOB, AND SET UP THE MESSAGE CLASS	CR17 00240017
* QUEUE MANAGER PARAMETER AREA.	CR17 00260017
* ALSO,	00280000
* TO PREPARE A DATA SET END PARAMETER LIST AND	* 00300000
* A DSNAME DEQ TABLE	* 00320000
* CONDITIONALLY INVOKE A DIRECT SYSOUT (DSO)	099 00322019
* ROUTINE TO SELECT ELIGIBLE DSO DEVICES TO BE	099 00324019
* USED BY THE DEQUEUED JOB. JOB CLASS AND	099 00326019
* SYSOUT CLASS ARE USED TO SELECT ELIGIBLE DSO	099 00328019
* DEVICES. DIRECT SYSOUT CONTROL BLOCKS (DSOCH)	099 00330019
* CONTAIN NECESSARY DATA FOR SELECTION AND	099 00332019
* MODIFICATION OF THE RESPECTIVE SYSOUT'S SIOT	099 00334019
* AND JFCB TO THAT OF THE DSO DEVICE.	099 00336019
*	00340000
* ENTRY POINTS - IEFSD161 FROM IEFSD160,IEFSD166,AND IEFSD105	00360014
*	00380000
* INPUT: REGISTER ONE CONTAINS THE LCT ADDRESS	* 00400000
*	* 00420000
* OUTPUT: REGISTER ONE CONTAINS THE LCT ADDRESS AND THE LCT CONTAINS	* 00440000
* THE CURRENT JCT AND SCT CORE ADDRESSES	* 00460000
*	* 00480000
*	00500000
*EXTERNAL ROUTINES: IEFQMDQ2 - REQUEST WORK	00540000
* IEFQMUNC - DELETE WORK REQUEST	00560000
* IEFQMRAW - Q MNGR READ/WRITE ROUTINE	00580000
* IEFQDSTBL - DATA SET INTEGRITY	00600000
* IEFQDSLST - DATA SET INTEGRITY	00620000
*	00640000
*	* 00660000
* EXITS - 1. JOB TO PROCESS - IEFSD162	00670014
* 2. STOP INIT - BACK TO MASTER	00680014
* 3. MODIFY INITIATOR - IEF060SD	* 00686015
* 4. WAIT FOR WORK - IEFSD105	* 00692015
* 5. DIRECT SYSOUT (DSO) SELECTION - IEFDSOSL	099 00694019

```

*
* EXTERNAL REFERENCES: IEFQSSS,IEFDSTHL
* TABLES/WORK AREAS: JCT,SCT, LIFE OF TASK BLOCK, AND JOB CSCB ARE
*                      IN MAIN STORAGE AT EXIT
*                      QUEUE MANAGER RESIDENT CORE AREA
*                      DIRECT SYSOUT CONTROL BLOCKS (DSOCB)
*
*ATTRIBUTES REENTRANT
*
*****
R0      EQU 0
R1      EQU 1
R2      EQU 2
R3      EQU 3
R4      EQU 4
R5      EQU 5
R6      EQU 6
R7      EQU 7
R8      EQU 8
R9      EQU 9
R10     EQU 10
R11     EQU 11
R12     EQU 12
R13     EQU 13
R14     EQU 14
R15     EQU 15
BALR R12,0
        SET PROGRAM BASE REGISTER
        USING IEFJCT,R11
        USING IEFL0T,R10
        USING IOPARAMS,R9
        USING RSAR,R13
        USING *,R12
*****
* THIS SECTION OF CODE IS FOR PROVIDING A METHOD OF IDENTIFYING
* THIS MODULE IN A MEMORY DUMP
*
B FIRST
DC X'CCCC'
DC C'IEFSD161'
DC X'05031966'
FIRST DS OH
*****
EJECT
LR R10,R1
LA R9,QMGR1
*
LA R13,REGSAVE
*
*
TM LCTOPSW1,LCTSTMDP
BND ECBTEST
TM LCTINTSW,LCTSTOP
BND SD612021
        GET BASE FOR LIFE OF TASK BLOCK
        GET BASE FOR QUEUE MANAGER PARAMETER
        AREA(QMPA)
        GET BASE FOR RSAR DSECT(THE 1ST 18
        WORDS OF WHICH SERVE AS A SAVE AREA
        FOR THIS PROGRAM)
        STOP/MODIFY PROCESSING WANTED
        YES, GO CHECK FOR STOP
        INTERNAL STOP WANTED
        NO, GO PROCESS THE JOB
*****
099 00696019
00700000
00720000
00740000
00760015
099 00770019
00780015
00800015
00820015
00840000
00860000
00880000
00900000
00920000
00940000
00960000
00980000
01000000
01020000
01040000
01060000
01080000
01100000
01120000
01140000
01160000
01180000
01200000
01220000
01240000
01260000
01280000
01310014
01320000
01340000
01360000
01380000
01400000
01420000
01440000
01460000
01480000
01500000
01520000
01540000
01560000
01580000
01610015
01640014
01670014
01700014
01730014
01760014
1272 01760419
1272 01760819
1272 01761219
1272 01761619

```

	R	SD061037	YES GO PREPARE TO STOP	1272	01762019
ECBTEST	EQU	*	TEST FOR EXTERNAL STOP	1272	01762419
	L	R0,ECBLIST	GET POINTER TO ECB LIST		01763015
	L	R1,0(RR)	LOAD ADDRESS OF COMMUNICATION ECB		01766015
	TM	0(R1).STOPBIT	IS THE STOP BIT POSTED		01769015
	RD	SD61550	YES BRANCH TO STOP		01772015
SD61549	EQU	*			01775015
	USING	COMP, R2	COMM PARM AREA ADDRESSABILITY	1272	01776019
	L	R2,LCCTCOM	NO, COMM PARM AREA PTR -CPA	1272	01777019
	L	R3,COMCIBPT	POINTER TO CIB	1272	01778019
	DROP	R2		1272	01779019
	LA	R3,0(R3)	GET RID OF HIGH ORDER BYTE		01784015
	LTR	R3,R3	IS IT ZERO		01787015
	RZ	SD612021	YES, BRANCH AND SCAN REST OF ECBLIST		01790015
	RAL	R7,SD612270	BRANCH TO UNCHAIN ECBS		01796015
*		ADDRESS OF LCT IS IN R10 FOR IEFS0160		1272	01798019
	L	R15,IEFS060V	ADDRESS OF MODULE IEFS0060		01801015
	RR	R15	GO TO IEFS0060		01803015
SD61550	NI	0(R1),X'AF'	TURN OFF POST AND STOP BITS		01805015
	END	MF=(F,IEFS061C)	END ON GROUP CONTROL BLOCK CHAIN		01808015
	L	R7,CVTPTR	LOAD POINTER TO CVT		01811015
	USING	CVTDSECT,R7			01814015
	L	R7,CVTMSER	LOAD ADDRESS OF BASE A		01817015
	USING	BASE,R7			01820015
	LA	R5,BATRM	LOAD ADD OF GCB POINTER		01823015
	L	R4,BATRM	LOAD GCB POINTER		01826015
	DROP	R7			01829015
SD61551	EQU	*			01832015
	CLC	16(R,R4),LCTIDENT	ARE IDENTIFIERS EQUAL		01841015
	RE	SD61552	YES, BRANCH AND CONTINUE		01844015
	LA	R5,4(R4)	NO, SAVE ADDRESS OF LAST GCB PLUS 4		01847015
	L	R4,4(R4)	LOAD POINTER TO NEXT GCB		01850015
	R	SD61551	BRANCH AND CHECK THIS GCB		01853015
SD61552	EQU	*			01856015
	SR	R7,R7	ZERO OUT REGISTER		01859015
	IC	R7,25(R4)	SET STOP COUNT		01862015
	LTR	R7,R7	IS STOP COUNT ZERO		01865015
	BNZ	SD61553	NO, BRANCH AND CONTINUE		01868015
	USING	COMP, R2	COMMUN PARM AREA ADDRESSABILITY	1272	01868219
	L	R2,LCCTCOM	ADDRESS OF COMMUN PARM AREA	1272	01868419
	L	R1,COMCIBPT	ADDRESS OF CIB	1272	01868619
	USING	CIBDSECT,R1	CIB ADDRESSABILITY	1272	01868819
	LTR	R1,R1	IS THERE A CIB	1272	01869019
	RZ	SD61549A	NO, GO CHECK FOR MODIFY	1272	01869219
	CLI	CIBVERB,CIBSTOP	IS THIS STOP CIB	1272	01869419
	RNE	SD61549A	NO, GO CHECK FOR MODIFY	1272	01869619
	LA	R0,COMCIBPT	ADDRESS OF CIB PTR	1272	01869819
	DROP	R2		1272	01870019
	R	FREESTOP	GO FREE STOP CIB	1272	01870219
SD61549A	EQU	*		1272	01870419
	DEQ	MF=(E,IEFS061C)	DEQ ON GROUP CONTROL BLOCK CHAIN		01871015
	R	SD61549	GO CHECK FOR MODIFY		01874015
SD61553	EQU	*			01877015
	RCTR	R7,R0	DECREMENT STOP COUNT		01880015
	STC	R7,25(R4)	STORE STOP COUNT		01883015
	LA	R0,28(R4)	ADDRESS OF CIB POINTER IN GCB		01883718
	L	R1,28(R4)	POINTER TO THE CIB		01884418

```

FREESTOP EQU *
QEDIT BLOCK=(1),ORIGIN=(0) FREE THE STOP CIR 01885118
IC R7,24(R4) LOAD MEMBER COUNT 01886015
BCTR R7,R0 DECREMENT MEMBER COUNT 01889015
LTR R7,R7 IS MEMBER COUNT ZERO 01892015
BNZ SD61554 NO, BRANCH AND CONTINUE 01895015
MVC O(4,R5),4(R4) YES, MOVE POINTER TO NEXT GCB 01898015
* INTO LAST GCB 01901015
LH R0,0(R4) LOAD LENGTH OF GCB 01904015
LA R1,255 LOAD SUBPOOL 01907015
SLL R1,24 MOVE SUBPOOL TO HIGH ORDER BYTE 01910015
OR R0,R1 OR IN SUBPOOL NUMBER 01913015
LR R1,R4 LOAD ADDRESS OF GCB 01916015
FREEMAIN R,LV=(0),A=(1) FREE GCB 01919015
B SD61555 BRANCH AND CONTINUE 01922015
SD61554 EQU * 01925015
STC R7,24(R4) STORE MEMBER COUNT 01928015
USING COMPA,R7 CPA ADDRESSABILITY 1272 01929019
L R7,LCTCOM CPA POINTER 1272 01930019
L R7,COMCIBPT ADDRESS OF CIR 1272 01931019
DROP R7 1272 01932019
LA R7,0(R7) GET RID OF HIGH ORDER BYTE 01937015
LTR R7,R7 IS A MODIFY OUTSTANDING 01940015
BZ SD61555 NO, BRANCH AND CONTINUE 01943015
SR R7,R7 YES, ZERO REGISTER 01946015
IC R7,26(R4) LOAD MODIFY COUNT 01949015
BCTR R7,R0 DECREMENT MODIFY COUNT 01952015
STC R7,26(R4) PUT MODIFY COUNT BACK 01955015
SD61555 EQU * 01958015
DEQ MF=(E,IEFSD61C) DEQ ON GROUP CONTROL BLOCK CHAIN 01961015
B SD061037 01964015
SD612021 EQU * 01967015
L R11,LCTJCTAD LOAD ADDRESS OF JCT 1272 01975019
LTR R11,R11 IS THERE AN IN-CORE JCT 1272 01983019
BNZ JCTIN YES, DONT GET CORE FOR JCT 1272 01991019
GETMAIN R,LV=176,SP=TCORE GET CORE FOR JCT 02000014
LR R11,R1 GET BASE FOR JCT 02030014
***** 02060014
SD612020 EQU * 02080015
LA R8,4(R8) POINT TO NEXT ECB POINTER 02100015
L R1,0(R8) LOAD ECB POINTER 02120015
TM O(R1),POSTBIT IS IT POSTED 02140015
RD SD612010 YES, BRANCH TO SET UP FOR Q-MANAGER 02160015
SD612030 EQU * 02180015
TM O(R8),ENDBIT IS THIS END OF ECB LIST 02200015
BZ SD612020 NO, BRANCH AND CHECK NEXT ECB 02220015
H SD061028 YES, SET UP FOR WAIT ROUTINE 02240015
***** 02660014
* SET UP TO GO TO QUEUE MANAGER AND SEE IF THERE IS WORK * 02690014
***** 02720014
SD612010 EQU * 02740015
LA R3,REMLST LOAD ADDRESS OF REMOTE AREA 02760015
ST R3,QMPCL PLACE ADDRESS IN QMPA 02780015
MVI QMPOP,QMDTOP INDICATE DEQUEUE 02800015
MVC QMTPY(1),4(R1) INDICATE JOB CLASS 02820015
ST R11,LCTJCTAD PUT JCT CORE ADDR IN LCT 02870014
ST R11,REMLST PUT JCT CORE ADDR IN REMOTE 02900014

```

```

*          XC      REMST+4(4),REMLST+4      LIST      02930014
          ST      R1,QMCAN      ZERO ITR SLOT IN REMOTE LIST      02960014
          L      R15,IEFQMVCN      GET ADDR OF Q MANAGER RTN      03020015
          LR      R1,R9      QMPA ADDR INTO R1      03110014
          HALR     R14,R15      GO TO QUEUE MANAGER TO TRY TO      03140014
          *          GET WORK      03170014
          B      SD061020(R15)      BRANCH INDEXED BY RETURN CODE      03200014
SD061020 B      SD061040      JOB DEQUEUED - GO TO PROCESS IT      03230014
          B      SD612030      NO WORK, CONTINUE CHECKING      03260014
          RC      0,SD061020      NO-OP,NO SPACE      03290015
          ***** I/O ERROR ***** TEMP ABEND AND LOOP      03320014
          B      BAD10      COND CODE=12, GO TO ABEND ROUTINE      03350014
          *****      099      03410019
          SD061028 TM      WTOSWCH,X'FF'      CHECK TO SEE IF 'INPUT QUEUE EMPTY'      03470014
          *          MESSAGE HAS ALREADY BEEN PUT OUT      03500014
          *          ONCE DURING THIS INITIATOR'S WAIT      03530014
          *          FOR WORK      03560014
          RC      8,SD061030      YES - BRANCH AROUND WTO      03590014
          NI      WTOSWCH,X'00'      TURN OF SWITCH SO MESSAGE WILL NOT      03620014
          *          BE PUT OUT AGAIN UNTIL AFTER THIS      03650014
          *          INIT HAS RECEIVED WORK AND PROCESSED      03680014
          *          IT AND GONE INTO A WAIT AGAIN      03710014
          LA      R0,WTO429EX      SIZE OF WTO AND INIT ID      A27668 03740019
          GETMAIN R,LV=(0)      CORE FOR MSG      A27668 03744019
          *          R1 WILL BE MAINTAINED FOR MSG AREA BASE      A27668 03746019
          L      R2,MSGAD      WTO CSECT      A27668 03748019
          MVC     DO(WTO429A,R1),DO(R2)      1ST HALF OF WTO MSG & QUOTE A27668 03750019
          LA      R15,WTO429A(R1)      WHERE INIT ID IS TO GO      A27668 03752119
          LR      R0,R15      SAVE FOR LENGTH (LNG) COMPUTE      A27668 03753019
          MVC     DO(D8,R15),LCTIDENT INIT FOR MSG      A27668 03757019
          LA      R14,D8      LOOP LIMITING REG      A27668 03758019
          *          WILL NOW FIND 1ST BLANK IN ID IF ONE      A27668 03760019
SD061750 CLI     DO(R15),BLANK      IS ID BYTE BLANK      A27668 03762019
          BE      SD61751      YES - END OF ID      A27668 03764019
          LA      R15,D1(R15)      NO - TEST NEXT HIGHER BYTE      A27668 03766019
          BCT     R14,SD61750      LOOP IF NOT END OF ID SCAN      A27668 03768019
          *          R15 NOW POINTS TO BYTE AFTER END OF ID      A27668 03770019
SD061751 MVC     DO(WTO429B,R15),WTO429A(R2)      LAST HALF OF MSG      A27668 03772019
          *          STARTING WITH SECOND QUOTE      A27668 03774019
          SR      R15,R0      COMPUTE ID LNG      A27668 03776019
          AH      R15,DO(R1)      ADD WTO MSG LNG TO THIS VALUE      A27668 03778019
          STH     R15,DO(R1)      UPDATE WTO MSG LNG      A27668 03780019
          LR      R2,R1      SAVE MSG ADR      A27668 03782019
          WTO     MF=(E,(1))      ISSUE MSG TO OPERATOR      A27668 03784019
          LA      R0,WTO429EX      SIZE OF WTO AND INIT ID      A27668 03786019
          LR      R1,R2      SET R1 AGAIN WITH MSG CORE ADR      A27668 03788019
          FREEMAIN R,LV=(0),A=(1)      FREE MSG CORE      A27668 03790019
SD061030 LR      R1,R11      GET JCT ADDRESS      03810014
          FREEMAIN R,LV=176,A=(1),SP=TCORE      FREE JCT CORE      03820014
          XC      LCTJCTAD(D4),LCTJCTAD      ZERO OUT JCT ADDRESS      1272 03825019
          LR      R1,R10      GET LIFE OF TASK BLOCK ADDR      0389
          L      15,16      MOTHER CVT      0390
          L      15,0(0,15)      DOUBLEWORD      0391
          L      15,4(0,15)      THE TCB      0392
          L      15,12(0,15)      OUR TIOT      0393
          CLI     0(15),C'F'      FIXED PARTITION?      0394

```

HNE	LACTATE	NO		0395
L	R15,IEFSD62V	GET ADDRESS OF IEFSD062.		
LA	R15,DISPLACE(R15)	RUMP IT TO FIT FLAG.		
MVI	0(R15),X'FF'	SET FIT FLAG ON.		
HITIT	L R1,ECBLIST	LOAD ECB LIST PTR.		
WAIT	1,ECBLIST=(1)	WAIT FOR WORK		0397
R	FIRST+2	SKIP MICKEY MOUSE		0398
LACTATE	EQU *			0399
XCTL	EP=IEFSD105,MF=(E,(1))	XCTL TO LINK PACK AREA TO WAIT		03890014
*		FOR WORK OR STOP INIT - THUS		03920014
*		MAKING AVAILABLE THE SPACE THE		03950014
*		IEFSD061 LOAD IS OCCUPYING		03980014
*****				04010014
SD061037	EQU *			04040015
L	R3,LCTSREG			04070015
TM	LCTOPSW1,LCTICMDF	PROCESSING INITIATOR COMMANDS	M3161	04100019
BO	NOTTHRES	NO, DONT REDUCE THRESHOLD	M3161	04130019
L	R7,16	ADDRESS OF CVT		04160000
USING	CVT0SECT,R7			04180000
L	R7,CVTJOB	GET ADDRESS OF RESID CORE		04200000
USING	QMRDSECT,R7			04220000
*				04240000
END	MF=(E,IEFSD61B)	ENQUEUE ON ASSIGN/DELETE		04260000
LH	RC,QMTBT	GET SIZE OF THRESHOLD		04280000
TM	QMQRK,QMRSV	TEST IF HAVE RESERVE		04300000
BZ	SD061038	NO-THIS INIT'S THRSOLD WILL BE RESV		04320000
SH	RO,QMTRS	SUB SIZE GARENTEED AN INIT		04340000
STH	RO,QMTBT	SAVE NEW SIZE		04360000
SD061038	OI QMQRK,QMRSV	SET RESERVE BIT ON		04380000
DEQ	MF=(E,IEFSD61B)	DEQUEUE ON ASSIGN/DELETE		04400000
NOTTHRES	EQU *		M3161	04460019
LA	R7,SD612073			04522015
SD612270	EQU *			04524015
SD612070	EQU *			04528015
L	R6,ECBLIST	ADDRESS OF ECBLIST	1272	04528319
LR	R8,R6	SAVE ECBLIST ADDRESS	1272	04528619
LTR	R8,R8	IS THERE AN ECBLIST	1272	04528919
BCR	D8,R7	NO, DONT UNCHAIN IT	1272	04529219
NEXTECB	EQU *		1272	04529519
LA	R8,4(R8)	LOAD ADDR OF POINTER TO ECB		04530015
L	R1,0(R8)	LOAD ECB POINTER		04532015
LA	R1,0(R1)	CLEAR HIGH ORDER BYTE		04533015
TM	0(R1),POSTBIT	TEST FOR POSTED		04534015
BZ	SD612072	NO, BRANCH AND UNCHAIN		04536015
SD612071	EQU *			04538015
TM	0(R8),ENDBIT	TEST FOR END OF LIST		04540015
BZ	NEXTECB	NO, GO TEST NEXT ECB	1272	04542019
BR	R7	YES RETURN		04544015
SD612072	EQU *			04546015
L	R15,QMNCVC0N	GET ADDRESS OF UNCHAIN ROUTINE		04548015
BALR	R14,R15	GO TO UNCHAIN		04550015
MVI	0(R1),X'40'	POST UNCHAINED ECB		04551016
B	SD612071	GO CHECK FOR END OF LIST		04552015
SD612073	EQU *			04552515
LTR	R6,R6	IS THERE AN ECBLIST	1272	04553519
BZ	TESTKEY	NO, DONT FREE IT	1272	04554519
LR	R1,R6	LOAD POINTER TO ECB LIST		04556015

LA	R0,4	LOAD CONSTANT OF 4	04558015
SR	R1,R0	SUBTRACT TO GET TO LENGTH OF ECBLIST	04560015
L	R0,0(R1)	LOAD LENGTH	04562015
LA	R8,TCORF		04564015
SLL	R8,24		04566015
OR	R0,R8		04568015
FREEMAIN	R, LV=(0), A=(1)		04570015
TESTKEY	FQU *		1272 04572019
TM	LCTOPSW1, LCTPKFYF	WAS PROTECT KEY GOTTEN	1272 04574019
NO	NOTFREE	NO, SO DONT FREE IT	1272 04576019
L	R6,16	LOAD ADDR OF CVT INTO BASE REG 6	04580000
USING	CVTDSECT, R6		04600000
L	R6, CVTICHP	LOAD ADDR OF TCB INTO BASE REG 6	04620000
L	R6, 4(R6)		04640000
XR	R1, R1	CLEAR REG 1	04660000
IC	R1, 28(6)	INSERT PROTECT KEY INTO BASE REG 1	04680000
SRL	R1, 4(0)	MOVE KEY TO LOW ORDER BITS OF REG	04700000
LA	R2, 1(0)	LOAD 1 INTO REG 2	04720000
SLL	R2, 0(1)	SHIFT LEFT KEY NUMBER OF BITS	04740000
L	R6, 16	LOAD ADDR OF CVT INTO BASE REG 6	04760000
USING	CVTDSECT, R6		04780000
L	R6, CVTMSER	LOAD ADDR OF MASTER INTO REG 6	04783019
USING	BASE, R6		04786019
EX	R2, EXXI	TURN OFF KEY SWITCH	04789019
SRL	R2, 8(0)	SHIFT TO HI EIGHT BITS	04792019
FX	R2, EXXIA	TURN OFF KEY SWITCH	04795019
NOTFREE	FQU *		1272 04798019
L	R9, LCTEXIT	SAVE ADDRESS OF EXIT LIST	1272 04801019
*****			04804019
LR	R1, R10	ADDRESS OF LIFE OF TASK BLOCK	04807019
LA	R0, IEFEND-IEFLOT	LENGTH OF LOT	04810019
LA	R14, TCORE	SUBPOOL NUMBER	04813019
SLL	R14, 24	PLACE IN HIGH ORDER BYTE OF	04816019
OR	R0, R14	LENGTH REGISTER	04819019
FREEMAIN	R, LV=(0), A=(1)	FREE LOT	04822019
LR	R1, R11		04825019
*****			04828019
USING	IELEXITS, R9	EXIT LIST ADDRESSABILITY	1272 04831019
LA	R9, ZERO(R9)	ZERO HIGH ORDER BYTE	1272 04834019
LTR	R9, R9	IS THERE AN EXIT LIST	1272 04837019
BZ	RETURN	NO, EXIT TO MASTER SCHED	1272 04840019
LR	R1, R9	EXIT LIST ADDRESS	1272 04843019
CLI	RTNEXLK, IELEXADD	IS THIS A BRANCH EXIT	1272 04846019
BE	BRANCH	YES, PREPARE TO BRANCH	1272 04849019
LA	R0, D16	REMOTE LIST + SUP PARMLIST	M4339 04852019
GETMAIN	R, LV=(0)	GET CORE FOR REMOTE LIST	M4339 04855019
XC	ZERO(D16, R1), ZFRQ(R1)	CLEAR CORE	M4339 04858019
LR	R15, R1	ADDR OF SUP PARMLIST	M4339 04861019
LA	R2, D8(R1)	ADDR OF REMOTE LIST	M4339 04864019
LR	R1, R9	EXIT LIST ADDRESS	M4339 04867019
MVC	ZERO(D6, R2), RTNEXNM	MOVE EXIT NAME	M4339 04870019
MVI	D6(R2), BLANK	PAD WITH BLANKS	M4339 04873019
MVI	D7(R2), BLANK	PAD WITH BLANKS	M4339 04876019
ST	R2, ZERO(R15)	REMOTE LIST ADDR	M4339 04879019
LR	R2, R15	ADDR OF SUP PARMLIST	M4339 04882019
CLI	RTNEXLK, IELEXCTL	IS EXIT BY XCTL	1272 04889619
BE	GOXCTL	YES, PREPARE TO XCTL	1272 04889719

```

*   IF EXIT IS NOT BY BR OR XCTL, LINK IS ASSUMED
LINK MF=(E,(1)),SF=(E,(15)) GO TO EXIT ROUTINE
R RETURN GO RETURN
GOXCTL EQU *
XCTL MF=(E,(1)),SF=(E,(15)) GO TO EXIT ROUTINE
BRANCH EQU *
L R15,RTNEXAD ADDRESS OF EXIT ROUTINE
BR R15 BRANCH TO EXIT ROUTINE
RETURN EQU * RETURN TO ATTACHER
LR R14,R3 INITIALIZE RETURN REGISTER
BR R14
MOVEMES MVC O(O,R3),31(R2)
ENDBIT EQU X'80'
*****
EXXI XI BARSW+5,0 TURN OFF PROTECT KEY
EXXIA XI BARSW+4,0
SD061040 EQU *
USING IEFCSCB,R1 CSCB ADDRESSABILITY
OI CHSTS,CHDL INDICATE FREE WHEN JOB TERM
DROP R1
ST R1,LCTQDRTY ADDRESS OF JOB CSCB
USING IOPARAMS,R9 OMGR PARAM AREA ADDRESSABILITY
JCTIN EQU *
LA R0,WTPCBSIZ SIZE OF WTPCB
O R0,M253 SUBPOOL IN HIGH ORDER BYTE
GETMAIN R,LV=(0) GET CORE FOR WTPCB
LR R6,R1 SAVE ADDRESS
XC ZERO(WTPCBSIZ,R1),ZERO(R1) ZERO WTPCB
GETJSCB SYSTEM=MVT GET MVT JSCB
USING IEZJSCB,R1 JSCB ADDRESSABILITY
ST R1,LCTJSCB ADDRESS OF JSCB TO LCT
ST R6,JSCBWTB ADDRESS OF WTPCB TO JSCB
MVC JSCRCSCB+ONE(D3),LCTQDRTY+ONE ADDR CSCB
TM LCTOPSW2,LCTTIMEF IS TIMING WANTED
BND YESTIME YES, CONTINUE
MVC JCTJMRTL,HOUR24 NO, INSERT MAX TO
* INDICATE NO TIMING
YESTIME EQU *
SR R0,R0 CLEAR REGISTER
STC R0,FRCPRTY CLEAR FORCE SLOT IN LCT
L R1,ECBLIST ADDRESS OF ECBLIST
LTR R1,R1 IS THERE AN ECBLIST
BZ NOFORCE NO, DONT INSERT FORCE IN LCT
LA R1,4(R1) LOAD ADDRESS OF FIRST JOB ECB
SR R8,R1 DETERMINE OFFSET INTO
SRL R8,2 FORCE TABLE
LA R1,LCTFORCE LOAD ADDRESS OF FORCE TABLE
IC R0,0(R8,R1) LOAD FORCE VALUE
LTR R0,R0 WAS FORCE SPECIFIED
BZ SD061044 NO, BRANCH AND CONTINUE
STC R0,FRCPRTY YES, STORE FORCE PRTY IN SLOT
SD061044 EQU *
L R8,ECBLIST LOAD POINTER TO ECBLIST
BAL R7,SD612070 GO UNCHAIN ECBS
NOFORCE EQU *
L R1,LCTQDRTY RESTORE REGISTER ONE
*****

```

```

MVN    JCTJPRTY(1),QMPRI  REPLACE THE JOB'S ORIGINAL      CR17 04984017
*      PRIORITY IN THE JCT WITH ITS PRESENT      04986014
*      PRIORITY FROM THE QMPA. IN CASE ITS      04988014
*      PRIORITY WAS CHANGED WHILE IN THE      04990014
*      INPUT QUEUE                                04992014
***** 04994014
OI      QMPRI,X'20'          INDICATE INITIATOR QMPA      994 05000000
L      R6,16                LOAD ADDR OF CVT INTO BASE R6 05020000
USING  CVTDSECT,R6          05040000
L      R6,CVTTCBP           05060000
L      R6,4(R6)             LOAD ADDR OF TCB INTO BASE R6 05080000
L      R5,12(R6)           LOAD ADDR OF TIOT INTO R5      05100000
DROP   R6                   05120000
TM      LCTOPSW1,LCTICMDF   PROCESS INIT COMMANDS      M4339 05126019
BO      NOMVJBNM           NO, DONT UPDATE T&OT      M4339 05132019
MVC     R(8,R5),QMNAM       MOVE JOBNAM INTO TIOT SSS FIELD 05140000
NOMVJBNM EQU *              M4339 05150019
TM      QMSTA,CANCELED     WAS JOB CANCELLED IN-QUEUE    05160000
RC      8,SD061042        NO, CONTINUE                05180000
USING  IELEXITS,R2         EXIT LIST ADDRESSABILITY      I272 05182019
L      R2,LCTEXIT          ADDRESS OF EXIT LIST          I272 05184019
LA      R2,ZERO(R2)        CLEAR HIGH-ORDER BYTE        I272 05186019
LTR     R2,R2              IS THERE AN EXIT LIST         I272 05188019
BZ      NOEXITL            NO, NO RETURN CODE            I272 05190019
MVI     IELRTNCD,IERR      YES, INSERT I/O RETURN CODE   I272 05192019
NOEXITL EQU *              I272 05194019
OI      JCTJSTAT,INCMSTS   YES, SET JOB-FAIL BIT IN JCT 05200000
OI      LCTQNTY,LCTERRM    TURN ON TERM BIT             05202014
XC      JCTDETD(3),JCTDETD ZERO OUT TTR OF DSENG TABLE 05205014
*      SO DATA SETS WILL NOT BE
*      ENQUEUED UPON      05210014
*      05215014
SD061042 EQU *            05220000
***** 05240000
*      INSERT PROTECT KEY FROM * 05260000
*      INITIATOR TCB AND PLAC * 05280000
*      IN JCT FOR UNIQUE JOB * 05300000
*      SERIAL NO. * 05320000
***** 05340000
USING  IEFCSCB,R1         05380000
*      05400000
*      05420000
*      05440000
*      05460000
*      05480000
*      05500000
*      05520000
*      05540000
*      05560000
LA      R2,CHKEY           GET POINTER TO JOB NAME      05580000
ST      R2,TRSTKINF+4      PUT IN 0 AND JOB NAME PTR    05600000
MVC     TRSTKINF+4(1),JCTNTRK MOVE TRACKS USED      05603018
MVC     CHJCL(4),JCTDSKAD  PLACE JCT TTR IN CSCB OF JD CR17 05606017
MVI     CHJCL+3,X'00'      BE SURE LAST BYTE IS ZERO    CR17 05612017
L      R1,CHSWT           R1 = TCB ADDR                05620000
TCBPKF EQU 28              05640000
DROP   R1                  M2942 05641019
L      R2,LCTQDRTY         ADDRESS OF CSCB              M2942 05642019

```

USING	IEFCSB,R2	CSCB ADDRESSABILITY	M2942	05643019
TM	CHSWT,CHSYS	IS THIS FOR A SYSTEM TASK	M2942	05644019
BND	NOTZEROK	NO, PUT IN PROTECT KEY	M2942	05645019
DROP	R2		M2942	05646019
* 119450			M4008	05647019
MVI	JCTJSRNO,FOXFOX	YES, INSERT NONZERO NUMBER	I272	05649019
B	ZERUKEY	SKIP MOVE FROM ICB	I272	05652019
NOTZEROK	EQU *		I272	05655019
MVC	JCTJSRNO(1),TCBPKF(R1)	MOVE PROTECT KEY TO JCT		05660000
ZERUKEY	EQU *		I272	05670019
DROP	R1			05680000
GETMAIN	R,LV=176,SP=TCORE	GET CORE FOR SCT		05700000
ST	R1,REMLST	ADDRESS OF GOTTEN CORE		05720000
ST	R1,LCTSTAD	SCT ADDRESS TO LCT		05740000
MVC	REMLST+4(4),JCTSSR	ALTERNATE SCT TTR IN REMOT CR17		05770017
SR	R2,R2	ZERO A REGISTER TO		05800000
ST	R2,QMCAN	ZERO OUT ECB SLOT		05820000
MVI	QMPCL,1	INDICATE ONE OPERATION		05840000
MVI	QMPDP,4	INDICATE READ OPERATION		05860000
LR	R1,R9	ADDRESS OF Q MGR PAR AREA		05880000
L	R15,IEFQMRD	ADDRESS OF READ MODULE		05900000
BALR	R14,R15	GO TO THE READ MODULE		05920000
LTR	R15,R15	ANY NON-ZERO COND CODE	099	05980019
BNZ	BADIO	YES - GO TO ABEND ROUTINE	099	06040019
*****				06120000
*****				06140000
*			*	06160000
* AT THIS POINT, CHANGE THE INITIATOR'S PRIORITY TO REFLECT THE			*	06180000
* PRIORITY OF THE JOB GOTTEN FROM THE QUEUE			*	06200000
*			*	06220000
*****				06240000
SD061060	EQU *			06260000
L	R1,LCTSTAD	LOAD ADDRESS OF SCT	CR17	06265017
USING	IEFSCT,R1		CR17	06270017
DI	SCTSDP,SCTFSTEP	TURN ON FIRST STEP BIT	CR17	06275017
SR	R1,R1	ZERO OUT R1		06280000
IC	R1,QMPRI	MSGLEVEL AND PRIORITY TO R1		06300000
LA	RO,X'OF'	LOAD A MASK		06320000
NR	R1,RO	'AND' OUT MESSAGE LEVEL		06340000
STC	R1,LCTQDRTY	PLACE LPMOD VALUE IN LCT	I241	06430018
*****				06520000
* THE FOLLOWING CODE WILL READ THE SYSOUT CLASS DIRECTORY			*	06540000
* FROM DISK AND FROM THE FIRST ENTRY IN THIS DIRECTORY,WHICH			*	06560000
* IS THE MSGCLASS ENTRY,WILL BUILD THE MSGCLASS QMANAGER			*	06580000
* PARAMETER AREA. THE POINTER TO THIS NEW AREA WILL BE STORED			*	06600000
* IN THE LCT.(OFFSET 68)-			*	06620000
* DIRECT SYSOUT (DSO) ROUTINES WILL ALSO SCAN THE SCD LATER ON			099	06626019
* IN THIS PROGRAM.			099	06632019
*****				06640000
LA	R1,REGSAVE+72	Q MGR EXTERNAL PAR AREA		06660000
XC	O(8,R1),O(R1)	CLEAR AREA TO ZERO		06680000
ST	R1,QMPCL	STORE POINTER TO EXT PARM AREA		06700000
LR	R3,R1	SAVE POINTER		06720000
GETMAIN	R,LV=176	GETMAIN FOR THE SCD		06740000
*		DIRECTORY READ IN AREA	*	06760000
ST	R1,O(0,R3)	STORE POINTER TO READ IN AREA		06780000
*		INTO EXT PARAM AREA	*	06800000

```

MVC 4(3,R3),JCTDSSHA      MOVE SCD DISK ADDR INTO EXT      06820000
*                               PARAM AREA                      * 06840000
MVI QMPOP,X'04'           INDICATE READ TO QMANAGER      06860000
MVI QMPCL,X'01'           INDICATE ONE RECORD TO BE READ  06880000
LR R4,R1                   SAVE POINTER TO READ.IN AREA   06900000
LA R13,REGSAVE            POINTER TO SAVE AREA           06920000
LR R1,R9                   LOAD POINTER TO QMPA           06940000
L R15,IEFQMRD             06960000
BALR R14,R15              BRANCH TO QMANAGER             06980000
LTR R15,R15 ANY NON-ZERO COND CODE                      099 06986019
RNZ BADIO YES - GO TO ABEND ROUTINE                      099 06992019
***** 07000000
* THE FOLLOWING CODE WILL GET CORE FOR AND BUILD THE MSGCLASS * 07020000
* QMGR PARAM AREA AND ITS ASSOCIATED IOB * 07040000
***** 07060000
L R1,LCTSQMP              ADDRESS OF SYSOUT QUEUE         07080000
*                               MANAGER PARAMETER AREA       07100000
XC 0(36,R1),0(R1)         CLEAR GOTTEN AREA             07120000
MVC QMWTQ2(4),QMWTQ1      MOVE TRK STACK INFO ADDR FROM  07140000
*                               MAIN QMPA TO ALTERNATE ONE     07160000
MVC 9(3,R1),1(R4)         MOVE THE QMFLT AND QMTST FIELDS 07180000
*                               FROM THE MSGCLASS SYSOUT CLASS * 07200000
*                               DIRECTORY INTO THE MSGCLASS * 07220000
*                               QMGR PARAM AREA BEING BUILT * 07240000
MVC 14(1,R1),4(R4)        MOVE QMNOT FIELD INTO QMPA     07260000
MVC 20(2,R1),5(R4)        MOVE QMTID FIELD INTO QMPA     07280000
OI 17(R1),X'20'           INDICATE INITIATOR QMPA       994 07300000
LR R3,R1                   SAVE POINTER TO QMPA          07320000
***** 07340000
* GET CORE FOR, BUILD AND INITIALIZE THE IOB * 07360000
***** 07380000
LA R0,QIOBL+4             LENGTH OF ECB/IOB+4            07400000
LA R14,TCORE              GETCORE IN SUBPOOL 253         07420000
SLL R14,24                SHIFT TO HIGH ORDER BYTE     07440000
OR R0,R14                 OR INTO LENGTH REGISTER       07460000
LR R5,R0                   SAVE SUBPOOL NO. AND LENGTH   07480000
GETMAIN R,LV=(0)          GET CORE FOR ECB/IOB          07500000
ST R5,0(R1)               SAVE SUBPOOL AND LENGTH OF     07520000
*                               ECB/IOB IN PRIOR 4 BYTES       07540000
LA R5,4(R1)               ADDR OF ECB/IOB               07560000
ST R5,28(0,R3)            STORE ADDR IN QMFLA FIELD OF   07580000
*                               QMGR PARAM AREA * 07600000
MVC 0(QIOBL,R5),QMECB      MOVE BASIC ECB/IOB MAP TO GOTTEN 07620000
*                               AREA * 07640000
ST R5,QIOBA(R5)           SET ECB ADDR INTO IOB         07660000
LA R1,QIOBC(R5)           STORE ADDR OF CCW1 INTO        07680000
ST R1,QIOBB(R5)           IOB                           07700000
MVC QIOBD+1(3,R5),QIOBB+1(R5) AND INTO CCW2 FIELD      07720000
LA R1,QIOBC-5(R5)         ADDR OF IOBSEEK FLD + 3       07740000
ST R1,QIOBC(R5)           INTO THE IOB                  07760000
OI QIOBC(R5),QMCOP        07780000
L R1,CVTPTTR              GET CVT PTR                   07800000
USING CVTDSECT,R1         ADDRESS THE CVT               07820000
L R1,CVTJOB               ADDR OF SCHEDULER DCB         07840000
ST R1,QIOBCB(R5)          STORE DCB ADDR IN IOB SLOT    07860000
DROP R1                   07880000
*                               099 07880519

```

```

* * * * * DETERMINE IF DIRECT SYSOUT (DSO) IS TO          099 07881019
* BE INVOKED FOR THIS JOB.                                099 07881519
*                                                         099 07882019
*   TM   LCTOPSW2,LCTDSOF  IS DSO SUPPRESSED BY L-SHAPE    099 07882519
*   RD   NODSO             YES - BYPASS DSO TESTS          099 07883019
*   L    R1,CVTPTR         LOCATION OF CVT                 099 07883519
*   USING CVTDSECT,R1      CVT BASE                        099 07884019
*   L    R1,CVTMSER        ADR OF MASTER SCH DATA AREA    099 07884519
*   USING BASE,R1          M/S BASE REG                    099 07885019
*   L    R1,BADSO          FIRST DSOCH ADR                 099 07885519
*   DROP R1                                                        099 07886019
*   LA   R1,NO(,R1)        CLEAR H/O                      099 07886519
*   LTR  R1,R1              ANY ADR TO DSOCH CHAIN          099 07887019
*   RZ   NODSO              NO-BYPASS ALLOCATION OF DSOCH'S  099 07887519
*                                                         099 07888019
* * * * * FOLLOWING CODE IS INCASE THE DSO ROUTINE IS      099 07888519
* EVER MADE A SEPERATE MODULE SINCE CODE IS ALSO          099 07889019
* SIMILAR FOR IEFS0511 IN MFT.                            099 07889519
*                                                         099 07890019
*   L    R1,QMPCL          GET QMGR EXTERNAL PARM LIST     099 07890519
*   ST   R4,NO(,R1)        ADR OF SCD INTO PARM LIST FOR QMGR 099 07891019
*   MVC  LCTPARM1(N4),IEFQMRD  ADR OF Q MGR RD/WR          099 07891519
*   LA   R13,REGSAVE       REG SAVE AREA                  099 07892019
*   LR   R1,R10            ADR OF LOT                      099 07892519
*   L    R15,DSOVCON        ADR OF DSO SELECT ROUTINE      099 07893019
*   BALR R14,R15           BRANCH TO DSO SELECT ROUTINE    099 07893519
*                                                         099 07894019
*   LTR  R15,R15           ANY I/O ERRORS                  099 07894519
*   BNZ  BADIO             YES - GO TO ABEND                099 07895019
*                                                         099 07895519
*   NODSO EQU *                                                    099 07896019
*                                                         099 07896519
*   LR   R1,R4             LOAD POINTER TO SCD AREA        07900000
*   FREEMAIN R,LV=176,A={1}  FREE SCD CORE                 07920000
* *****                                                    07940000
* * * * *                                                    07960000
* * INITIALIZE TIMER WORK AREA FOR THE JOB,JOBLIB DCB PTR,QMOSC FIEL CR17 07970017
* * IN QMPA,AND STEP NO. IN LCT.                                CR17 07980017
* * * * *                                                    08000000
* *****                                                    08020000
* * * * *                                                    08040000
*   XC   LCTTMWRK(16),LCTTMWRK  ZERO TIMER WORK AREA      08060000
*   XC   LCTJOBLB,LCTJOBLB     ZERO THE JOBLIB POINTER    08080000
*   XC   QMOSC(1),QMOSC        ZERO OUT QMOSC FIELD IN Q CR17 08086017
*   MVC  LCTSNUMB(1),JCTSNUMB  MOVE STEP NO. FROM JCT TO CR17 08092017
* * * * *                                                    08100000
* *****                                                    08120000
* * * * * DATA SET INTEGRITY CODE                            08140000
* * * * *                                                    08160000
* * FUNCTION:1.TO READ THE DATA SET ENQ TABLE PREPARED BY THE INTERPRETER 08180000
* * 2.WRITE A NEW TABLE WITHOUT DUPLICATES FOR USE IN IEFS0166 08200000
* * 3.TO PREPARE AN ENQ PARAMETER LIST FOR USE IN IEFS0102. 08220000
* * * * *                                                    08240000
* *****                                                    08260000
* * * * *                                                    08280000
* * * * *                                                    08300000
*   USING DSNIN,R4
DIMODULE XC   LCTPARM1,LCTPARM1          CLEAR TO ZERO    08301019

```

USING	IEFCSCH,R1	CSCB ADDRESSABILITY	M4351	08302019
L	R1,LCTQDRTY	ADDRESS OF CSCB	M4351	08303019
TM	CHSWT,CHDSI	IS DATA SET INTEGRITY WANTED	M4351	08304019
BND	WANTDSI	YES. BRANCH	M4351	08305019
XC	JCTDETD(03),JCTDETD	NO, ZERO SO NO DEQ LATER	M4351	08306019
B	SD061200	BYPASS DATA SET INTEG PROC	M4351	08307019
WANTDSI	EQU	*	M4351	08308019
TM	JCTJSTAT,INCMSTS	HAS JOB BEEN CANCELLED	M4351	08309019
BND	GODSI	NO - GO SET UP OSI	M4351	08310019
XC	JCTDETD(03),JCTDETD	ZERO SO NO DEQ LATER	M4351	08311019
B	SD061200	BYPASS DATA SET INTEGRITY	M4351	08312019
GODSI	EQU	*	M4351	08313019
OC	JCTDETD(3),JCTDETD	IF TTR PARM EQ ZERO, THERE		08340000
BZ	SD061200	ARE NO DSNAMEs FOR JOB		08360000
*				08380000
XC	QEXPARM(40),QEXPARM	CLEAR WORK AREAS		08400000
GETMAIN	R,LV=368	DSNAME I/O AREAS		08420000
*		AND TREE-BUD AREA		08440000
LR	R4,R1	SAVE POINTER TO I/O AREA		08460000
*				08480000
*	INITIALIZE	PARAMETER LIST FOR TREE ROUTINES		08500000
XC	BUD,BUD	CLEAR TO ZERO		08520000
LA	R3,BUD			08540000
ST	R3,TREE	ADDR OF BUD		08560000
MVI	TREE,X'02'	HIGH ORDER BYTE		08580000
LA	R2,TREE-8			08600000
ST	R2,TBLPARM	STORE IN TBUILD PARM LIST		08620000
*				08640000
XC	OCTTR(14),OCTTR	CLEAR OUTPUT FIELDS		08660000
MVI	OTABID,X'0F'	TABLE ID IN OUTPUT RECORD		08680000
*				08700000
*	SETUP	Q MANAGER PARAMETERS AND READ		08720000
LA	R3,QEXPARM			08740000
ST	R3,QMPCL	STORE ADDR IN EXT PARM AREA		08760000
MVC	QTR,JCTDETD	STORE TTR FOR 1ST READ		08780000
LA	R2,DSTART	ADDR OF 1ST OUTPUT RECORD		08800000
SD061070	ST	R4,QEXPARM		08820000
MVI	QMPCL,X'04'	INDICATE READ TO QMGR		08840000
MVI	QMPCL,X'01'	INDICATE ONE REC TO BE READ		08860000
LA	R13,REGSAVE	POINTER TO SAVE AREA		08880000
LR	R1,R9	POINTER TO QMPA		08900000
L	R15,IEFQMRD			08920000
BALR	R14,R15	BRANCH TO QMANAGER		08940000
LTR	R15,R15	ANY NON-ZERO COND CODE	099	09000019
BNZ	BADIO	YES - GO TO AREND ROUTINE	099	09060019
OKRET	EQU	*		09120000
OC	TTRBUF1,TTRBUF1	TEST FOR ZERO TTR		09140000
BZ	SD061080			09160000
MVC	TTRBUF2,ICTTR	MOVE CURRENT TTR TO BUFFER2		09180000
B	SD061090			09200000
SD061080	MVC	TTRBUF1,ICTTR		09220000
SD061090	LA	R5,14(R4)		09240000
*		RECORD IN BLOCK		09260000
SD061100	LA	R6,1(R5)		09280000
MVC	LENGTH+1(1),1(R5)	STORE LGTH OF CURNT DSNAME		09300000
AH	R6,LENGTH	R6-ADDR OF LAST CHAR OF DSNAME		09320000
CLI	1(R5),X'00'	ZERO LENGTH INDICATES END		09340000

BE	SD061150	OFBLOCK	09360000
LA	R5,2(R5)	R5 NOW POINTS TO 1ST CHAR	09380000
*		OF DSNAME	09400000
LA	R1,TBLPARM	ADDR OF TBUILD PARM LIST	09420000
STM	R5,R6,DSNBEG	STORE TBUILD PARAMETERS	09440000
LA	R13,REGSAVE		09460000
L	R15,TBLVCON		09480000
BALR	R14,R15	BRANCH TOIEFDSTBL	09500000
*****			09520000
*			09540000
	UPON RETURN FROM IEFDSTBL,R1 CONTAINS ADDR-B OF A		09560000
	4 BYTE AREA CONTAINING THE ADDR OF A 4 BYTE AREA		09580000
	WHICH CONTAINS:		09600000
	BYTE 0-X'00' IF SHARED SRR		09620000
	X'00' IF EXCLUSIVE SRR		09640000
	BYTE 1+2-NOT USED		09660000
	BYTE 3-CONTAINS ZERO IF DSNAME INTRODUCED TO		09680000
	IEFDSTBL WAS NEW		09700000
*****			09720000
BCTR	R5,R0		09740000
BCTR	R5,R0	DECREMENT R5 BY 2	09760000
L	R1,8(R1)	LOAD ADDR OF 4BYTE BUD	09780000
OC	3(1,R1),3(R1)	TEST FOR NEW DSNAME	09800000
BZ	SD061110	BRANCH IF TRUE	09820000
NC	0(1,R1),0(R5)	UPDATE E/S ATTRIBUTE	09840000
B	SD061130		09860000
SD061110 MVC	0(1,R1),0(R5)	STORE 1ST E/S ATTRIBUTE	09880000
MVI	3(R1),X'01'	CHANGE DSNAME STATUS TO OLD	09900000
SD061120 LH	R3,LENGTH	*R3 WILL CONTAIN SPACE	09920000
LA	R3,2(R3)	NEEDED IN OUTPUT BLOCK	09940000
LA	R7,0END	*R7 WILL CONTAIN REMAINING	09960000
SR	R7,R2	SPACE	09980000
CLR	R3,R7		10000000
BH	SD061170	BRANCH IF NO SPACE	10020000
BCTR	R3,R0	SUB 1 FROM LENGTH(EXECUTE	10040000
*		INSTRUCTION REQUIRES ONE	10060000
*		LESS THAN ACTUAL LENGTH	10080000
EX	R3,SD061140	MOVE DSNAME TO OUTPUT AREA	10100000
LA	R2,1(R2,R3)	INDEX R2 TO NEXT OUTPUT REC	10120000
LH	R3,LENGTH	INCREASE DSNAME CHAR COUNT	10140000
A	R3,0CHAR	BY LENGTH OF LAST DSNAME	10160000
ST	R3,0CHAR		10180000
LH	R3,0CNT	INCREASE DSNAME COUNT BY 1	10200000
LA	R3,1(R3)		10220000
STH	R3,0CNT		10240000
SD061130 LA	R7,172(R4)	TEST FOR ANOTHER POSSIBLE	10260000
CLR	R6,R7	RECORD IN BLOCK	10280000
BH	SD061150	BRANCH IF NO POSSIBILITY	10300000
LA	R5,1(R6)	R5-ADDR OF NEXT RECORD	10320000
B	SD061100		10340000
SD061140 MVC	0(0,R2),0(R5)	OBJECT OF EX INSTRUCTION	10360000
SD061150 OC	1LTTR,1LTTR	END OF DSNAME ENQ TABLE IF	10380000
BZ	SD061160	LAST TTR FIELD IS ZERO	10400000
MVC	QTTR,1LTTR	MOVE NEXT TTR TO EXT PARM	10420000
*		AREA	10440000
B	SD061070		10460000
SD061160 MVI	TESTEOD,X'FF'	SET EN: OF DATA INDICATOR	10480000

SD061170	LA	R7,0END-2	CHECK FOR INSERTION OF TWO	10500000
	CLK	R2,R1	ZEROS AT END OF BLOCK	10520000
	BH	SD061180		10540000
	XC	0(2,R2),0(R2)	INSERT ZEROS	10560000
SD061180	LA	R3,0C11K	ADDR OF OUTPUT AREA	10580000
	ST	R3,0EXPARM	STORE ADDR IN EXT PARM AREA	10600000
	MVC	QTTR,TTRBUF1	STORE TTR IN EXT PARM AREA	10620000
	MVC	0CTTR,0TTR	CURRENT TTR FIELD IN OUTPUT	10640000
	MVI	QMPDP,X'03'	INDICATE WRITE TO QMGR	10660000
	MVI	QMPCL,X'01'	INDICATE ONE REC TO BE READ	10680000
	LA	R13,REGSAVE	POINTER TO SAVE AREA	10700000
	LR	R1,R9	POINTER TO QMPA	10720000
	L	R15,IEFQMRD		10740000
	BALR	R14,R15	BRANCH TO QMANAGER	10760000
	LTR	R15,R15	ANY NON-ZERO COND CODE	099 10770019
	BZ	OKRET1	NO - CONT PROCESSING	099 10780019
*				099 10790019
BADIO	EQU	*	QMGR I/O ERRORS ENTER HERE	099 10800019
*				099 10810019
*****			I/O ERROR	10860000
	LA	R1,X'0B0'		10880000
	SLA	R1,12		10900000
	ABEND	(1),DUMP		10920000
OKRET1	EQU	*		10940000
	MVC	0LTTR,0CTTR	SETUP 0LTTR FOR NEXT WRITE	10960000
	MVC	TTRBUF1,TTRBUF2	PUT NEXT TTR IN BUFFER 1	10980000
	XC	TTRBUF2,TTRBUF2	CLEAR TTR BUFFER 2	11000000
	CLI	TESTEOD,X'FF'	TEST FOR END OF DATA	11020000
	BE	SD061190		11040000
	LA	R2,0START	ADDR OF 1ST OUTPUT RECORD	11060000
	B	SD061120		11080000
*				11100000
* PREPARE FOR ENTRY TO IEFDSLST				11120000
TBLVCON	DC	V(IEFDSTBL)	TBUILD	11140000
LSTVCON	DC	V(IEFDSLST)	TSTRIP	11160000
MAJNAME	DC	C'SYSOSN '	ENQ MAJ RESOURCE NAME	11180000
*				11200000
SD061190	MVC	CHARSAV(6),0CHAR	S AVE DSNNAME COUNTS	11220000
	MVC	JCTDETD(3),QTTR	STORE TTR FOR USE IN	11240000
*			IEFSD166 DEQ FUNCTION	11260000
	LR	R1,R4	FREE I/O AREAS	11280000
		FREEMAIN R,LV=352,A=(1)		11300000
*				11320000
* SETUP PARAMETERS FOR IEFDSLST AND GET CORE FOR ENQ PARM LIST				11340000
*				11360000
	LA	R3,12(0,0)	DETERMINE CORE NEEDED FOR	11380000
	MH	R3,CNTSAV	ENQ PARM LIST	11400000
	A	R3,CHARSAV	*	11420000
	LA	R3,8(R3)	*	11440000
	LR	R0,R3	LENGTH OF ENQ PARM LIST	11460000
	LA	R2,255(0,0)	SUBPOOL NO. 255	11480000
	SLL	R2,24	PLACE IN HIGH ORDER BYTE	11500000
	OR	R0,R2	OF LENGTH REGISTER	11520000
	ST	R0,LCTPARM2	SAVE FOR USE IN IEFSD102	11540000
	GETMAIN	R,LV=(0)	ENQ PARM LIST	11560000
	ST	R1,DSNBEG	POINTER TO ENQ PARM LIST	11580000
*			AREA FOR IEFDSLST	11600000

MVC	O(8,1),MAJNAME	1ST 8BYTES ARE ENQ MAJ NAME	11620000
S	R3,CHARSAV		11640000
AR	R3,R1		11660000
ST	R3,DSNEND	ADDR OF 1ST DSNNAME ENTRY	11680000
ST	R1,LCTPARM1	POINTER TO ENQ PARM LIST	11700000
		FOR USE IN IEFS0102	11720000
LA	R1,TBLPARM	ADDR OF IEFDLSST PARM LIST	11740000
LA	R13,REGSAVE	POINTER TO SAVE AREA	11760000
L	R15,LSTVCON		11780000
HALR	R14,R15	BRANCH TO IEFDLSST	11800000

			11820000
			11840000
*	UPON RETURN FROM IEFDLSST ROUTINE,THE ENQ PARAMETER LIST HAS BEEN		11860000
*	BUILT FOR USE IN IEFS0102. A POINTER TO IT IS STORED IN		11880000
*	LCTPARM1 AND ITS LENGTH AND SUBPOOL ARE STORED IN LCTPARM2.		11900000
*	IEFDLSST BUILDS THE ENQUEUE PARAMETER LIST TO ISSUE AN	A21364	11903019
*	UNCONDITIONAL ENQUEUE. CODE CHANGES MADE TO IEFS0102 FOR APAR	A21364	11906019
*	21364 REQUIRE THAT A CONDITIONAL ENQUEUE BE ISSUED. THE	A21364	11909019
*	FOLLOWING CODE CHANGES THE ENQUEUE PARMS TO ISSUE A	A21364	11912019
*	CONDITIONAL ENQUEUE.	A21364	11915019
			11920000

			11940000
L	R1,LCTPARM1	ADDRESS Q PARM LIST	A21364 11941019
LA	R1,D8(R1)	ADDRESS 1ST ENTRY	A21364 11942019
USING	QLIST,R1	ENQ LIST BASE	A21364 11943019
SD061193 EQU	*	SETUP FOR COND ENQ	A21364 11944019
OI	QATT,SYSTEM+USE	ENQ SYSTEM, RET = USE	M4008 11945019
CLI	QLSTEND,LSTEND	END OF LIST?	A21364 11946019
BE	SD061196	YES - BRANCH	A21364 11947019
LA	R1,QENTLGTH(R1)	INCREMENT TO NEXT ENTRY	A21364 11948019
R	SD061193	PROCESS NEXT ENTRY	A21364 11949019
SD061196 EQU	*	CONTINUE PROCESSING	A21364 11950019
LA	R1,TREE	FREE TREE	11960000
FREEMAIN	R,LV=8,A=(1)		11980000
			12000000
*			12020000
SD061200 LR	R1,R10	LCT ADDRESS	12020000
L	R15,IEFS062V	ADDRESS OF IEFS0062	12040000
BR	R15	GO TO IEFS0062	12060000
IEFS060V DC	V(IEFS060)		12070015
IEFS062V DC	V(IEFS062)		12080000
IEFQMRD DC	V(IEFQMRW)		12100000
IEFQMVCON DC	V(IEFQMDQ2)		12120000
DSOVCON DC	V(IEFDLSL)	DSO SELECT DSOCH ROUTINE	099 12130019

			12140000
QIOBA EQU	8		12160000
QIOBB EQU	20		12180000
QIOBDCB EQU	24	DCB ADDR OFFSET FROM ECB/IOB	12200000
QIOBC EQU	44		12220000
QIOBD EQU	52		12240000
QMCOP EQU	X'31'		12260000
DISPLACE EQU	760		
ZERO EQU	0	DISPLACEMENT	1272 12260619
ONE EQU	1	DISPLACEMENT	1272 12261219
D3 EQU	3	DISPLACEMENT	1254 12261819
D4 EQU	4	DISPLACEMENT	1272 12262419
D6 EQU	6	DISPLACEMENT	1272 12263019
D7 EQU	7	DISPLACEMENT	1272 12263619

DR	EQU	8	DISPLACEMENT	I272	12264219
DO	EQU	0		A27668	12264319
D1	EQU	1		A27668	12264419
D16	EQU	16	DISPLACEMENT	M4339	12264519
WTPLEN	EQU	16	WTPCR LENGTH	I254	12264819
BLANK	EQU	X'40'	USED TO PAD WITH BLANKS	I272	12265419
FOXFOX	EQU	X'FF'	INSERT UNIQUE NONZERO ID	I272	12266019
IOFRR	EQU	4	I/O ERROR RETURN CODE	I272	12266619
	DS	OF		I254	12267219
M253	DC	X'FD000000'	MASK FOR SP253	I254	12267819
HOUR24	DC	AL3(8640000)		I272	12268419
	EJECT				12270017
	XNAP	QMECH,RI,QMECH,,,,MF=L	ECB/IOB BASIC PATTERN		12280000
QMPTF	EQU	*	END OF PATTERN IOB		12300000
QIOBL	EQU	QMPTF-QMECH	LENGTH OF PATTERN IOB		12320000
QMNCVCON	DC	V(IEFQMIJNC)			12340000
POSTBIT	EQU	X'40'			12360000
TCORE	EQU	253			12380000
	EJECT				12390017
*****					12390919
*	ENQ/DEQ MACRO PARAMETER LIST			*	12391819
*****					12392719
IEFSD61A	ENQ	(IEFSQCHS,IEFQ6QCB,E,2,SYSTEM),MF=L	ENQ/DEQ PARM LIST		12393619
IEFSQCHS	DC	C'SYSIEFSD'	MAJOR NAME		12394519
IEFQ6QCB	DC	C'Q6'	MINOR NAME STOP INIT.		12395419
IEFSD61B	ENQ	(IEFSQCHS,IEFQ2QCB,E,2,SYSTEM),MF=L	ENQ/DEQ PARM LIST		12396319
IEFQ2QCB	DC	C'Q2'	MINOR NAME ASGN-DEL Q MNGR		12397219
IEFSD61C	ENQ	(IEFSQCHS,IEFRATRM,E,5,SYSTEM),MF=L			12398119
IEFSQCHS	DC	C'SYSJMPCH'			12399019
IEFRATRM	DC	C'BATRM'			12399919
STOPBIT	EQU	X'10'			12400819
MSGAD	DC	A(QMTMSG)	ADDR OF MESSAGE CSECT	0847	12401719
*****				A21364	12401819
QLIST	DSECT		ENQ, DEQ PARAMETER LIST	A21364	12401919
*			ANY NUMBER OF ENTRIES	A21364	12402019
*			EACH ENTRY IS DESCRIBED	A21364	12402119
*			AS FOLLOWS:	A21364	12402219
QLSTEND	DC	X'00'	END OF LIST INDICATOR	A21364	12402319
LSTEND	EQU	X'FF'	FF INDICATES END OF LIST	A21364	12402419
QMINLGTH	DC	X'00'	MINOR NAME LENGTH	A21364	12402519
QATT	DC	B'00000000'	ENQ OR DEQ ATTRIBUTES	A21364	12402619
SYSTEM	EQU	X'40'	SYSTEM IS ISSUING NO(DQ)	A21364	12402719
HAVE	EQU	X'01'	RET = HAVE	A21364	12402819
USE	EQU	X'03'	RET = USE	A21364	12402919
QRETCDE	DC	X'00'	RETURN CODE	A21364	12403019
QMAJOR	DC	A(*)	ADDRESS OF MAJOR NAME	A21364	12403119
QMINOR	DC	A(*)	ADDRESS OF MINOR NAME	A21364	12403219
QENTLGTH	EQU	*-QLSTEND	LENGTH OF ONE ENTRY	A21364	12403319
*****				A21364	12403419
QMTMSG	CSECT			0847	12403519
	DS	OD		0847	12404419
	PRINT	DATA		A27668	12404519
WTO429	WTO	'IEF429I INITIATOR '''' WAITING FOR WORK',MF=L,		A27668*	12404619
		ROUTCDE=(2),DESC=(4)		A27668	12404719
	PRINT	NODATA		A27668	12404819
WTO429EN	EQU	*-WTO429	LENGTH OF ENTIRE WTO	A27668	12404919
WTO429EX	EQU	WTO429EN+08	SIZE OF WTO AND INIT ID	A27668	12405019

```

*
*   N O T E - IF WTD 'WTD429' CHANGES, CHECK EQU 'WTD429A' TO
*               ASSURE ITS VALUE IS STILL CORRECT
*
WTD429A EQU    23  NUMBER OF BYTES IN 1ST HALF OF WTD 'WTD429'
*               UP TO AND INCLUDING THE 1ST QUOTE OF
*               GENERATED TEXT. INCLUDE 4 BYTE HEADER IN CT
*
WTD429B EQU    WTD429EN-WTD429A      LNG OF 2ND HALF OF WTD MSG
*               STARTING WITH 2ND QUOTE
*               INCLUDING MCS DATA.
*****
*
*   EJECT
IEFDSOSL CSECT
*
* * * * * * * * THIS DIRECT SYSOUT (DSO) ROUTINE FIRST
*               BUILDS A SYSOUT ENTRY BLOCK OF ALL SYSOUT
*               CLASSES THIS JOB HAS CALLED FOR IN ITS JCL
*               FROM THE SYSOUT CLASS DIRECTORY (SCD). THIS
*               BLOCK WILL BE SEARCHED LATER IN THE ROUTINE.
*               THE NEXT FUNCTION IS TO SEARCH THE DIRECT
*               SYSOUT CONTROL BLOCK (DSOCB) CHAIN TESTING
*               EACH DSOCB TO SEE IF IT'S ELIGIBLE FOR THIS
*               JOB. THE DEQUEUED JOB'S JOB CLASS AND THE
*               SYSOUT ENTRY BLOCK WILL BE USED TO SELECT
*               THE DSOCB'S. SELECTION IS INDICATED BY
*               PLACING THE PROBLEM PROGRAMS STORAGE PROTECT
*               KEY FROM THE JOBS JCT (JCTJSRNO) INTO THE
*               SELECTED DSOCB. IF ONE OR MORE DSOCBS WERE
*               SELECTED BIT 6 IN JCTRSW2 (JCT) WILL BE TURNED ON TO
*               INDICATE TO OTHER DSO ROUTINES THAT THERE ARE
*               ALLOCATED DSOCB'S FOR THE JOB.
*
*               REGISTER ONE POINTS TO LOT/LCT AREA
*               LCTPARM1 CONTAINS ADR OF QMGR READ/WRITE ROUTINE
*               EPA POINTED TO BY THE QMPCL IN THE INPUT QMPA HAS
*               BEEN INITIALIZED WITH THE CORE ADR OF THE READ IN SCD.
*
*   SAVE (14,12)      SAVE CALLERS REGS
*
*   LR   R11,R1        ADR OF LCT/LOT AREA
*   USING IEFLDT,R11
*
*   BALR R12,R0        SET   MODULE BASE REG
*   USING *,R12
*   B     DSOGD
*   DC    X'CCCC'
*   DC    C'IEFDSOSL'
*   DC    X'06201969'
*   DC    X'CCCC'
DSOGD   DS     OH
*
*   LA    R0,SAVARASZ+SYSIZE  SIZE OF REG SAVEAREA
*                               AND SYSOUT BLOCK AREA
*                               FOR GETMAIN
*   GETMAIN R,LV=(0)        GET REG SAVE AND SYSOUT BLOCK AREA

```

```

A27668 12405119
A27668 12405219
A27668 12405319
A27668 12405419
A27668 12405519
A27668 12405619
A27668 12405719
A27668 12405819
A27668 12405919
A27668 12406019
A27668 12406119
0847 12407119
12408019
099 12408919
099 12409819
099 12410719
099 12411619
099 12412519
099 12413419
099 12414319
099 12415219
099 12416119
099 12417019
099 12417919
099 12418819
099 12419719
099 12420619
099 12421519
099 12422419
099 12423319
099 12424219
099 12425119
099 12426019
099 12426919
099 12427819
099 12428719
099 12429619
099 12430519
099 12431419
099 12432319
099 12433219
099 12434119
099 12435019
099 12435919
099 12436819
099 12437719
099 12438619
099 12439519
099 12440419
099 12441319
099 12442219
099 12443119
099 12444019
099 12444919
099 12445819
099 12446719
099 12447619

```

```

*
XC      NO(SAVARASZ+SYSIZE,R1),NO(R1)  CLEAR ENTIRE AREA      099 12448519
ST      R13,N4(,R1)      SAVE CALLERS REG SAVEAREA ADR      099 12449419
ST      R1,N8(,R13)      MY REG SAVEAREA ADR INTO CALLERS    099 12450319
*
*      REG SAVEAREA      099 12451212
LR      R13,R1      MY REG SAVEAREA AND SYSOUT BLOCK AREA    099 12452119
USING   SAVESYS,R13    SET BASE      099 12453019
*
*      099 12453919
L        R9,LCTJCTAD      JCT CORE ADR      099 12454819
USING   IEFJCT,R9        SET BASE      099 12455719
*
*      099 12456619
LA      R10,QMGR1      ADR OF INPUT QMPA      099 12457519
USING   IOPARAMS,R10    SET BASE      099 12458419
*
*      099 12459319
L        R7,QMPCL QMGR EXTERNAL PARM ADR      099 12460219
L        R7,NO(,R7) ADR OF SCD AREA USED BY CALLING PROG    099 12461119
USING   SCDAREA,R7      SET BASE FOR SCD      099 12462019
MVI     SCDPAD,HEXFF    CONTINUATION FLAG IN 1ST SCD        099 12462919
*      AFTER PROBABLE 24TH ENTRY      099 12463819
*      DROP R7      099 12464719
*      099 12465619
*      099 12466519
* * * * * BUILD SYSOUT ENTRY BLOCK FROM SCD      099 12467419
*      099 12468319
LR      R2,R7      INDEX REG FOR SCD      099 12469219
USING   SCDAREA,R2    SET SCD BASE      099 12470119
LA      R4,SYSCLS    SYSOUT ENTRY BLOCK AREA      099 12471019
USING   SYSCLS,R4     SET SYSOUT BLOCK BASE      099 12471919
LA      R3,SYSNO      MAX ENTRIES LOOP REG      099 12472819
*      099 12473719
DS02A   EQU      *      099 12474619
TM      SCDCLS,HEXFF  TEST FOR END OF ENTRIES/END OF      099 12475519
*      FIRST SCD/SYSOUT CLASS ENTRY      099 12476419
RD      DS02C      ALL ONES - END OF FIRST SCD, READ IN      099 12477319
*      NEXT SCD IF ONE AND CONTINUE      099 12478219
BZ      DS03      ALL ZERO - END OF ENTRIES IN SCD      099 12479119
*      099 12480019
*      MIXED - SAVE SYSOUT CLASS      099 12480919
*      099 12481819
MVC     SYSCLS,SCDCLS  SAVE SCD SYSOUT CLASS      099 12482719
*      099 12483619
LA      R2,SCDENTRY(,R2)  INCREASE SCD BASE REG      099 12484519
LA      R4,SYSENTY(,R4)  INCREASE SYSOUT BLOCK BASE      099 12485419
BCT     R3,DS02A      LOOP LIMITER      099 12486319
B       DS03      END OF ENTRIES IN SCD      099 12487219
DROP    R2,R4      099 12488119
*      099 12489019
*      099 12489919
* * * * * READ IN NEXT SCD IF ONE      099 12490819
*      099 12491719
DS02C   EQU      *      099 12492619
USING   SCDAREA,R7      SET SCD BASE      099 12493519
MVI     SCDNXT+N3,NO    ZERO L/O OF TTR BEFORE TEST      099 12494419
L        R2,SCDNXT      GET TTR OF NEXT SCD      099 12495319
DROP    R7      099 12496219
LTR     R2,R2      IS THERE A TTR      099 12497119
BZ      DS03      NO - END OF ENTRIES IN SCD      099 12498019
*      099 12498919

```

```

*                               YES - READ IN NEXT SCD                               099 12499819
L      R1,QMPCL  QMGR EXTERNAL PARM ADR                                         099 12500719
ST     R2,N4(,R1) TTR OF SCD TO RE READ                                         099 12501619
ST     R7,N0(,R1) READ IN AREA FOR NEXT SCD (SAME AS                               099 12502519
*                               1ST SCD)                                         099 12503419
MVI    QMPOP,QMREAD  READ FUNCTION                                             099 12504319
BAL    R6,QMGRRW      GO TO QMGR READ WRITE ROUTINE                             099 12505219
LR     R2,R7          SCD AREA                                                  099 12506119
R      DSO2A          CONTINUE PROCESSING WITH THIS SCD                         099 12507019
QMGRRW EQU *          QMGR READ WRITE INTERFACE                               099 12507919
MVI    QMPCL,HEX01    READ/WRITE ONE RECORD                                    099 12508819
LR     R1,R10         ADR OF QMGR1                                              099 12509719
*                               R13 POINTS TO SAVEAREA                             099 12510619
L      R15,LCTPARM1    ADR OF QMGR READ/WRITE ROUTINE                          099 12511519
BALR   R14,R15        GO TO QMGR                                              099 12512419
*                               099 12513319
LTR    R15,R15        QMGR ERROR TEST                                           099 12514219
BNZ    QMGR10         QMGR I/O ERROR                                           099 12515119
BR     R6             RETURN TO CALLER                                           099 12516019
*                               099 12516919
*                               099 12517819
*                               099 12518719
*                               099 12519619
DS03   EQU *          099 12520519
LA     R1,SYSD0        MAX ENTRIES IN SYSOUT BLOCK                             099 12521419
SR     R1,R3          LOOP COUNT FROM MAX ENTRIES                             099 12522319
*                               099 12523219
*                               GIVES NUMBER OF SYSOUT CLASSES AVAILABLE         099 12524119
STC    R1,SYSENTS     SAVE NUMBER OF SYSOUT CLASSES                           099 12525019
*                               099 12525919
* * * * * LOOP THROUGH DSOB'S USING THE DEQUEUED JOB'S                         099 12526819
*                               JOB CLASS AS THE FIRST TEST IN SELECTING DSOB'S 099 12527719
*                               099 12528619
L      R7,CVTPTR      LOCATION OF CVT                                           099 12529519
USING  CVTDSECT,R7    CVT BASE                                                  099 12530419
L      R7,CVTMSER      ADR OF MASTER SCH DATA AREA                           099 12531319
USING  BASE,R7        M/S BASE REG                                              099 12532219
*                               099 12533119
ENQ    MF=(E,DSOGBNQ) ENQ ON DSOB CHAIN                                         099 12534019
*                               099 12534919
L      R7,BADSDO FIRST DSOB ADR                                                  099 12535819
USING  IEFDSOGB,R7    SET BASE FOR DSOB                                         099 12536719
*                               099 12537619
LA     R4,L'DSOJCLS    LENGTH OF 'DSOJCLS' FIELD IN DSOB.                     099 12538519
*                               REG WILL ALSO BE USED AS A SWITCH                 099 12539419
*                               099 12540319
DS03B EQU *          099 12541219
LA     R7,N0(,R7)      CLEAR H/O                                               099 12542119
LTR    R7,R7          DSOB ADR ZERO                                             099 12543019
BZ     DSO5           YES - DO END OF DSOB PROCESSING                           099 12543919
CLI    DSOID,HEXFD     NO - CHECK FOR DSOB ID                                  099 12544819
RNE    DSO3A          WRONG ID, GET NEXT DSOB                                   099 12545719
TM     DSOKEY,HEXFF     IS THIS DSOB IN USE                                     099 12546619
BNZ    DSO3A          YES - GET NEXT DSOB                                        099 12547519
TM     DSOIND1,DSOINDP NO - IS DSOB FROZEN                                       099 12548419
BNZ    DSO3A          YES - GET NEXT DSOB                                        099 12549319
TM     DSOIND2,DSOSM    IS A STOP/MODIFY PENDING                               099 12550219
BNZ    DSO3A          YES - THEN BYPASS THIS DSOB

```

```

*                               NO - SET REGS TO EXAMINE DSOCH 099 12551119
*                               JOB CLASSES 099 12552019
*                               ELIGIBLE JOB CLASSES AREA 099 12552919
*                               SET BASE 099 12553819
*                               LOOP LIMIT - R4 WILL BE NEG IF 099 12554719
*                               ANY DSOCH'S HAVE BEEN SELECTED, 099 12555619
*                               BUT VALUE REMAINS UNCHANGED 099 12556519
*                               099 12557419
DS03C EQU * 099 12558319
      CLI DSOJCLS,NO END OF ENTRIES TEST 099 12559219
      RF DSO3A YES - GET NEXT DSOCH 099 12560119
      CLC QMTPY,DSOJCLS JOB'S JOB CLASS VS DSOCH JOB CLASSES 099 12561019
      RF DSO4 A MATCH - GO TEST SYSOUT CLASS 099 12561919
*                               099 12562819
*                               LA R2,N1(,R2) INCREASE REG BY ONE 099 12563719
*                               RCT R3,DS03C LOOP AGAIN IF MORE ENTRIES 099 12564619
*                               099 12565519
*                               DROP R2 099 12566419
*                               099 12567319
DS03A EQU * 099 12568219
      L R7,DS0NXT GET NEXT DSOCH ADR 099 12569119
      R DSO3H TEST JOB CLASSES IN NEXT DSOCH 099 12570019
*                               099 12570919
*                               099 12571819
* * * * * A JOB CLASS MATCH HAS OCCURRED SO NOW MATCH THE 099 12572719
*                               DSOCH'S SYSOUT CLASS TO THE SYSOUT BLOCK ENTRIES 099 12573619
*                               099 12574519
DS04 EQU * 099 12575419
      LA R2,SYSCLS START OF SYSOUT BLOCK TO BE SCANNED 099 12576319
      USING SYSCLS,R2 SET BASE 099 12577219
      SR R3,R3 CLEAR REG 099 12578119
      IC R3,SYSENTS NUMBER OF ENTRIES IN SYSOUT BLOCK 099 12579019
*                               099 12579919
DS04A EQU * 099 12580819
      CLC DSOCLS,SYSCLS TEST SYSOUT CLASS 099 12581719
      RE DSO4B EQUAL - MARK DSOCH ALLOCATED 099 12582619
      LA R2,SYSENTY(,R2) UNEQUAL - INCREASE INDEX REG 099 12583519
      RCT R3,DS04A LOOP AGAIN IF MORE ENTRIES 099 12584419
      R DSO3A END OF ENTRIES - GET NEXT DSOCH 099 12585319
*                               099 12586219
DS0FX MVC SYSCLS(NO),SYSCLS+SYSENTY TO SHIFT SYSOUT 099 12587119
*                               BLOCK UP 1 BYTE 099 12588019
*                               MOVE INCLUDES ONE BYTE 099 12588919
*                               PAD AT END OF BLOCK 099 12589819
*                               099 12590719
*                               DROP R2 099 12591619
*                               099 12592519
* * * * * JOB CLASS AND SYSOUT CLASS MATCH HAS OCCURRED 099 12593419
*                               SO MARK DSOCH ALLOCATED FOR THIS JOB AND 099 12594319
*                               STRIKE THE SYSOUT ENTRY FROM THE SYSOUT 099 12595219
*                               BLOCK TO PREVENT SELECTION OF ANOTHER POSSIBLE 099 12596119
*                               DSOCH OF SAME JOB AND SYSOUT CLASS. 099 12597019
*                               099 12597919
DS04B EQU * 099 12598819
      MVC DSOKEY,JCTJSRND MOVE PROTECT KEY FROM JCT 099 12599719
*                               INTO THE DSOCH TO MARK 099 12600619
*                               IT ALLOCATED (XXXX0000) 099 12601519
*                               LNR R4,R4 NEG REG INDICATES ONE OR MORE

```

```

*                               DSOCH'S HAVE BEEN SELECTED
* CLC  DSOSCLS,JCTJMGPD      IS THE MSGCLASS EQUAL TO
*                               THE DSOCH'S SYSOUT CLASS
* HNE  DS04C                  NO - CONT PROCESSING
* OI   DS0IND1,DSOMSGCL YESTURN ON MSGCLASS BIT IN DSOCH
*
* * * * * * * * * * STRIKE SYSOUT CLASS FROM SYSOUT BLOCK
*
DS04C  EQU  *
      LA   R3,SYSCLS+(SYSNO*SYSENTRY)-1  END OF SYSOUT BLOCK
      SR   R3,R2                        TO GET NUMBER OF BYTES TO BE MOVED
*                                           NOTE - A LOW ORDER VALUE ONLY NOW
*                                           EXISTS IN R3
* EX     R3,DSOEX                  SHIFT SYSOUT BLOCK ENTRIES
*                                           LEFT ONE BYTE PLUS PAD BYTE
* IC     R3,SYSENTS                PRESENT NUMBER OF ENTRIES IN BLOCK
      RCTR R3,R0                     SUBTRACT ONE
      STC  R3,SYSENTS              STORE REVISED COUNT
      LTR  R3,R3                     IS COUNT ZERO
      BZ   DS05                     YES - DO END OF DSOCH PROCESSING
      B    DS03A                    NO - GO GET NEXT DSOCH
*
QMGRID EQU  *
      LR   R7,R15                   SAVE CONDITION CODE FROM QMGR
      B    I,05B                     DO END OF MODULE PROCESSING
*
* * * * * * * * * * ALL DSOCH'S HAVE BEEN SCANNED AND TESTED,
* OR ALL SYSOUT ENTRIES IN THE SYSOUT BLOCK
* HAVE BEEN PROCESSED.
* NOW TEST IF ANY DSOCH'S WERE SELECTED.
* IF YES MARK THE JCT TO INDICATE THIS,
* RELEASE SAVE AND WORK AREAS AND RETURN
* TO CALLER.
*
DS05   EQU  *
      LTR  R4,R4                     TEST IF ANY DSOCH'S WERE SELECTED
      BP   DS05A                     PLUS - NO DSOCH'S SELECTED
      OI   JCTRSW2,JCTDS0JB          NEG- ONE OR MORE DSOCH'S WERE
*                                     SELECTED
*
DS05A  EQU  *
      SR   R7,R7                     CLEAR TEMP CONDITION CODE REG
*
      DEQ  MF=(E,DSOCHNO)            DEQ ON DSOCH CHAIN
*
DS05B  EQU  *                       ENTER HERE IF PASSING CONDITION CODES
      LR   R1,R13                     AREA TO BE RELEASED
      L    R13,SVEARA+N4              GET CALLER'S SAVEAREA ADR
      XC   SVEARA+N8(N4),SVEARA+N8    CLEAR MY SAVEAREA
*                                     ADR IN CALLERS
*                                     SAVEAREA
*
      DROP R13
*
      LA   R0,SAVARASZ+SYSSIZE        SIZE OF AREA TO BE RELEASED
      FREEMAIN R,LV=(0),A=(1)
*
      LR   R15,R7                     LOAD ANY CONDITION CODES

```

099 12602419
099 12603319
099 12604219
099 12605119
099 12606019
099 12606919
099 12607819
099 12608719
099 12609619
099 12610519
099 12611419
099 12612319
099 12613219
099 12614119
099 12615019
099 12615919
099 12616819
099 12617719
099 12618619
099 12619519
099 12620419
099 12621319
099 12622219
099 12623119
099 12624019
099 12624919
099 12625819
099 12626719
099 12627619
099 12628519
099 12629419
099 12630319
099 12631219
099 12632119
099 12633019
099 12633919
099 12634819
099 12635719
099 12636619
099 12637519
099 12638419
099 12639319
099 12640219
099 12641119
099 12642019
099 12642919
099 12643819
099 12644719
099 12645619
099 12646519
099 12647419
099 12648319
099 12649219
099 12650119
099 12651019
099 12651919
099 12652819

```

      RETURN (14,12),RC=(15)      RETURN TO CALLER
*
* * * * * EQUATES
*
N0      EQU      0
N1      EQU      1
N2      EQU      2
N3      EQU      3
N4      EQU      4
N7      EQU      7
NR      EQU      8
HEXFF   EQU      X'FF'
HEX01   EQU      X'01'
HEXFD   EQU      X'FD'
*
* * * * * ENQ/DEQ FOR DIRECT SYSOUT CONTROL BLOCKS (DSOCH)
*
DSOCHNO  ENQ      (DSOCHNOQ,DSOCHNR,E,05,SYSTEM),MF=L
*
* * * ENQ/DEQ Q AND R NAMES
*
DSOCHNOQ DC      CL8'SYSIFFSD'  Q - MAJOR NAME
DSOCHNR  DC      C'DSOCH'      R - MINOR NAME
*
* * * * * SYSOUT CLASS DIRECTORY (SCD)
*
SCDAREA  DSECT
SCDCLS   DS      CL1  SYSOUT CLASS
          DS      CL6  Q MGR DATA FOR SYSOUT Q
SCDENTRY EQU      *-SCDCLS  SIZE OF EACH SYSOUT ENTRY
          DS      CL(SCDENTRY*23)  24 ENTRIES IN SCD
SCDPAD   DS      CL4  PAD
SCDNXT   DS      F    TTR OF NEXT SCD
*
* * * * * SYSOUT CLASS BLOCK
*
SAVESYS  DSECT
SVFEARA  DS      CL72      REG SAVEAREA
SAVARASZ EQU      *-SVFEARA  SIZE OF SAVE AREA
SYSENTS  DS      CL1  NUMBER OF ENTRIES ENTERED IN BLOCK
SYSCLS   DS      CL1  SYSOUT CLASS
SYSENTRY EQU      *-SYSCLS  SIZE OF EACH SYSOUT ENTRY
SYSNO    EQU      36  MAX NUMBER OF ENTRIES
*
          DS      CL(SYSENTRY*(SYSNO-1))  FINISH OUT BLOCK
          DS      CL1  PAD FOR 'MOVE' TO STRIKE SYSOUT ENTRIES
          FROM THE SYSOUT BLOCK.  HEX'00'
*
SYSIZE   EQU      *-SYSENTS  TOTAL SIZE OF SYSOUT BLOCK
*
*
IEFDSOCH DSECT
IEFDSOCH
EJECT
*
RSAR     DSECT
SAVEAREA DS      18F
REMLST   DS      2F
THIS DSECT IS USED TO MAP ALLOCATE/
TERMINATE REG SAVE AREAS FOR USE BY
THIS MODULE

```

```

099 12653719
099 12654619
099 12655519
099 12656419
099 12657319
099 12658219
099 12659119
099 12660019
099 12660919
099 12661819
099 12662719
099 12663619
099 12664519
099 12665419
099 12666319
099 12667219
099 12668119
099 12669019
099 12669919
099 12670819
099 12671719
099 12672619
099 12673519
099 12674419
099 12675319
099 12676219
099 12677119
099 12678019
099 12678919
099 12679819
099 12680719
099 12681619
099 12682519
099 12683419
099 12684319
099 12685219
099 12686119
099 12687019
099 12687919
099 12688819
099 12689719
099 12690619
099 12691519
099 12692419
099 12693319
099 12694219
099 12695119
099 12696019
099 12696919
099 12697819
099 12698719
099 12699619
099 12700519
099 12701419
099 12710014
099 12720014
099 12730014

```

NOWRKECH DS	2F		12740014
ECBLST DS	2F		12750014
WTOSWCH DS	C	SWCH ON SAYS WRITE 'INPUT QUEUE EMPTY' MESSAGE TO CONSOLE	12760014 12770014 12780014
* IEEBASEA			
EJECT			12870017
CVTDSECT DSECT			12880000
CVT			12900000
EJECT			12920000
IEFCSCB DSECT			12940000
IEECHAIN			12960000
EJECT			12980000
IEFJCT DSECT			13000000
IEFAJCTB			13020000
IEFSCT DSECT			13026017
IEFASCTB			13032017
* JCYDETD EQU INJMCT+88 1ST TTR IN END OSNAME TABLE			13040000
EJECT			13060000
*****			13080000
*****			13100000
* THE FOLLOWING REPRESENTS A BLOCK OF MAIN STORAGE WHICH IS			* 13120000
* ALLOCATED FOR THE LIFE OF			* 13140000
* OF THIS INITIATOR. IT CONTAINS:			* 13160000
* (1) THE LCT (LINKAGE CONTROL TABLE)			* 13180000
* (2) A 2 LEVEL REGISTER SAVE AREA FOLLOWED BY THE			* 13200000
* THE QUEUE MANAGER PARAMETER AREA			* 13220000
* (3) AN ALTERNATE QUEUE MANAGER PARAMETER AREA			* 13240000
* *****			* 13260000
*****			* 13280000
*****			* 13300000
*****			13320000
IEFLDT DSECT			13340000
IEFALLCT			13360000
LCTBATMN DS	F	*****TO BE ADDED TO LCT	13380000
LCTSDQMP DS	F	*****TO BE ADDED TO LCT(VMS ONLY) POINTER TO SYSOUT QUEUE MANAGER PARAMETER AREA	13400000 13420000
LCTSRG DS	F	*****TO BE ADDED TO LCT(MVT ONLY) - PTR TO STOP INIT	13440014
* ECB			13460014
LCTINTSW DS	0C	INITIATOR INTERNAL SWITCHES	1272 13462019
LCTIHIER EQU	128	RUN IN HIERARCHY ONE	1272 13464019
LCTSDOXX EQU	32	ATTACH IEFSDOXX	1272 13466019
LCTMINRG EQU	16	JOB FLUSH - USE MINPAR	1272 13468019
LCTSTART EQU	8	TASKNAME NOT ON COMMAND	1272 13470019
LCTSTOP EQU	4	INITIATOR INTERNAL STOP	1272 13472019
LCTABEND EQU	2	EXECUTED PROGRAM ABENDED	1272 13474019
LCTSLP DS	F	*****TO BE ADDED TO LCT(MVT ONLY) - PTR TO START PAR-	13480014
* AMETER LIST			13500014
LCTTMWRK DS	4F	*****TO BE ADDED TO LCT(VMS ONLY) TIMER WORK AREA	13520000
LCTJOBLEB DS	F	*****TO BE ADDED TO LCT(VMS ONLY) JOBLIB DCB PTR	13540000
LCTATLST DS	F	*****TO BE ADDED TO LCT(VMS ONLY) POINTER TO ALLOCATE/TERMINATE PARAMETER LISTS	13560000 13580000
REGSAVE DS	36F	ALLOCATE/TERMINATE REGSAVE AREA	13600000
QMGR1 DS	9F	QUEUE MANAGER PARAMETER AREA	13620000
QMGR2 DS	9F	ALTERNATE QUEUE MANAGER AREA	13640000
TRSTKINF DS	2F	NEEDED FOR TRACK STACKING AND QUEUE BREAK INFORMATION. 1ST BYTE TO CON-	13660000 13680000
*			

*			TAIN NUMBER OF BUFFERS, NEXT 3 BYTES	13700000
*			FOR STACK ADDR, NEXT 4 BYTES FOR	13720000
*			QUEUE BREAK INFORMATION	13740000
QMWTO1	EQU	QMR1+24	PTR IN QMPA TO TRK STACK INFO	13760000
QMWTO2	EQU	QMR2+24	PTR IN QMPA TO TRK STACK INFO	13780000
ECHLIST	DS	1F		13786015
ICTIDENT	DC	2F'0'	HOLDER FOR IDENTIFIER	13792015
LCTFORCF	DS	CL8		1241 13793018
LCTLIMIT	DS	C		1241 13794018
PROPRTY	DS	C		1241 13795018
INITPRTY	DS	C	INITIATOR'S PRTY	1241 13796018
*	THESE FIELDS ARE NEEDED FOR L-SHAPE/INIT MERGE		*	1272 13796219
	DS	OF		1272 13796419
LCTOPSW1	DS	OC	INITIATOR OPTIONS BYTE 1	1272 13796619
LCTPKEYF	EQU	128	DONT GET PROTECT KEY	1272 13796819
LCTQWFF	EQU	64	DONT PROCESS DEDICATED WORKF	1272 13797019
LCTSTMODF	EQU	32	DONT PROCESS STOP/MODIFY	1272 13797219
LCTMINPF	EQU	16	GET REGION SIZE SPECIFIED	1272 13797419
LCTCANF	EQU	8	ALLOW CANCEL ONLY AT ALLOC	1272 13797619
LCTONEJF	EQU	4	PROCESS ONLY ONE JOB	1272 13797819
LCTICMODF	EQU	2	DONT PROCESS INITIATOR CMDS	1272 13798019
LCTEXIT	DS	F	ADDRESS INITIATOR EXIT LIST	1272 13798219
LCTOPSW2	DS	OC	INITIATOR OPTIONS BYTE 2	1272 13798419
LCTTIMEF	EQU	128	DONT TIME THIS JOB	1272 13798619
LCTCRF	EQU	64	DONT ALLOW CHECKPT/RESTART	1272 13798819
LCTDSOF	EQU	32	DONT PROCESS DSN	1272 13799019
LCTCOM	DS	F	COMMUNICATIONS PARM AREA PTR	1272 13799219
LCTJSCB	DS	F	ADDRESS OF JSCB FOR TASK	1254 13799419
IEFEND	EQU	*	END OF LIFE-OF-TASK BLOCK	13800000
	EJECT			13820000
QMRDSECT	DS	OF		13840000
	IEFQMRRES			13860000
	IEFQMRNGR			13880000
QMDSC	EQU	QMTID+2		CR17 13890017
CANCELED	EQU	X'01'	SET BY ALTER ROUTINE FOR IN-Q JOB CAN35L	13900000
	EJECT			13910017
DSNIN	DSECT			13920000
	DS	OD		13940000
*	DSNAME ENQ TABLE INPUT AREA			13960000
ICTTR	DS	CL3	CURRENT TTR	13980000
	DS	CL1	TABLE ID	14000000
ILTTR	DS	CL3	LAST TTR	14020000
	DS	CL1	ALIGNMENT ZERO	14040000
ICHAR	DS	F	ACCUM. DSNAME CHAR COUNT	14060000
ICNT	DS	H	ACCUM. DSNAME COUNT	14080000
	DS	CL162	DSNAME RECORDS	14100000
*				14120000
*	DSNAME DEQ TABLE OUTPUT AREA			14140000
OCTTR	DS	CL3	CURRENT TTR	14160000
OTABID	DS	CL1	TABLE ID	14180000
OLTTR	DS	CL3	LAST TTR	14200000
	DS	CL1		14220000
OCHAR	DS	F	ACCUM. DSNAME CHAR COUNT	14240000
OCNT	DS	H	ACCUM. DSNAME COUNT	14260000
OSTART	DS	CL162	DSNAME RECORDS	14280000
OEND	EQU	*		14300000
*				14320000

```

* TWO 4 BYTE AREAS FOR TBUILD ROUTINE
TREE      DS      00
          DS      F
          DS      00
RUD        DS      F
*
IEFLOT     DSECT
          ORG     REGSAVE+72
* EXTERNAL PARAMETER AREA FOR QMGR
QEXPARM    DS      F
QTTR       DS      CL3
          DS      CL1
*
* WORK AREA
*
LENGTH     DS      H
TTRBUF1    DS      CL3
TTRBUF2    DS      CL3
TESTEOD    DS      CL1
CHARSAV    DS      F
CNTSAV     DS      H
*
* PARAMETER LIST FOR TBUILD AND TSTRIP ROUTINES
*
TBLPARAM   DS      F
DSNBEG     DS      F
DSNEND     DS      F
IEL        DSECT
          IEZIEL  EXITS=RTN
COMPA      DSECT
          IEZCOM
          IEZJSCB
CIBDSECT   DSECT
          IEZCIB
          IEFWPCB
          END
CORE ADDR FOR READ OR WRITE
TTR
CURRENT DSNAME LENGTH
BUFFERS TO HOLD TTR S FOR
Q MGR WRITE FUNCTION
END OF DATA INDICATOR-X'FF'
DSNAME COUNT AND CHARACTER
SAVE AREAS
ADDR OF TREE-B
2ND PARM LIST ENTRY
1ST PARM LIST ENTRY

```

14340000
14360000
14380000
14400000
14420000
14440000
14460000
144800 0
14500000
14520000
14540000
14560000
14580000
14600000
14620000
14640000
14660000
14680000
14700000
14720000
14740000
14760000
14780000
14800000
14820000
14840000
14860000
14862019
14864019
14866019
14868019
14870019
1272 14872019
1272 14874019
M4339 14877019
14880000

```

      TITLE 'IEFSD104-MS/1 STEP DELETE INTERFACE'
      IEFSD104 CSECT
*
*
*011200-011800,016200-016400,021400-021600
*002600,004400,005600,006400,010800-015800,028600
*
      BALR R12,0
*
*****
*
      MS/1 STEP DELETE INTERFACE
*
*****
* STATUS - CHANGE LEVEL 000
*
* FUNCTION (1) READ LCT FROM JOB QUEUE FOR SYSTEM TASKS
* ENTRY POINT: IEFSD104 FROM IEFSD263.
* INPUT: REGISTER ONE POINTING TO PARAMETER LIST CONSISTING OF:
*         (1) USER PARAMS POINTER
*         (2) FCB LIST POINTER
*         (3) ATTACH L LIST POINTER
*         (4) LCT POINTER
*         (5) SYSTEM TASKS - POINTER TO PARM AREA
*         (6) TIOT LIST POINTER
*         (7) DCH ADDRESS
* OUTPUT - THE INPUT PARAMETER LIST CONTAINING AN UPDATED TIOT LIST
*          POINTER
* EXTERNAL REFERENCES - IEFSD514
* EXIT - IEFSD164
* TABLES/WORK AREAS TIOT,LCT,CSCB
* ATTRIBUTES - REENTRANT
*****
      EJECT
      USING *,R12
      L    R5,ASD062      GET ADDRESS OF IEFSD062.
      LA   R6,DISPLACE(R5) BUMP THE ADDRESS TO THE FTT FLAG.
      MVI  0(R6),X'FF'    SET THE FLAG 'FROM AFAR'.
      USING IEFCSCH,R5
      USING IEFLOT,R6
      USING IOPARAMS,R7
*****
* THE FOLLOWING SECTION OF CODE SERVES ONLY TO IDENTIFY THIS MODULE
* IN A MEMORY DUMP
*
      H      FIRST

```

DC	X'CCCC'		* 00900000
DC	C'IEFSD104'		* 00920000
DC	X'06081966'		* 00940000
DC	X'CCCC'		* 00960000
FIRST	DS	OH	* 00980000
*			* 01000000
*****			* 01020000
FJFCT			01040000
LR	R2,R1	SAVE PARAMETER LIST REGISTER	01060000
L	R3,X16(R2)	LOAD SYSTEM TASK INDICATOR	1272 01070019
LTR	R3,R3	TEST FOR SYSTEM TASK	1272 01080019
BZ	SD104005	BRANCH NOT SYSTEM TASK	1272 01090019
L	R5,X8(R3)	LOAD CSCR ADDRESS	1272 01100019
LA	R0,X104	GETMAIN FOR 24 PARM AREA AND	1272 01110019
GETMAIN R, LV=(0)		72 REG SAVE AREA + 8 REMOTE LI	1272 01120019
LA	R13,X32(R1)	SAVE AREA ADDRESS	1272 01130019
LA	R7,CHQPA	ADDRESS OF QMPA IN CSCR	1272 01140019
ST	R7,X0(R1)	STORE ADDRESS IN PARM	1272 01150019
XC	X4(X20,R1),X4(R1)	ZERO PARM AREA	M2696 01160019
MVC	X4(X3,R1),X4(R3)	MOVE TTR TO PARM AREA	M2696 01170019
LA	R4,X24(R1)	ADDRESS OF REMOTE LIST	1272 01180019
ST	R4,QMPCL	REMOTE LIST PTR IN QMPA	1272 01190019
MVI	QMPDP,X4	INDICATE READ OPERATION	1272 01200019
LR	R10,R1	SAVE ADDRESS	1272 01210019
L	R15,SD514RD	ADDRESS OF IEFSD415	1272 01220019
HALR	R14,R15	READ LCT FROM JOB QUEUE	1272 01230019
LTR	R15,R15	TEST FOR I/O ERROR	1272 01240019
BZ	SD104002	BRANCH NO I/O ERROR	1272 01250019
LA	R1,X'080'	ORO ABEND	1272 01260019
SLA	R1,X12		1272 01270019
ABEND (1),DUMP			1272 01280019
SD104002 L	R6,X8(R1)	ADDRESS OF LCT	1272 01290019
MVC	QMGR1(X36),CHQPA	MOVE MAIN QMPA TO LCT	1272 01293019
MVC	QMGR2(X36),CHSQA	MOVE SYSOUT QMPA TO LCT	1272 01296019
LA	R7,QMGR1	ADDRESS OF QMPA	1272 01300019
L	R11,QMWTQ	ADDRESS OF TRACK STACK INFO	1272 01303019
MVC	TRSTKINF(X8),X0(R11)	MOVE TS INFO	1272 01306019
LA	R4,REGSAVE+72	ADDRESS OF REMOTE LIST	1272 01310019
ST	R4,QMPCL	ADDRESS OF REMOTE LT IN QMPA	1272 01320019
LA	R4,TRSTKINF	ADDRESS OF TRACK STACK INFO	1272 01330019
ST	R4,QMWTQ1	STORE TRACK STACK INFO	1272 01340019
ST	R4,QMWTQ2	IN BOTH QMPAS	1272 01350019
LA	R4,REGSAVE	ADDRESS OF SAVE AREA	1272 01360019
ST	R4,LCTCMCBA	STORE IN LCT	1272 01370019
LA	R4,QMGR2	ADDRESS OF SYSOUT Q MGR	1272 01380019
ST	R4,LCTSNQMP	PARM AREA IN LCT	1272 01390019
L	R8,LCTJSCH	ADDRESS OF JOB STEP JSCH	1272 01391019
USING IEZJSCH,R8			1272 01392019
L	R8,JSCHWTP	ADDRESS OF WTP CONTROL BLK	1272 01393019
DROP R8			1272 01394019
USING WTPCB,R8			1272 01395019
MVC	WTPQMPAD,LCTSNQMP+X1	MOVE ADDRESS OF QMPA	1272 01396019
DROP R8			1272 01397019
LA	R0,X104	LENGTH TO BE FREED	1272 01400019
LR	R1,R10	RESTORE ADDRESS	1272 01410019
FREEMAIN R, LV=(0),A=(1)		FREE PARM REMOTE LT AND RSA	1272 01420019
MVC	LCTPARM1(X16),X0(R3)	MOVE LCT PARMS INTO LCT	1272 01430019

FREEFMAIN R.LV=24,A=(11),SP=TCORE FREE SYSPARM AND TS INFO			1272	01440019
XC	X16(X4,R2),X16(R2)	ZERO PARM PTR	1272	01450019
ST	R6,12(R2)	STORE LCT ADDRESS IN PARM LIST	1272	01460019
SD104005 EQU	"			01600000
LR	R1,R2	RESTORE PARAMETER LIST ADDRESS		02300000
L	R15,SD64VCON	ADDRESS OF IEFS0064		02320000
RR	R15	GO TO IEFS0064		02340000
SD64VCON DC	V(IEFS0064)			02380000
SD514RD DC	V(IEFS0514)			1272 02400019
ASD062 DC	V(IEFS0062)			
R0	EQU	0		02500000
R1	EQU	1		02520000
R2	EQU	2		02540000
R3	EQU	3		02560000
R4	EQU	4		02580000
R5	EQU	5		02600000
R6	EQU	6		02620000
R7	EQU	7		02640000
R8	EQU	8		02660000
R9	EQU	9		02680000
R10	EQU	10		02700000
R11	EQU	11		02720000
R12	EQU	12		02740000
R13	EQU	13		02760000
R14	EQU	14		02780000
R15	EQU	15		02800000
X0	EQU	0		1272 02802019
X1	EQU	1		1272 02803019
X4	EQU	4		1272 02804019
X3	EQU	3		M2696 02804619
X7	EQU	7		M2696 02805219
X8	EQU	8		1272 02806019
X12	EQU	12		1272 02808019
X16	EQU	16		1272 02810019
X20	EQU	20		M2696 02811019
X24	EQU	24		1272 02812019
X32	EQU	32		1272 02814019
X36	EQU	36		1272 02815019
X104	EQU	104		1272 02816019
TCORE	EQU	253		1272 02818019
DISPLACE EQU	760			
EJECT			1272	02818319
IEFWTPCH			1272	02818619
EJECT			1272	02818919
IEZJSCB			1272	02819219
EJECT			1272	02819519
IEFL0T	DSECT			02820000
IEFALLCT				02840000
DS	1F		1272	02846019
LCTSDOMP DS	1F	ADDRESS OF MESSAGE CLASS QUEUE	1272	02852019
*		MANAGER PARAMETER AREA	1272	02858019
DS	8F		1272	02864019
REGSAVE DS	36F		1272	02870019
QMGR1 DS	9F	MAIN QMGR PARAMETER AREA		02880000
QMGR2 DS	9F	'ALTERNATE' QMGR PARAMETER AREA		02900000
QMCAPI EQU	QMGR1+24	PTR IN QMCA TO TRK STACK INFO		02920000
TRSTKINF DS	2F	NEEDED FOR TRACK STACKING AND QUEUE		02940000

-61-

*				BREAK INFORMATION. 1ST BYTE TO CON-	02960000
*				TAIN NUMBER OF BUFFERS, NEXT 3 BYTES	02980000
*				FOR STACK ADDR, NEXT 4 BYTES FOR	03000000
*				QUEUE BREAK INFORMATION	03020000
OMWT01	EQU	OMGR1+24	PTR IN	OMPA TO TRACK STACK INFO	1272 03022019
OMWT02	EQU	OMGR2+24	PTR IN	OMPA TO TRACK STACK INFO	1272 03024019
	DS	8F			1272 03025019
LCTJSCB	DS	F		ADDRESS OF P/P JSCH	1272 03026019
IEFEND	EQU	*			1272 03028019
	EJECT				1272 03029019
IEFOMNGR					1272 03030019
	EJECT				1272 03031019
IEFCSCB	DSECT				1272 03032019
IEFCHAIN					1272 03034019
END					03040000

IEFW42SD CSECT

```

R0      EQU      0
R1      EQU      1
R2      EQU      2
R3      EQU      3
R4      EQU      4
R5      EQU      5
R6      EQU      6
R7      EQU      7
R8      EQU      8
R9      EQU      9
R10     EQU      10
R11     EQU      11
R12     EQU      12
R13     EQU      13
R14     EQU      14
R15     EQU      15
        ENTRY    IEFV4221
        SAVE     (14,12).T
        DS       OH
        HALR     R9,R0
        USING    *,R9
        B        FIRST
        DC       X'CCCC'
        DC       CL8'IEFSD420'
        DC       X'05161966'
        DC       X'CCCC'
FIRST    DS       OH
        USING    LCT,R12
        L        R12,0(R1)
        LR       R2,R1
        L        R6,LCTQENTY
        SRL      R6,24
        SLL      R6,24
        SLL      R13,8
        SRL      R13,8
        AR       R13,R6
        ST       R13,LCTQENTY
        SR       R3,R3
        L        R15,CVTPTR
        USING    CVTDSECT,R15
        L        R10,CVTMSER
        USING    BASE,R10
        TM       MSLOGST,MSLOGEND
        BZ       IEFV4221
        DROP     R10,R15
        L        R0,ENQLOGCR
        GETMAIN  R,LV=(0)
        LA       R3,NO(,R1)
        MVC      NO(ENQLOGF,R3),ENQLOG
        USING    ENQLOG,R1
        ENQ      (ENQLOGQR,ENQLOGQR),MF=(F,(1))
        DROP     R1
        LTR      R15,R15
        BNZ      IEFV4221
        LNR      R3,R3
SYSOUT   XCTL    EP=IEFVLDSP

```

```

IEFV4221 EQU *
      DROP R9
      BALR R9,0
      USING *,R9
      L R8,ASD062          GET ADDR. OF IEFSD062.
      LA R8,DISPLACE(R8)   BUMP TO FIT FLAG.
      MVI 0(R8),X'FF'      REMOTE SET.
      USING LCT,R12
      LTR R3,R3
      BZ SECOND
      BP NOLOG
      LPR R1,R3
      DEQ MF=(E,(1))
NOLOG EQU *
      L R0,ENQLOGCR
      LPR R1,R3
      FREEMAIN R,LV=(0),A=(1)
SECOND DS OH
      MVC LCTJCTAD(4),4(R2)
      MVC LCTSCAD(4),8(R2)
      MVC LCTPARM1(4),12(R2)
      ST R2,LCTATLST
      L R5,LCTTCHAD
      USING TCB,R5
      USING CVTDSECT,R15
      L R15,CVTPTR
      TM CVTDCB,X'20'
      BO IEFV42H
      XC TCBTID(4),TCRTID
IEFV42H EQU *
      L R3,LCTJCTAD
      USING IEFJCT,R3
      TM JCTJSTAT,INCMSTS+JCTJOBFL+JCTABEND+JCTSTPFL
      BZ RJENOSSET
      LA R3,QMGR2
      USING IOPARAMS,R3
      OI QMSTA,RJENOFAL
      DROP R3
RJENOSSET EQU *
      GETMAIN R,LV=176
      ST R1,LCTSMBAD
      XC 0(176,R1),0(R1)
      USING SCT,R2
      L R2,LCTSCAD
      L R10,LCTJCTAD
      USING IEFJCT,R10
      MVC 0(4,R1),SCTADSMB
      MVI TTRO(R1),SMBID
TESTWTP EQU *
      L R15,CVTPTR
      USING CVTDSECT,R15
      TM CVTDCB,MFT
      DROP R15
      BN GOON
      L R11,LCTJSCB
      B WTPPTR
GOON L R11,TCBJSCB(R5)

```

```

WIPPER      L      R11,WIPCRPTR(R11)
            LA      R11,HIGHRYTE(R11)
            LTR     R11,R11
            USING   WIPDSECT,R11
            BNZ     TESTWTP1
HOLDON      EQU     *
            TM      SCTSTYPE,X'20'
            RD      IFFV42R
WTPRETN      EQU     *
            L      R13,LCTCMCHA
            USING   IOPARAMS,R
            LA      R1,LCTOMPAM(R13)
            LA      R4,72(R13)
            MVC     0(4,R4),LCTSMHAD
            LTR     R11,R11
            RZ      NOWTP
            MVC     04(1TR,R4),WTPSMR
            MVI     07(R4),HEX0H
            R       WTPOVER1
NOWTP        EQU     *
            MVC     4(4,R4),SCTADSMR
WTPOVER1     EQU     *
            L      R7,LCTSMHAD
            MVI     0MPDP,X'04'
            ST      R4,0MPCL
            MVI     0MPCM,X'01'
READNEXT     EQU     *
            L      R15,IFFV42V1
            RALR    R14,R15
            LTR     R11,R11
            RZ      IFFV42R
            TM      WTPFLGSA,WTPINVKD
            RZ      IFFV42R
            MVI     NEXTHYTE(R7),FULL
IFFV42R      DS      0H
            TM      TCHFLGS,TCHFA
            RC      1,IFFV42S
            TM      JCTRSW2,X'01'
            RZ      IFFV42T
            NI      JCTRSW2,X'FE'
            NI      LCTOENTY,LCTERRM
            MVC     LCTERROR(4),IFFV42C1
IFFV42T      EQU     *
            TM      SCTSTAT.4
            RD      IFFNEXT
            NI      JCTRSW1,255-JCTCKFT
IFFNEXT      EQU     *
            L      R15,IFFV42V3
            RR      R15
IFFV42S      DS      0H
            L      R15,IFFV42V4
            RR      R15
TESTWTP1     EQU     *
            TM      WTPFLGSA,WTPINVKD
            RZ      HOLDON
            MVC     SCTCRWTP(04),WTPCRSMR
            R       WTPRETN

```

OROP R11

*
* CONSTANTS AND DSECTS.
*

DISPLAC EQU 760

NEXTSMR EQU 4
D7 EQU 7
TTR0 EQU 3
TTR EQU TTR0
HIGHRYTF EQU 0
D4 EQU 4
HEXHH EQU X'00'
NEXTBYTE EQU 13
FULL EQU X'A1'
WTPCHPTR EQU X'FC'
TCRJSCB EQU X'H4'
MFT EQU X'20'

ASD062 DC V(IEFSD062)

IEFV42V1 DC V(IEFQMRW)
IEFV42V3 DC V(IEFYN)
IEFV42V4 DC V(IEFIDUMP)
ENQLOG FNO (, ,E,06,SYSTEM),RFT=USE,MF=L
ENQLOGQO DC CL8'SYSIEFSD'
ENQLOGQR DC CL6'SYSLQG'
ENQLOGE EQU *-ENQLOG
DS OF
ENQLOGCR DC AL1(255)
DC AL3(ENQLOGE)
NO EQU 0
N255 EQU 255
IEFV42C1 DC C'DSDR'
RJENDFAL EQU 8
JCTJOBFL EQU 64
JCTSTPFL EQU 16

*
TCB DSECT
DS CL12
TCBTIO DS CL4
DS CL13
TCBFLGS DS CL5
TCBFA EQU X'80'

IEFWTPCH
LCT DSECT
IEFALLCT
LCTRATMN DS F
LCTSDQMP DS F
LCTRTRN DS F
LCTCSCH DS F
LCTTMWRK DS 4F
LCTJOBFL DS F
LCTATLST DS F
REGSAVE DS 36F
QMGR1 DS 9F
QMGR2 DS 9F
IEFEND EQU *
LCTJSCH EQU LCT+360
SCT DSECT

IEFASCTR
IEFQMPAM
CVTDSECT DSECT
CVT
IEFJCT DSECT
IEFAJCTR
IEFRASEA
END

```

TITLE 'IEFSD165 VMS INITIATOR ERROR MODULE'
IEFSD065 CSECT
*2186016700-026300
*****
*
*               IEFSD065
*       NO ATTACH TERMINATE INTERFACE
*
*****
*STATUS: CHANGE LEVEL 000
*
*FUNCTION:  TO INTERFACE WITH THE TERMINATE ROUTINES WHEN AN ERROR
*           IN ALLOCATE CAUSES THE JOB STEP TO BE NOT ATTACHED.
*
*ENTRY POINTS: IEFSD065 FROM THE ALLOCATE/ATTACH MODULE IEFSD062
*
*INPUT:      REG 1 POINTING TO A PARAMETER LIST CONTAINING POINTERS
*           TO-*****
*               *   LCT   *
*               *   *   *
*               *   JCT   *
*               *   *   *
*               *   SCT   *
*               *   *   *
*               * TIO TLIST *
*               *   *   *
*               *   0   *
*               *   *   *
*
*OUTPUT: REG 1 POINTING TO LCT
*
*EXTERNAL ROUTINES: IEFQMRAW - Q MNGR READ/WRITE ROUTINE
*                   IEFW42SD - TERMINATE
*
*TABLES AND WORK AREAS: LCT-TCB
*
*EXITS  IEFSD162 - IF MORE STEPS TO BE RUN
*EXITS  IEFSD166 - IF END OF JOB
*EXITS  IEFSD168 - IF JOB IS TO BE SUSPENDED
*
*ERROR MESSAGES: NONE
*
*ATTRIBUTES: REENTRANT
*
*****
*
*       EJECT
*       BALR 12,0
*       USING *,12
*       USING IEFL0T,R11
*****
*       THE FOLLOWING CODE SERVES ONLY TO MAKE THIS MODULE EASIER
*       TO IDENTIFY IN A DUMP LISTING.

```

```

00020000
00040000
00050017
00060000
* 00080000
* 00100000
* 00120000
* 00140000
* 00160000
* 00180000
* 00200000
* 00220000
* 00240000
* 00260000
* 00280000
* 00300000
* 00320000
* 00340000
* 00360000
* 00380000
* 00400000
* 00420000
* 00440000
* 00460000
* 00480000
* 00500000
* 00520000
* 00540000
* 00560000
* 00580000
* 00600000
* 00620000
* 00640000
* 00660000
* 00680000
* 00700000
* 00720000
* 00740000
CR17 00750017
CR17 00760017
CR17 00770017
* 00800000
* 00820000
* 00840000
* 00860000
* 00880000
* 00900000
* 00920000
* 00940000
* 00960000
* 00980000
* 01000000
* 01020000
* 01040000
* 01060000
* 01080000
* 01100000

```

```

***** 01120000
R      FIRST                                01140000
DC     X'CCCC'                             01160000
DC     C'IEFSD165'                         01180000
DC     X'05031966'                         01200000
DC     X'CCCC'                             01220000
FIRST  DS     OH                           01240000
***** 01260000
*      SETUP TO BUILD DUMMY TCB TO SATISFY REQUIREMENTS TO ENTER * 01280000
*      TERMINATE.                               * 01300000
***** 01320000
LR     R3,R1                                SAVE POINTER TO PARAMETER LIST 01340000
*      BUILT BY SD062                               * 01360000
L      R1,0(R3)                                LOAD LCT BASE                241 01362018
SR     R1,R1                                CLEAR REG                    241 01364018
JC     R1,LCTQDRTY                           GET JOB PRIORITY             241 01366018
LA     R0,15                                PUT 15 IN REG                241 01368018
SR     R0,R1                                GET 15-PRTY                  241 01370018
CHAP   (0)                                CHAP INIT UP TO UNSHADED PRTY 241 01372018
LA     R1,TCORE                               SUBPOOL                      1272 01376019
SLL    R1,024                               SUBPOOL IN HIGH-ORDER BYTE  1272 01380019
LA     R0,TCLEN                               LENGTH OF TCB                1272 01384019
OR     R0,R1                                SUBPOOL IN HIGH-ORDER BYTE  1272 01388019
GETMAIN R,LV=(0)                             GET CORE FOR DUMMY TCB      1272 01392019
XC     ZERO(TCHLEN,R1),ZERO(R1) ZERO DUMMY TCB M4340 01422019
ST     R1,LCTTCBAD                           STORE POINTER TO DUMMY TCB IN 01460000
*      LCT                                           * 01480000
TCHTCT EQU 164                                SMF1 01482018
L      R4,LCTQDRTY                           PT TO CSCB                   SMF1 01484018
USING  IEFCSCB,R4                                SMF1 01486018
L      R4,CHSWT                               NOW PT TO TCB                SMF1 01488018
MVC    TCBJSCB+D1(D3,R1),LCTJSCB+D1 ADDR JSCB TO TERM M4340 01490019
DROP   4                                        01490018
MVC    TCHTCT(4,R1),TCHTCT(R4) COPY TCT PTR TO FAKE TCB SMF1 01492018
***** 01500000
*      SETUP TO GETCORE FOR REGSAVE AREA AND BRANCH TO TERMINATE * 01520000
*      AFTER RESTORING REG 1 TO POINT TO PARAMETER LIST BUILT BY * 01540000
*      SD062.                               * 01560000
***** 01580000
LR     R6,R1                                SAVE ADDR DUMMY TCB        M4340 01590019
USING  IEFSCT,R5                                CR17 01610017
USING  IEFCSCB,R7                                CR17 01620017
USING  IOPARAMS,R10                             CR17 01630017
L      R5,LCTSCTAD                               SCT BASE.                    CR17 01640017
L      R7,LCTQDRTY                               CSCB BASE.                   CR17 01650017
LA     R10,QMGR1                                QMPA BASE.                   CR17 01660017
SD65005 EQU *                                02640000
GETMAIN R,LV=72,SP=TCORE                      02660000
LR     R13,R1                                02680000
LR     R1,R3                                RESTORE PARAM LIST POINTER  02700000
L      R15,SD42VCON                            ADDRESS OF TERMINATE        02720000
BALR   R14,R15                                GO TO TERMINATE             02740000
***** 02760000
*      UPON RETURN FROM TERMINATE- SETUP TO FREECORE FOR THE DUMMY * 02780000
*      TCB. TERMINATE RETURNS A CODE IN REGISTER 15 TO DENOTE IF JOB * 02800000
*      TERMINATE WAS ENTERED OR NOT. 0= NOT ENTERED- 4= ENTERED * 02820000
***** 02840000

```

LR	R7,R15	SAVE RETURN CODE	02860000
LR	R1,R3	ADDRESS OF PARAMETER LIST	02880000
FREEFMAIN	R, LV=24, A=(1), SP=TCORF		02900000
LA	R1, TCORF	SUBPOOL	1272 02904019
SLL	R1, 024	SUBPOOL IN HIGH-ORDER BYTE	1272 02908019
LA	R0, TCHLEN	LENGTH OF TCR	1272 02912019
OR	R0, R1	COMPOOL IN HIGH-ORDER BYTE	1272 02916019
LR	R1, R6	RESTORE ADDR DUMMY TCR	M4340 02926019
FREEFMAIN	R, LV=(0), A=(1) FREE DUMMY TCR		1272 02940019
LR	R1, R13	RESTORE POINTER TO REG SAVE AREA	02960000
FREEFMAIN	R, LV=72, A=(1), SP=TCORF		02980000
LA	R15, 4		CR17 03000017
CR	R7, R15	IS THIS JOB TERMINATION.	CR17 03020017
RNF	SD65226	NO.	CR17 03040017
		YES, JOB TERMINATION	M4340 03042019
L	R1, LCTJSCR	ADDR EXEC PGM JSCR	M4340 03044019
USING	IFZJSCR, R1	JSCR ADDRESSABILITY	M4340 03046019
L	R1, JSCRWTP	ADDR WTPCH	M4340 03048019
DROP	R1		M4340 03050019
LA	R1, ZERO(R1)	CLEAR HIGH-ORDER BYTE	M4340 03052019
USING	WTPCH, R1	WTPCH ADDRESSABILITY	M4340 03054019
XC	WTPSMR(WTPCHS17), WTPSMR	ZERO WTPCH	M4340 03056019
DROP	R1		M4340 03058019
LR	R1, R11	YES.	CR17 03060017
L	R15, SD66VCNN		CR17 03080017
BR	R15	GO TO IFESD166	CR17 03100017
SD65226	EQU *		CR17 03120017
L	R1, LCTJSCR	ADDR EXEC PGM JSCR	M4340 03121019
USING	IFZJSCR, R1	JSCR ADDRESSABILITY	M4340 03122019
L	R1, JSCRWTP	ADDR EXEC PGM WTPCH	M4340 03123019
DROP	R1		M4340 03124019
LA	R1, ZERO(R1)	CLEAR HIGH-ORDER BYTE	M4340 03125019
USING	WTPCH, R1	WTPCH ADDRESSABILITY	M4340 03126019
XC	WTPSMR, WTPSMR	ZERO WTPSMR SLOT	M4340 03127019
NI	WTPFLGSA, WTPCLEAN	TURN OFF ALL STEP FLAGS	M4340 03128019
XC	WTPHYTES(08), WTPHYTES	ZERO 2 WORDS IN WTPCH	M4340 03129019
DROP	R1		M4340 03130019
LA	R1, REGSAVE+72	YES, OMGR REMOTE AREA.	CR17 03140017
ST	R5, 0(0, R1)	SCT CORE ADDRESS TO REMOTE AREA	03240000
MVC	4(4, R1), SCTANSCT	SCT TTR TO REMOTE AREA	03260000
ST	R1, QMPCL	REMOTE AREA TO QMPAM	03280000
MVI	QMPOP, 4	INDICATE READ OPERATION	03300000
MVI	QMPCL, 1	INDICATE 1 OPERATION	03320000
LA	R13, REGSAVE	REGISTER SAVE AREA	03340000
L	R15, QMVCON	ADDRESS OF Q MGR	03360000
LR	R1, R10	ADDRESS OF PARM AREA	03380000
RALR	R14, R15	GO TO READ IN SCT	03400000
R	SD65020(R15)	CHECK RETURN CODE	03420000
SD65020	R	RETURN CODE OK	03440000
RC	0, SD65020	NO WORK, IMPOSSIBLE ON READ	03460000
BC	0, SD65020	NO SPACE, IMPOSSIBLE ON READ	03480000
*****		I/O ERROR	03500000
*****		CODE TO PROCESS THIS CONDITION	03520000
LA	R1, X'080'	SHOULD BE PLACED HERE	03540000
SLL	R1, 12	TEMPORARILY, ABEND	03560000
AREND	(1), DUMP		03580000
*****			03600000

```

SD65030 EQU *                                03620000
        IC R1,LCTSNMB                         UPDATE          03640000
        LA R1,1(0,R1)                         STEP           03660000
        STC R1,LCTSNMB                        NUMBR IN ICT     03680000
*****
*                                * 03700000
*                                * 03720000
* LCTPARM1 IS TESTED IN IFFSD102 FOR DATA SET ENQUEUING,AND MUST BE * 03740000
* SET TO ZERO BEFORE EXITING IFFSD165 TO IFFSD101.                * 03760000
*                                * 03780000
* XC LCTPARM1,LCTPARM1                                         03800000
*                                * 03820000
*****                                03840000
        L R15,SD62VCON GET ADDRESS OF IFFSD062.
        LA R15,DISPLACE(R15) BUMP IT TO FTT FLAG.
        MVI 0(R15),X'FF' SET THE FLAG.
        LR R1,R11 LCT ADDRESS TO R1                                03860000
        L R15,SD62VCON                                03880000
        BR R15 BRANCH TO SD062 TO ALLOCATE FOR
                                AND ATTACH ANOTHER JOB STEP * 03900000
*                                * 03920000
SD68VCON DC V(IEFSD068) CR17 03926017
        EJECT                                03932017
SD66VCON DC V(IEFSD066)                                03940000
SD62VCON DC V(IEFSD062)                                03960000
QMVCN DC V(IEFQMRAW)                                03980000
SD42VCON DC V(IEFW42SD)                                04000000
DISPLACE EQU 760
TCORE EQU 253
TIOTPTR EQU 12 DISPLACEMENT OF TIOT PTR IN TCB 04020000
TCBFLGS EQU 29 CR17 04050017
TCBLEN EQU 184 LENGTH OF TCB 1272 04052019
TCRJSCB EQU 180 JSCR DISPLACEMENT IN TCB 1254 04054019
ZERO EQU 0 DISPLACEMENT M4340 04054519
D1 EQU 1 DISPLACEMENT M4340 04055019
D3 EQU 3 DISPLACEMENT M4340 04055519
D4 EQU 4 DISPLACEMENT 1254 04056019
D8 EQU 8 DISPLACEMENT M4340 04057019
D24 EQU 24 DISPLACEMENT 1272 04058019
*****                                04060000
* REGISTER EQUATE STATEMENTS * 04080000
*****                                04100000
R0 EQU 0 04120000
R1 EQU 1 04140000
R2 EQU 2 04160000
R3 EQU 3 04180000
R4 EQU 4 04200000
R5 EQU 5 04220000
R6 EQU 6 04240000
R7 EQU 7 04260000
R8 EQU 8 04280000
R9 EQU 9 04300000
R10 EQU 10 04320000
R11 EQU 11 04340000
R12 EQU 12 04360000
R13 EQU 13 04380000
R14 EQU 14 04400000
R15 EQU 15 04420000
EJECT 04430017

```

IEFCSCB	DSECT		04440000
	IEECHAIN		04460000
	EJECT		04470017
IHADCB	DSECT		04480000
DCRSTART	EQU *		04500000
	DCB DSORG=PO,MACRF=E		04520000
IEFXLIST	DC F'0'	DCB EXIT LIST	04540000
DCREND	EQU *		04560000
	EJECT		04580000
IEFSCT	DSECT		04600000
	IEFASCTB		04620000
	EJECT		04630017
	IEFQMNGR		04640000
QMOSC	EQU QMTID+2	CR17	04650017
IEFLOT	DSECT		04660000
*****			04680000
*		*	04700000
*	THE FOLLOWING REPRESENTS A BLOCK OF MAIN STORAGE WHICH IS	*	04720000
*	ALLOCATED FOR THE LIFE OF	*	04740000
*	OF THIS INITIATOR. IT CONTAINS:	*	04760000
*		*	04780000
*	(1) THE LCT (LINKAGE CONTROL TABLE)	*	04800000
*	(2) A 2 LEVEL REGISTER SAVE AREA FOLLOWED BY THE	*	04820000
*	THE QUEUE MANAGER PARAMETER AREA	*	04840000
*	(3) AN ALTERNATE QUEUE MANAGER PARAMETER AREA	*	04860000
*		*	04880000
*****			04900000
IEFALLCT			04920000
LCTBATMN DS	F *****TO BE ADDED TO LCT		04940000
LCTSOQMP DS	F *****TO BE ADDED TO LCT(VMS ONLY) POINTER TO		04960000
*	SYSOUT QUEUE MANAGER PARAMETER AREA		04980000
LCTRTRN DS	F *****TO BE ADDED TO LCT(VMS ONLY) INITIATOR'S		05000000
*	RETURN POINT		05020000
LCTCSCB DS	F *****TO BE ADDED TO LCT(VMS ONLY) INITIATOR CSCB		05040000
*	ADDRESS		05060000
LCTTMWRK DS	4F *****TO BE ADDED TO LCT(VMS ONLY) TIMER WORK AREA		05080000
LCTJOBRLB DS	F *****TO BE ADDED TO LCT(VMS ONLY) JOBLIB DCR PTR		05100000
LCTATLST DS	F *****TO BE ADDED TO LCT(VMS ONLY) POINTER TO		05120000
*	ALLOCATE/TERMINATE PARAMETER LISTS		05140000
REGSAVE DS	36F ALLOCATE/TERMINATE REGSAVE AREA		05160000
QMGR1 DS	9F QUEUE MANAGER PARAMETER AREA		05180000
QMGR2 DS	9F ALTERNATE QUEUE MANAGER AREA		05200000
DS	8F TO CORRECT DISPLACEMENT	M4340	05201019
* THESE FIELDS ARE NEEDED FOR L-SHAPE/INIT MERGE			* 1272 05202019
DS	0F		1272 05203019
LCTOPSW1 DS	0C INITIATOR OPTIONS BYTE 1		1272 05204019
LCTPKEYF EQU	128 DONT GET PROTECT KEY		1272 05205019
LCTDWFF EQU	64 DONT PROCESS DEDICATED WORKF		1272 05206019
LCTSTMDF EQU	32 DONT PROCESS STOP/MODIFY		1272 05207019
LCTMINPF EQU	16 GET REGION SIZE SPECIFIED		1272 05208019
LCTCANF EQU	8 ALLOW CANCEL ONLY AT ALLOC		1272 05209019
LCTONEJF EQU	4 PROCESS ONLY ONE JOB		1272 05210019
LCTICMDF EQU	2 DONT PROCESS INITIATOR CMDS		1272 05211019
LCTEXIT DS	F ADDRESS INITIATOR EXIT LIST		1272 05212019
LCTOPSW2 DS	0C INITIATOR OPTIONS BYTE 2		1272 05213019
LCTTIMEF EQU	128 DONT TIME THIS JOB		1272 05214019
LCTCRF EQU	64 DONT ALLOW CHECKPT/RESTART		1272 05215019

-72-

LCIOSUE	FOU	32	POINT PROCESS DSO	1272	05216019
LCICOM	DS	F	COMMUNICATIONS PARAM AREA PTR	1272	05217019
LCIJSCH	DS	F	ADDRESS OF JSCH FOR NEW TASK	1254	05218019
IEFEND	FOU	*	END OF LIFE-OF-TASK BLOCK		05220000
		IEZJSCH			05226019
WTPPTR	FOU	JSCHWTP-IEZJSCH	DISPLACEMENT OF WTP PTR	1254	05232019
		IEFWTPCH		M4340	05234019
WTPCLEAN	FOU	X'4C'		M4340	05236019
		EJECT			05240000
		END			05260000

0

```

TITLE 'IEFSD162 VMS INITIATOR STEP PROCESSOR'
IEFCVT  DSECT
      CVT
      EJECT
IEFSD062 CSECT
* 055610-055630,055700,055805,059400,063110,082750
* 066390-066550
* 069800-070200,070800
      ENTRY IEFSD062A
* 059000-060600,063260-063740,083000-083200,086200
*2056043971
*00059
*1785043669,043687,068200,068400,069600,070800,083800
*0642
*011400,044400-045400,045800,051190,063800,066600,072620,074800-
*076800,08100-081600,084000,085660-085720,090800-091000
*016400,017600,027200-029600,029610-029630,029800-030200,031000
*036500,043414-043597,043633-043642,080400,099200-099400
*029800,031120,031200-031405,031420,031420,031490-031500
* 030890-030917,030935-031023,031502
*2229056500-056700
*2842
*
* 031293-031294
*
* 030926-031261
*****
*
      VMS ALLOCATE ENTRANCE/ATTACH INTERFACE MODULE
*
*****
* STATUS-      CHANGE LEVEL - 0
* FUNCTION: 1. COMPLETE ENTRANCE INTERFACE FOR ALLOCATE
*           1A. ENTRANCE INTERFACE FOR TIOT WRITE (IEFSD0514) ROUTINE CR17 00250017
*           1B. UPDATE THE CSCH - SCAN THE SIOT FOR IFFRDR IF TASK 1272 00252019
*              NAME IS 'STARTING' - SCAN THE TIOT ON RETURN 1272 00254019
*              FROM ALLOCATION FOR IFFRDR. 00256019
*           2. DETERMINE (BY ALLOCATE EXIT PARAMETER) WHETHER OR NOT * 00260000
*              STEP SHOULD BE ATTACHED. IF SO, BUILD AN ATTACH * 00290000
*              PARAMETER LIST FOR ATTACH. IF NOT, PASS THE ALLOCATE * 00300000
*              EXIT PARAMETER LIST TO IEFSD165 * 00320000
*           3. IF OPEN JOBLIB OR OPEN FETCH DATASET FAILS, ATTACH * 00340000
*              IEFSD0XX INSTEAD OF THE USFRS JOB STEP. THIS WILL * 00360000
*              FAIL ON THE OPEN AGAIN AND ABEND THE JOB STEP. * 00380000
*           4. CONDITIONALLY LINK TO IEFSD0WR TO WRITE JOB 099 00385019
*              SEPARATORS AND/OR SYSTEM MESSAGES (SMH'S) 099 00390019
*              FOR DIRECT SYSOUT DATA SETS. 099 00395019
*
* ENTRY POINT: IEFSD162 FROM IEFSD161, IEFSD164, AND IEFSD165.
*              IEFSD062A FROM IEFSD164 FOR IMMEDIATE RESTART CR17 00430017
*
* INPUT: 1. LCT POINTED TO BY REGISTER ONE * 00460000
*        2. LPMOD VALUE IN 1ST BYTE OF LCTODRTY * 00480000
*        3. IF THE ENTRY IS TO IEFSD062A, REGISTER 1 POINTS TO TH CR17 00481017
*           PARAMETER LIST THAT CONTAINS- CR17 00482017

```

[illegible]

```

R6      EQU      6                      01380000
R7      EQU      7                      01400000
R8      EQU      8                      01420000
R9      EQU      9                      01440000
R10     EQU      10                     01460000
R11     EQU      11                     01480000
R12     EQU      12                     01500000
R13     EQU      13                     01520000
R14     EQU      14                     01540000
R15     EQU      15                     01560000

EJECT                                     01580000
BALR    R12,0                           01600000
USING   IEFPARAM,R11                    01620000
USING   IOPARAMS,R9                     01660000
USING   IEFLDT,R8                       01680000
USING   IEFJCT,R7                       01700000
USING   IEFSCCT,R6                      01720000
USING   IEFCSCH,R5                      01740000
USING   *,R12                           01780000
R        FIRST                           CR17 01781017
IEFS062A BALR    R12,0                    CR17 01782017
LA      R11,IEFS062A-IEFS0062           CR17 01783017
SR      R12,R11                         CR17 01784017
L        R8,0(R1)                       LCT - BASE REG. CR17 01785017
L        R7,4(R1)                       JCT - BASE REG. CR17 01786017
L        R6,8(R1)                       SCT - BASE REG. CR17 01787017
LR      R2,R1                           PARAM LIST ADDR.(APL) CR17 01787219
LA      R9,QMGR1                        ADDR OF QMPA CR17 01787419
L        R1,CVTPTR                      POINTER TO CVT I272 01787619
USING   IEFCVT,R1                      I272 01787819
L        R1,CVTTCP                      POINTER TO NEW/OLD TCB I272 01788019
DROP    R1                             I272 01788219
L        R1,X4(R1)                      ADDRESS OF CURRENT TCB I272 01788419
ST      R1,LCTTCRAD                     STORE TCB ADDRESS IN LCT I272 01788619
L        R3,TCHJSCB(R1)                 ADDRESS OF INIT JSCB I272 01788819
L        R0,LCTJSCB                     ADDRESS OF JOB STEP JSCB I272 01789019
ST      R3,LCTJSCB                     PUT JOB STEP JSCB IN INIT TCB I272 01789219
ST      R0,TCHJSCB(R1)                 FOR REFERENCE BY OPEN I272 01789419
XC      LCTPARM4(4),LCTPARM4           ZERO OUT LCTPARM4 FOR REGI CR17 01790017
R        SD062030                      CR17 01791017
EJECT                                     01800000
***** 01820000
* 01840000
* THIS SECTION OF CODE IS FOR PROVIDING A METHOD OF IDENTIFYING * 01860000
* THIS MODULE IN A MEMORY DUMP * 01880000
* 01900000
DC      X'CCCC'                        EYE-CATCHER 01940000
DC      C'IEFS0162'                    01960000
DC      X'05031966'                    01980000
DC      X'CCCC'                        EYE-CATCHER 02000000
FIRST   DS      OH                     02020000
* 02040000
***** 02060000
EJECT 02080000
***** 02100000
* 02120000
* THE FOLLOWING SECTION OF CODE BRANCHES TO THE ALLOCATE SUBROUTINE * 02140000

```

* AND, UPON ALLOCATE'S RETURN, CHECKS A RETURN CODE IN R15. IF R15 IS * 02160000
 * ZERO, PROCESSING CONTINUES NORMALLY. IF R15 IS NOT ZERO, THE * 02180000
 * ATTACH MODULE (IEFSD063) IS BYPASSED * 02200000
 * LINKING TO IEFSD063 IS ALSO PERFORMED TO HANDLE WRITING JOB 099 02206019
 * SEPARATORS AND SMPS IF DIRECT SYSOUT (DSO) IS ACTIVE FOR JOB 099 02212019

 * 02220000
 * 02240000

LR	R9,R1	ADDRESS OF LCT	
L	R6,LCTSCAD	ADDR OF SCT.	CR17 02480017
L	R7,LCTJCTAD	JCT ADDRESS FROM LCT	1272 02580019
LA	R9,OMGR1	LOAD ADDR OF OMGR PARAM AREA	02585019
L	R3,OMCAP	PTR TO TRK STACK INFO	02590019
SR	R5,R5		02595019
IC	R5,0(R3)	LOAD NO. OF BUFFERS IN STACK	02600019
LTR	R5,R5	IF IT IS ZERO,	02605019
HC	R,IEFSD022	BRANCH AROUND	02610019
GETMAIN	R,LV=72	GET CORE FOR REG SAVE AREA	02615019
LR	R13,R1	LOAD ITS ADDR INTO REG	02620019
LA	R1,OMGR1	PTR TO MAIN OMPA	02625019
L	R15,IEFSDSTI	ADDR OF STACK INIT. RTN	02630019
HALR	R14,R15	GO GET STACK	02635019

* IEFSD022 EQU *			
L	R5,LCTQRTY	ADDRESS OF CSCR	1272 02640019
TM	CHSWT,CHSYS	TEST FOR SYSTEM TASK	1272 02645019
BZ	IEFSD024	BRANCH NOT SYSTEM TASK	1272 02650019
CLC	SCTSCLP(X8),ACSTART	TASKNAME ON COMMAND	1272 02655019
HNE	SKIPSCAN	TASK SPECIFIED SKIP SCAN	1272 02660019
LA	R0,X256	GET CORE 8 BYTES FOR PARM.	1272 02665019
GETMAIN	R,LV=(0)	176 BYTES FOR READ AREA, 72 RSA	1272 02670019
LR	R3,R1	ADDRESS OF EXTERNAL LIST	1272 02675019
LA	R10,X8(R3)	ADDRESS OF READ IN AREA	1272 02680019
LA	R13,X184(R3)	ADDRESS OF REG SAVE AREA	1272 02685019
USING	SLOT,R10	TEMP SLOT BASE	02690019
ST	R10,X0(R3)	ADDRESS OF READ IN AREA	1272 02695019
MVC	X4(X4,R3),SCTFSIOT	TTR OF SIOT FROM SCT	1272 02700019
CLI	SCTNSIOT,NOSIOTS	ARE THERE ANY SIOTS	1272 02705019
RE	NODORDER	BRANCH NO SIOTS	1272 02710019
SR	R2,R2	ZERO REGISTER	1272 02715019
IC	R2,SCTNSIOT	NUMBER OF SIOTS	1272 02720019
ST	R3,OMPCM	PARM LIST	1272 02725019
MVI	OMPCM,ONFREC	READ ONE RECORD	1272 02730019
MVI	OMPOP,READCODE	READ OP CODE	1272 02735019
KEREAD	L R15,IEFQMGRV	ADDRESS OF IEFQMRAW	1272 02740019
LR	R1,R9	ADDRESS OF OMPA	1272 02745019
HALR	R14,R15		1272 02750019
LTR	R15,R15	TEST FOR IO ERROR	1272 02755019
HNE	SD062015	BRANCH I/O ERROR	1272 02760019
CLC	SCTDDNAM(X8),IEFRDR	IS DDNAME IEFRDR	1272 02765019
RE	DORDER	BRANCH YES	1272 02770019
RCT	R2,NEXTTTR	GET NEXT TTR	1272 02775019
R	NODORDER	IEFRDR NOT FOUND DEFAULT PROC	1272 02780019
NEXTTTR	MVC X4(X4,R3),SCTPSIOT	TTR OF NEXT TTR IN CHAIN	1272 02785019
R	REREAD		1272 02790019
NODORDER	MVC CHKEY(X8),JCTJNAME	MOVE PROC TO CSCR AS TASK NAME	1272 02795019
MVC	SCTJNAME(X8),JCTJNAME	MAKE JOBNAM STEPNAM	1272 02800019
			1272 02805019
			1272 02810019

	B	IEF16205		1272	02815019
SKIPSCAN	MVC	SCTSNAME(X8),SCTSC LPC	MOVE TASKNAME TO STEP NAME	1272	02830019
	MVI	SCTSC LPC,BLANK	INSERT BLANK	1272	02835019
	MVC	SCTSC LPC+X1(X7),SCTSC LPC	BLANK STEP	1272	02840019
	B	IEFSD024		1272	02845019
DDRDER	CLC	SCTUTYPE(X4),BLANKFW	IS UNIT TYPE SPECIFIED	1272	02850019
	RE	NODRDER	BRANCH ON NO UNIT	1272	02855019
	MVC	SCTSNAME(X8),SCTUTYPE	SAVE UNIT NAME IN SCT	1272	02860019
	MVC	CHKEY(X8),SCTUTYPE	SAVE UNIT WANTED IN CSCB	1272	02865019
IEF16205	EQU	*		1272	02870019
	MVI	SCTSC LPC,BLANK	INSERT BLANK	1272	02875019
	MVC	SCTSC LPC+X1(X7),SCTSC LPC	BLANK STEP THAT CALLED	1272	02880019
		DROP R10			02885019
		USING IEFTIOT,R10		1272	02890019
	OI	LCTINTSW,LCTSTART	SET SWITCH	1272	02895019
	L	R10,LCTTCBAD	ADDRESS OF INIT'S TCB	1272	02896019
	L	R10,TCBTIO(R10)	ADDRESS OF TIOT	1272	02897019
	MVC	TIOCSTEP(X8),CHKEY	TIOT STEP NAME = CHKEY	1272	02898019
	LR	R1,R3	SET UP TO FREE READIN AREA	1272	02900019
	LA	R0,X256	LENGTH TO BE FREED	1272	02902019
		FREEMAIN R,LV=(0),A=(1)	FREE CORE	1272	02905019
IEFSD024	EQU	*		1272	02910019
	L	R2,LCTPARM3	SAVE GWT ADDRESS IN REG	1272	02930019
	CLI	LCTPARM3,X'8'	DID AN ERROR OCCUR IN IEF5	LS17	02964017
	BNE	IEF06223	NO ERROR	LS17	02965017
	OC	SCTMSADR(4),SCTMSADR	IS THIS A SPECIFIC REQUEST	CR17	02966017
	RE	SD162F3	NON-SPECIFIC	CR17	02967017
	MVI	LCTPARM3,X'4'	ERROR IN SPECIFIC REQUEST	CR17	02968017
SD162F3	OI	JCTJSTAT,INCMSTS	TURN ON JOB FAILED BIT	CR17	02969017
IEF06223	EQU	*		CR17	02970017
				1272	02970519
		* ALLOCATION AND OPEN WILL NEED TO REFERENCE THE JOB STEP'S		1272	02971019
		* WRITE TO PROGRAMER CONTROL BLOCK. THEREFORE THE SUBTASKS		1272	02971519
		* JSCB WILL BE IN THE INITIATOR'S TCB. THE JSCB FOR THE INIT		1272	02972019
		* WILL BE KEPT IN THE LCT		1272	02972519
	L	R4,LCTTCBAD	ADDRESS OF INIT TCB	1272	02973019
	L	R3,LCTJSCB	JOB STEP JSCB	1272	02973519
	L	R0,TCBJSCB(R4)	INIT'S JSCB	1272	02974019
	ST	R3,TCBJSCB(R4)	JOB STEP JSCB IN INIT TCB	1272	02974519
	ST	R0,LCTJSCB	INIT'S JSCB IN LCT	1272	02975019
	LA	R0,X104	GETMAIN SIZE	099	02976019
	LA	R1,TCORE	LOAD SUBPOOL	1272	02982019
	SLL	R1,X24	SUBPOOL IN HIGH ORDER BYTE	1272	02988019
	OR	R0,R1	SUBPOOL AND LENGTH	1272	02994019
				099	03000019
		FOLLOWING GETMAIN INCLUDES 8 BYTES FOR A		099	03006019
		PARAMETER AREA TO BE PASSED TO IEFDSOWR.		099	03012019
		SAVEAREA IS RE-USED WHEN LINKING TO SAME.		099	03018019
				099	03024019
	GETMAIN	R,LV=(0)	72 BYTE SAVE AREA 24 PARM AREA	1272	03030019
	LA	R13,X24(R1)	ADDRESS OF SAVE AREA		03040000
	L	R3,LCTJCTAD	JCT ADDRESS FROM LCT		03060000
	ST	R3,4(0,R1)	TO PARAMETER LIST		03080000
	ST	R8,0(0,R1)	LCT ADDRESS TO PAR LIST		03080919
	TM	JCTJSTAT,INCMSTS	HAS JOB BEEN PREVIOUSLY FAILED	099	03081819
	RO	DSOOK	YES-DON'T DO ANY DSO PROCESSING	099	03082719
	CLC	SCTPGMMN(EIGHT),RESTART	IS STEP RESTARTING FROM	M3150	03083619
			A CHECKPOINT?	099	

```

*      BNE      DSOOK                      NO- DON'T DO DSO RESTART CHECK      099 03084519
      TM      JCTRSW2,JCTDSOJB+JCTDSOCR IS DSO RESTART ENVIRONMENT 099 03085419
*      BZ      DSOOK                      CHECK NEEDED?                      099 03086319
      LINK    EP=IEFDSOCR,MF=(F,(1)) LINK TO IEFDSOCR                099 03087219
      LTR     R4,R15                      IS JOB TO RUN                  099 03088119
*      RNZ     FREEUP                      NO - FREE RESOURCES              M4012 03089119
      BZ      FREEUP                      YES - GO TO ALLOCATION            M4012 03090119
      XC      JCTDSOSM(N3),JCTDSOSM FORCE DSO MSG REBUILD            M4012 03091119
      EQU     *                          M2945 03091819
      TM      LCTOPSW1,LCTCANE TEST FOR CANCELLABLE                  099 03092119
      RZ      ALLOCATE                     BRANCH IF ALREADY CANCELLABLE 1272 03092419
      LR      R3,R1                       SAVE PARM REGISTER           1272 03092719
*      ENQ     MF=(F,SYSCMD)               ENQUEUE TO MAKE SYSTEM      1272 03093019
      LR      R1,R3                       TASK CANCELLABLE             1272 03093319
      NI      CHACT,CHCL                   RESTORE PARM REGISTER        1272 03093619
      EQU     *                          SET CANCELLABLE BIT           1272 03093919
      SR      R4,R4                       CLEAR REGISTER FOR SWITCH     1272 03094219
*      LINK    EP=IEFW21SD,MF=(F,(1))      099 03094519
      FREEUP  EQU     *                          099 03094819
      ST      R2,LCTPARM3                 099 03095119
      LR      R3,R15                     RESTORE SAVED GWT ADDRESS      099 03095419
      LR      R2,R1                       SAVE RETURN CODE            099 03095719
      TM      LCTOPSW1,LCTCANE TEST FOR CANCELABLE TASK              1272 03096019
      RZ      IEFSD005                     SAVE NEW PARAMETER LIST ADDRESS 03096319
      NI      CHACT,X255-CHCL              BRANCH IF CANCELABLE TASK      1272 03096619
      DEQ     MF=(F,SYSCMD)               RESET TO NON-CANCELLABLE        1272 03096919
      EQU     *                          RELEASE                         1272 03097219
      IEFSD005 EQU     *                          1272 03097519
*      *      1272 03097819
*      *      1272 03098119
*      *      099 03098419
*      *      099 03098719
*      *      099 03099019
*      *      099 03099319
*      *      099 03099619
*      *      099 03099919
*      *      099 03100219
*      *      099 03100519
*      *      099 03100819
*      *      099 03101119
*      *      099 03101419
*      *      099 03101719
*      *      099 03102019
*      *      099 03102319
*      *      099 03102619
*      *      099 03102919
*      *      099 03103219
*      *      099 03103519
*      *      099 03103819
*      *      099 03104119
*      *      099 03104419
*      *      099 03104719
*      *      099 03105019
*      *      099 03105319
*      *      099 03105619
*      *      099 03105919
*      *      099 03106219

      TM      JCTRSW2,JCTDSOJB ARE THERE ANY DSOGR'S FOR JOB
      BZ      DSOCOMP NO-BYPASS LINK
*
      CLC     SCTPGMMN,RSTRCK IS C/R DSDR PHASE BEING
*      BE      DSOCOMP YES - BYPASS LINK ALLOCATED
*
      OC      JCTDSOSM,JCTDSOSM WAS FIRST SMR HANDLED
      RNZ     FAILTEST YES- BYPASS SMR CONSTRUCTION
      GETMAIN R,LV=176 GET CORE FOR READ IN AREA
      ST      R1,N72(,R13) SAVE POINTER IN EXT PARM
      LA      R1,N72(,R13) GET POINTER TO EXT PARM AREA
      MVC     N4(N3,R1),JCTSMBAD MOVE IN TTR OF 1ST SMR
      MVI     N7(R1),NO ZERO OUT LOW ORDER BYTE
      ST      R1,OMPCL EXT PARM AREA TO OMPA
      MVI     OMPPOP,OMREAD INDICATE READ
      MVI     OMPNC,N1 OF ONE RECORD
      LR      R1,R9 R1 MUST POINT TO OMPA
      L       R15,IEFOMGRV GO TO Q MANAGER

```

	BALR	R14,R15		099	03106519
	LTR	R15,R15	WAS READ OK	099	03106819
	BNZ	SD062015	BRANCH NO THERE WAS ERROR	099	03107119
*				099	03107419
* N O T E -		R2,R3,R4,R5,R6,R7,R8	ARE BEING SAVED	099	03107719
*				099	03108019
	STM	R2,R8,NC(R13)	STORE SOME REGISTERS TO	099	03108319
*			ALLOW FOR WORK REGISTERS.	099	03108619
	CLI	QMPNP,QMWRTF	EAS WRITE DONE	099	03108919
	BE	SMBOUT	R-YES.	099	03109219
	L	R2,QMPCL	GET POINTER TO EXTBARM AREA	099	03109519
	L	R2,NO(R2)	GET POINTER TO SMB	099	03109819
	LA	R2,NO(R2)	CLEAR HI ORDER BYTE	099	03110119
	USING	SMBDSKAD,R2		099	03110419
	LA	R3,SMBHDR+1	GET OFFSET TO SMBMESSAGE AREA	099	03110719
	LA	R3,L'DSQMSG(,R3)	GET INCREMENT FOR R3	099	03111019
	L	R8,CVTPTR	LOCATION OF CVT	099	03111319
	USING	IEFCVT,R8		099	03111619
	L	R8,CVTMSER	ADDR OF MAST SCHED DATA AREA	099	03111919
	USING	BASE,R8	M/S BASE REG	099	03112219
*				099	03112519
	END	MF=(E,DSOCHNO)	END ON DSOCH CHAIN	099	03112819
	L	R8,RADSO	FIRST DSOCH ADDR	099	03113119
	USING	IEFDSOCH,R8	SET BASE FOR DSOCH	099	03113419
DSOCHLP	LA	R8,NO(R8)	CLEAR HIGH ORDER BYTE	099	03113719
	LTR	R8,R8	ANY MORE DSOCHS	099	03114019
	RZ	WRTOUT	NO- BRANCH TO WRITE SMB	099	03114319
	CLC	DSOKEY,JCTJSRNO	IS THIS A SELECTED DSOCH	099	03114619
	BE	MSGSETUP	YES GO TO SET UP MESSAGE	099	03114919
NXTDSO	L	R8,DSOCHT	GET A OF NEXT DSOCH	099	03115219
	R	DSOCHLP	+ GO TO LOOP	099	03115519
MSGSETUP	EQU	*		099	03115819
	TM	DSOIND1,DSOCH+DSOCHNO	TEST IF DSOCH NOT TO BE ALLC	099	03116119
	BNZ	NXTDSO	IF YES-GET NEXT DSOCH	099	03116419
	MVC	NO(N1,R3),DSOCLS	MOVE SYSOUT CLASS TO MESSAGE	099	03116719
	MVI	N1(R3),COMMA	AND FOLLOW IT BY COMMA	099	03117019
	LA	R3,N2(R3)	INCREMENT R3 BY 2	099	03117319
	B	NXTDSO	AND CONTINUE LOOP	099	03117619
WRTOUT	EQU	*		099	03117919
	DEQ	MF=(E,DSOCHNO)	DEQ ON DSOCH CHAIN	099	03118219
	LA	R4,SMBHDR+1	SET UP TO WRITE OUT MESSAGE	099	03118519
	LA	R4,L'DSQMSG(,R4)	SET R4 TO MATCH WITH R3	099	03118819
	CR	R3,R4	EQUAL MEANS NO MESSAGE	099	03119119
	BE	SMBOUT	LB NOT TO WRITE MESSAGE	099	03119419
	MVI	NO(R3),NO	ZERO NXT AVAILBLE BYTE INCASF	M2945	03119719
*			MSG IS BEING REBUILT FROM A RESTART	M2945	03120019
	HCTR	R3,R0	DECREMENT COUNTER BY 1	099	03126219
	MVI	NO(R3),PERIOD	OVERLAY LAST COMMA WITH PERIOD	099	03126319
	LA	R4,SMBHDR	GET SMB HEADER LENGTH	099	03126419
	SR	R3,R4	CONVERT R3 TO MSGLENGTH	099	03126519
	STC	R3,SMBMSGLEN	+ STORE IT IN SMBMSGLENGTH	099	03126619
	LA	R3,N1(,R3)	INCREMENT BY 1 TP GET NEXT AVAIL BT	099	03126719
	STH	R3,SMBNAHPT	INDICATE NEXT AVAILBLE BYTE	099	03126819
	LA	R3,SMBHDR+1	GET OFFSET TO SMBMESSAGE AREA	099	03126919
	MVC	NO(L'DSQMSG,R3),DSQMSG	MOVE IN MESSAGE HEADER	099	03127019
	MVI	QMPNP,QMWRTF	INDICATE WRITE OPERATION	099	03127119
*				099	03127219

```

* N D T F - R2,R3,R4,R5,R6,R7,R8 ARE BEING RESTORED 099 03127319
* 099 03127419
* LM R2,R8,NC(R13) LOAD THE REGS 099 03127519
* OMGRMV 099 03127619
SMBOUT EQU * 099 03127719
MVC JCTDSUSM(N3),JCTSMHAD HAVE 1ST SMB POINTED TO TO 099 03127819
* INDICATE SMB WAS HANDLED. 099 03127919
* L R1,OMPCL GET FPA 099 03128019
* L R1,N0(,R1) GET CORE ADR 099 03128119
* FREEMAIN R,LV=176,A=(1) 099 03128219
* 099 03128319
* N D T F - R2,R3,R4,R5,R6,R7,R8 ARE BEING RESTORED 099 03128419
* 099 03128519
* LM R2,R8,NC(R13) LOAD THE REGS 099 03128619
* DROP R2 099 03128719
* 099 03128819
* RE-SET LCT BASE 099 03128919
* USING IEFLCT,R8 099 03129019
* 099 03129119
* FAILTEST EQU * 099 03129219
* TM JCTJSTAT,JCTJOBFL+JCTSTPFL HAS JOB/STEP FAILED M3164 03129319
* RNZ DSOCNMP YES-BYPASS LINK 099 03129519
* 099 03129619
* LTR R3,R3 ANY ALLOCATION ERRORS 099 03129719
* RNZ DSOCNMP YES- BYPASS LINK 099 03129819
* 099 03129919
* TM SCTSTAT2,SCTDSOCL DID IEFDSONL IN ALLOCATION 099 03130019
* FIND WORK FOR IEFDSONR 099 03130119
* 099 03130219
* 099 03130319
* 099 03130419
* 099 03130519
* LA R1,N72(,R13) GET 8 BYTE PARM AREA 099 03130619
* ST R8,N0(,R1) LCT ADR 099 03130719
* MVC N4(N4,R1),IEFQMGRV QMGR ADR 099 03130819
* 099 03130919
* LINK EP=IEFDSONR,MF=(E,(1)) 099 03131019
* 099 03131119
* 099 03131219
* 099 03131319
* * * RETURN CODES AND SETTING R3 099 03131419
* 1. LINK TO IEFDSONR ETC IS BYPASSED IF ALLOCATION/IEFDSONR 099 03131519
* RETURNS A NON-ZERO RETURN CODE. SEE ABOVE CODE. 099 03131619
* 2. IF ZERO RETURN CODE FROM ALLOCATION THEN - 099 03131719
* A. IF IEFDSONR ENCOUNTERS A JOB I/O ERROR A 099 03131819
* RETURN CODE OF 12 WILL BE SENT BACK INDICATING 099 03131919
* JOB SHOULD BE ABENDED WITH A 'JOB'. R3 WILL 099 03132019
* BE MADE NEG IN THIS CASE SO THIS PROG CAN 099 03132119
* DO FREEMAIN ETC BEFORE ABEND. 099 03132219
* B. IF IEFDSONR ENCOUNTERS A PERMANENT I/O ERROR 099 03132319
* WHILE WRITING JOB SEPARATORS OR SMB'S TO A 099 03132419
* DIRECT DEVICE THEN A RETURN CODE OF 4 WILL 099 03132519
* BE SENT BACK INDICATING THE JOB IS TO BE FAILED. 099 03132619
* R3 WILL BE SET PLUS AND WILL CAUSE THIS PROG TO 099 03132719
* TAKE THE SAME ROUTE THAT IS TAKEN WHEN ALLOCATION 099 03132819
* RETURNS A NON-ZERO RETURN CODE. 099 03132919
* 099 03133019
* LNR R3,R15 PRIME FOR POSSIBLE ABEND AND SAVE -

```

```

*                                     NOTE - A ZERO VALUE REMAINS UNCHANGED 099 03133119
*                                     AND PLUS                                099 03133219
LTR  R15,R15  ANY ERRORS 099 03133319
BZ   DSOCOMP NO-CONT MAINLINE 099 03133419
*
LA   R1,NR    YES-SET-UP TO TEST COND CODE 099 03133519
CR   R15,R1   TEST IF ABOVE/BELOW 8 099 03133619
BH   DSOCOMP  HIGH-COND CODE 12 = JOB I/O ERROR 099 03133719
*                                     KEEP R3 NEG FOR ABEND LATER 099 03133819
* COND CODE 8 NOT IN USE 099 03133919
LR   R3,R15   LOW-COND CODE 4 = I/O ERROR ON I/O 099 03134019
*                                     DEVICE WHILE WRITING SMPS, ETC. PRIME 099 03134119
*                                     R3 TO BYPASS EXECUTION OF P/P. 099 03134219
*                                     R3 NOW POSITIVE 099 03134319
OI   JCTJSTAT,INCMSTS  TURN ON JOB FAIL BIT 099 03134419
*                                     099 03134519
DSOCOMP EQU *      CONT MAINLINE 099 03134619
NI   SCTSTAT2,N255-SCTDSOCL  RESET FOR TERMINATION 099 03134719
*                                     099 03134819
LR   R1,R13   RESTORE ARSA POINTER. CR17 03141017
FREEMAIN R, LV=80, A=(1), SP=TCORE  FREE SAVE AND PARM AREA 099 03141219
*                                     099 03141419
LTR  R3,R3     SHOULD ATTACH BE BYPASSED. CR17 03141619
BZ   IEFSD010  NO. CR17 03141819
L    R1,LCTTCBAD  ADDRESS OF INIT'S TCB 1272 03142019
L    R14,TCBJSCB(R1)  JOB STEP JSCB 1272 03142219
L    R0,LCTJSCB  INIT'S JSCB 1272 03142419
ST   R14,LCTJSCB  JOB STEP JSCB IN LCT 1272 03142619
ST   R0,TCBJSCB(R1)  INIT'S JSCB IN INIT'S TCB 1272 03142819
L    R14,LCTEXIT  ADDRESS OF EXIT LIST 1272 03143019
LA   R14,XO(R14)  ZERO HIGH ORDER BYTE 1272 03143219
* ***** TEST FOR EXIT LIST TO BE REMOVED IN R20 ***** 1272 03143419
LTR  R14,R14  TEST FOR EXIT LIST 1272 03143619
BZ   FREEGWT  BRANCH NO EXIT LIST 1272 03143819
MVI  IELRTNCD(R14),RTNCD4  SET RETURN CODE TO 4 1272 03144019
FREEGWT EQU * 1272 03144219
LA   R14,TCORE  YES. LOAD SUBPOOL NO. CR17 03145017
SLL  R14,24  CR17 03146017
LA   R0,GPMINPAR+4-GETPTWT  LOAD LENGTH OF GETPART WK 1517 03146119
OR   R0,R14  LOAD SUBPOOL + LENGTH 1517 03146219
L    R1,LCTPARM3  LOAD ADDRESS OF GWT 1272 03146319
FREEMAIN R, LV=(0), A=(1)  FREE ITS CORE 1517 03146419
LTR  R3,R3  TEST REG STATUS 099 03146519
BH   SD062015  NEG REG - JOB I/O ERROR FROM 099 03146619
*                                     IEFDSUWR, GO TO ABEND 099 03146719
*                                     099 03146819
LTR  R4,R4  IS THIS ABNORMAL RETURN FROM 099 03146919
*                                     IEFDSOCR? 099 03147019
BZ   T0165  NO-GO TO IEFSD165 099 03147119
USING IEZJSCB,R1  JSCB ADDRESSABILITY 099 03147219
L    R1,LCTJSCB  ADDRESS OF JOB STEP JSCB 099 03147319
L    R14,JSCBWT  ADDRESS OF WTP CONTROL BLK 099 03147419
FREEJSCB SYSTEM=MVT  FREE JSCB 099 03147519
OROP R1 099 03147619
LR   R1,R14  ADDRESS OF WTPCH IN R1 099 03147719
LA   R14,TCORE  SUBPOOL 253 099 03147819
SLL  R14,X24  SUBPOOL IN HIGH ORDER BYTE 099 03147919
LA   R0,WTPCHSIZ  SIZE OF WTPCH

```

BR	R0,R14	SUBPOOL AND SIZE IN REG 0	099	03148019
FREE	MAIN R,LV=(0),A=(1)	FREE WTPCH	099	03148119
I	R15,IEFSD161	IEFSD161 LINKAGE	099	03149819
		RETURN TO SELECT ANOTHER JOB	099	03150019
LR	R1,R8	ICT ADR	0012	03150219
RR	R15	XCTL TO IEFSD161 VIA LINKAGE	099	03150419
		MODULE	099	03150619
IEFSD161	FOU *		099	03150819
I	R15,IEFSD161V		CR17	03151017
LR	R1,R2	RESTORE PARAM LIST PTR.	CR17	03152017
RR	R15	GO TO IEFSD0065	CR17	03153017
*****			CR17	03193017
*****			CR17	03233017
* BUILD PARAM LIST FOR IEFSD161 TO WRITE OUT TIO & GET REG SAVE A			CR17	03273017
*****			CR17	03313017
*****			CR17	03353017
IEFSD010	FOU *		CR17	03393017
CLI	CHVCO,SPKORAM	CHECK FOR STARTED PROGRAM	1272	03394019
BNE	IEFSD011	BRANCH NOT A STARTED TASK	1272	03395019
L	R10,12(R2)	LOAD TIO LIST PTR	1272	03396019
I	R10,0(R10)	PTR TO TIO	1272	03397019
TIOISRC	CLC TIOIDNM(X),IEFDR	IS DDNAME IEFDR	1272	03398019
BE	UCRADR	BRANCH YES	1272	03399019
SR	R3,R3	ZERO REGISTER	1272	03400019
IC	R3,TIOELNGH	LENGTH OF TIO ENTRY	1272	03401019
LTR	R3,R3	TEST FOR END OF TIO	1272	03402019
RZ	NORDER	BRANCH END OF TIO	1272	03403019
AR	R10,R3	ADD ENTRY LENGTH TO BASE	1272	03404019
B	TIOISRC	TEST NEXT DD ENTRY	1272	03405019
UCRADR	L R3,TIOESTH	LOAD UCR POINTER	1272	03406019
LA	R3,X0(R3)	ZERO HIGH ORDER BYTE	1272	03407019
LTR	R3,R3	TEST FOR ZERO UCR REQUEST FOR	1272	03408019
		DUMMY DEVICE	1272	03409019
RZ	NORDER	BRANCH IF ZERO	1272	03410019
USING	UCR,R3	ADDRESSABILITY FOR UCR	1272	03411019
CLI	UCRID,MAINUCH	IS IT A MAIN UCR	1272	03412019
BNE	MINRUCH	BRANCH IF DATACELL	1272	03413019
MOVE	NAME MVC CHUNIT(X3),UCRNAME	MOVE UCR NAME INTO CSCR	1272	03414019
TM	LCTINTSW,LCTSTART	WAS TASK NAME ON COMMAND	1272	03415019
RZ	IEFSD011	BRANCH NAME SPECIFIED	1272	03416019
MVI	CHKEY,BLANK	MOVE IN BLANK	1272	03417019
MVC	CHKEY+X1(X7),CHKEY	BLANK CHKEY FIELD	1272	03418019
MVC	CHKEY(X3),UCRNAME	MOVE UCR NAME INTO CSCR	1272	03419019
B	TIOKEY		1272	03420019
DRIP	R3		1272	03421019
MINRUCH	LH R4,DCFLBHR-DATACELL(R3)	PICK UP BIN NUMBER	1272	03422019
LA	R4,X1(R4)	INITIALIZE FOR LOOP	1272	03423019
CHKSUR	HCT R4,SURUCLN	SUBTRACT BRANCH ON ZERO	1272	03424019
S	R3,MAINLGTH	SUBTRACT MAIN UCR LENGTH	1272	03425019
B	MOVE		1272	03426019
SURUCLN	S R3,SURLGTH	SUBTRACT SUB UCR LENGTH	1272	03427019
B	CHKSUR	CHECK FOR MORE SURCELLS	1272	03428019
NORDER	TM LCTINTSW,LCTSTART	WAS TASK NAME ON COMMAND	1272	03429019
RZ	IEFSD011	BRANCH NAME SPECIFIED	1272	03430019
MVC	CHKEY(X8),CHCLS	MOVE PROC NAME INTO ID FIELD	1272	03431019
TIOKEY	L R10,LCTTCHAD	ADDRESS OF INIT'S TCH	1272	03432019
L	R10,TCHTIO(R10)	TIO ADDRESS IN TCH	1272	03434019

ADDRESS	OPERATION	OPERANDS	COMMENT	PC	PC+1
0000	MVC	TIOCTSTEP(XR),CHKKEY	TIOCT STEP NAME = CHKKEY	1272	03435019
0001	L	R10,12(R2)	LOAD TIOCT LIST POINTER	1272	03435219
0002	L	R10,X0(R10)	PIR TO TIOCT	1272	03435419
0003	MVC	TIOCTSTEP(XR),CHKKEY	TIOCT STEP NAME = CHKKEY	1272	03435619
0004	EQU	*		1272	03436019
0005					03436000
0006	GETMAIN	R,LV=96,SP=TCORF	GM 24-BYTE PARAM LIST & 72-	CR17	03436317
0007	LA	R13,24(R1)	REG SAVE AREA.	CR17	03436417
0008	ST	R9,0(R1)	1ST WORD = OMPA PTR.	CR17	03436517
0009	XC	4(4,R1),4(R1)	2ND WORD = ZEROS.	CR17	03436617
0010	L	R7,12(R2)	GET TIOCT LIST ADDR.	CR17	03436617
0011	MVC	R(R,R1),0(R7)	3RD & 4TH WORDS-TIOCT ADDR+	CR17	03436717
0012	XC	16(R,R1),16(R1)	5TH & 6TH WORDS = ZEROS.	CR17	03436817
0013	LA	R7,REGSAVE+72	REMOTE PARAM AREA.	CR17	03436817
0014	ST	R7,OMPCOL		CR17	03436917
0015	MVI	OMPCOL,3	INDICATE WRITE	CR17	03437017
0016	LR	R7,R1	SAVE PARAM LIST & REG AREA	CR17	03437117
0017	L	R15,SD514RTF		CR17	03437217
0018	HALR	R14,R15	GO TO IEFS0514	CR17	03437217
0019	LTR	R15,R15	ANY I/O ERROR.	CR17	03437317
0020	RN7	SD062015	YES.	CR17	03437417
0021	LR	R1,R7	NO.	CR17	03437517
0022	L	R7,4(R2)	RESTORE JCT BASE REG.	CR17	03437617
0023	MVC	JCTSTIOT(4),4(R1)	CONTROL TABLE ITR TO JCT.	CR17	03437617
0024	FREEFMAIN	R,LV=96,A=(1),SP=TCORF	FREE PARAM LIST & REG AR	CR17	03437717
0025					03438000
0026					03500000
0027					03520017
0028					03540017
0029					03560000
0030					03580000
0031					03600000
0032					03640000
0033					03660000
0034					03662017
0035					03664017
0036					03664418
0037					03664818
0038					03665218
0039					03665618
0040					03666018
0041					03666418
0042					03666818
0043					03667218
0044					03667618
0045					03668018
0046					03668418
0047					03668818
0048					03669218
0049					03669318
0050					03669418
0051					03669618
0052					03670018
0053					03670418
0054					03670818
0055					03671218
0056					03671618
0057					03671618
0058					03671618
0059					03671618
0060					03671618
0061					03671618
0062					03671618
0063					03671618
0064					03671618
0065					03671618
0066					03671618
0067					03671618
0068					03671618
0069					

	CL	R14,LCTTMWRK+DFC4	J. TIME LEFT GT S. TIME	SMF1 03672018
			NEEDED	SMF1 03672418
	HNL	IEFSD015	YES- USE STEP TIME FROM	SMF1 03672818
			SCT	SMF1 03673218
*				
LONG JOB	EQI	*	*	03673618
	ST	R14,DFC0(R1)	NO - USE JOB TIME LEFT	SMF1 03674018
	MVI	LCTTMWRK+DFC4,HEX80	IND THAT TIME IS FOR JOB	SMF1 03674418
IEFSD015	EQI	*		CR17 03676017
	LA	R9,QMGR1	A MGR PAR AREA	03680000
	LA	R13,REGSAVE	REGISTER SAVE AREA	03700000
	LA	R1,REGSAVE+72	REMOTE PARAMETER AREA	03720000
	ST	R1,OMPCL	ADDRESS OF REMOTE PARAMETER LST	03740000
	MVI	OMPPOP,3	WRITE INDICATOR	03760000
	MVI	OMPCL,2	INDICATE MULTIPLE OPERATION	03780000
	L	R7,4(0,R2)	LOAD JCT BASE REGISTER	03800000
	ST	R7,0(0,R1)	JCT CORE ADDRESS TO QUEUE MGR	03820000
			REMOTE PARAMETER AREA	03840000
	ST	R6,8(0,R1)	SCT CORE ADDRESS TO QUEUE MGR	03860000
			REMOTE PARAMETER AREA	03880000
	NI	JCTRSW2,X'C7'	KNOCK OFF BITS 2,3 AND 4.	CR17 03886017
	LA	R4,MASK	LOAD MASK TO GET C/R BITS	CR17 03892017
MASK	EQI	X'3H'	LOAD MASK TO GET C/R BITS	CR17 03898017
	SR	R10,R10	ZERO REGISTER.	CR17 03904017
	IC	R10,SCTSTAT	LOAD STATUS INDICATORS	CR17 03910017
	NR	R10,R4	GET RID OF ALL BITS BUT C/R	CR17 03916017
	EX	R10,SETHITS	EXEC INSTRUCTION TO SET BITS	CR17 03922017
	MVC	4(4,R1),JCTDSKAD	JCT TTR TO OMPA.	CR17 03932017
	MVC	12(4,R1),SCTDISKA	SCT TTR TO OMPA.	CR17 03972017
	MVI	15(R1),X'00'	SCT TTR	CR17 04012017
	LR	R1,R9	ADDRESS OF O MGR PARAMETER AREA	04060000
	L	R15,IEFQMGRV	ADDRESS OF O MGR	04080000
	HALR	R14,R15	HAVE THE TABLES WRITTEN	04100000
	MVI	OMPCL,0	ZERO OUT NUMBER OF OPERATIONS	04120000
	R	SD062010(R15)	CHECK RETURN CODE	04140000
SD062010	R	SD062020	RETURN CODE OK	04160000
	RC	0,SD062010	NO WORK, IMPOSSIBLE ON WRITE	04180000
	RC	0,SD062010	NO SPACE - IMPOSSIBLE, THESE	04200000
			TABLES ARE BEING WRITTEN TO	04220000
			ASSIGNED SPACE	04240000
*				04260000
			I/O ERROR	04280000
			CODE TO PROCESS THIS CONDITION	04300017
SD062015	LA	R1,X'0H0'	SHOULD BE PLACED HERE.	CR17 04300017
	SLA	R1,12	TEMPORARILY, ABEND	04320000
	ABEND	(1),DUMP		04340000
				CR17 04340717
SD062020	XC	LCTPARM4(X4),LCTPARM4	ZERO PARM4	1272 04341719
	L	R3,LCTPARM3	SET UP ADDRESSABILITY FOR	1272 04342719
	USING	GETPTWT,B3	GET PART WORK TABLE	1272 04343719
	L	R14,LCT CHAD	GET INITIATOR'S TCH.	
	L	R1,TCHTIN(R14)	HUMP TO THE TINT.	
	CLI	0(14),C'0'	FIXED INITIATOR?	
	BE	SD062040	YES - BRANCH TO KEEP CURRENT REGION.	
	L	R4,GPSIZEPP	REGION SIZE FOR P/P	1272 04344719
	LTR	R4,R4	TEST FOR ZERO SIZE	1272 04345719
	BE	SD062040	BRANCH - KEEP REGION	1272 04346719
	MVI	GPSURP,P00L246	REPLACE PARTITION SURPOOL	1272 04347719
	TM	LCTINTSW,LCTIHIER	TEST FOR HIERARCHY	1272 04348719

BN	SD062022	BRANCH HIERARCHY ONE	1272 04349719
MVC	GPADDH0+X1(X3),GPADDH0R+X1	0 OR SPECIFIC ADDR	1272 04350719
MVC	GPSIZEH0+X1(X3),GPSIZEPP+X1	MOVE REGION SIZE	1272 04351719
R	SD062040		1272 04352719
SD062022	MVC	GPSIZEH1+X1(X3),GPSIZEPP+X1	MOVE REGION SIZE
R	SD062040		1272 04354719
SD062040	EQU	*	LS17 04360017
LA	R14,TCORE	LOAD SUPPOOL NO	LS17 04360317
SLL	R14,24		LS17 04360617
LA	R0,GPMINPAR+4-GETPTWT	LOAD LENGTH OF GETPART WK	LS17 04360917
OR	R0,R14	LOAD SUPPOOL + LENGTH	LS17 04361217
CLI	GPSURP,246	DO WE REPLACE THE PARTITION	LS17 04361517
BE	SD162035	YES	LS17 04361817
L	R1,LCTPARM3	ADDRESS OF GETPART WORK 1A	LS17 04362117
FREEMAIN	R,LV=(0),A=(1)	FREE ITS CORE	LS17 04362417
XC	LCTPARM3(X4),LCTPARM3	ZERO GWT POINTER	1272 04362519
R	SD062030		LS17 04362717
SD162035	ST	R0,LCTPARM4	SET UP FOR FREEMAIN IN IFF
SD062030	EQU	*	CR17 04363017
DRDP	R3		CR17 04364517
USING	IHANDR,R3		1272 04364619
*****			1272 04364719
*****			CR17 04364817
*****			CR17 04365117
* GET CORE FOR THE USER PARAM LIST,BUILD PARAM LIST,SET UP THE PRO			CR17 04365417
* INTERFACE AND SAVE THE ADDR IN REGISTER 4.			CR17 04365717
*****			CR17 04366017
*****			CR17 04366317
LH	R3,SCTSEXEC	GET CORE FOR THE NO. OF USE	CR17 04366617
LA	R0,R(0,R3)	PLUS R FOR GETMAIN	18428 04366917
LA	R14,TCORE		CR17 04367217
SLL	R14,24		CR17 04367517
OR	R0,R14		CR17 04367817
GETMAIN	R,LV=(0)	GET USER PARAMS CORE.	CR17 04368117
LR	R4,R1	SAVE ADDR IN REG 4.	CR17 04368417
ST	R3,4(0,R1)	LENGTH TO LIST	18428 04368717
LA	R0,4(R1)	ADDR OF LENGTH FIELD IN LI	CR17 04369017
ST	R0,0(R1)		CR17 04369317
OI	0(R1),128	SET HIGH ORDER BIT TO 1.	CR17 04369617
BCTR	R3,0		CR17 04369917
LTR	R3,R3		CR17 04370217
BM	SD06220		CR17 04370517
LA	R0,176		CR17 04370817
GETMAIN	R,LV=(0)	GEYMAIN FOR READ BUFFER	CR17 04371117
LR	R5,R1	SAVE FOR FREEMAIN	CR17 04371417
LA	R9,QMGR1	Q MNGR PARAMETER AREA	CR17 04371717
LA	R13,REGSAVE	REGISTER SAVE AREA	CR17 04372017
LA	R1,REGSAVE+72	REMOTE PARAMETER AREA	CR17 04372317
ST	R1,QMPCL	STORE ADDR OF REMOTE PARAM	CR17 04372617
MVI	QMPDP,4	READ INDICATOR	CR17 04372917
MVI	QMPNC,X'01'	READ ONE RECORD	CR17 04373217
ST	R5,0(0,R1)	STORE BUFFER ADDRESS	CR17 04373517
MVC	4(4,R1),SCTXHTTR	MOVE TTR	CR17 04373817
LR	R1,R9	ADDRESS OF Q MGR PARAMETER	CR17 04374117
L	R15,IEFOMGRV	ADDRESS OF Q MNGR	CR17 04374417
RALR	R14,R15	READ THE RECORD.	CR17 04374717
EX	R3,IEFMOVE1		CR17 04375017
LR	R1,R5	RESTORE POINTER	CR17 04375317

LA	R0,176		CR17	04375617
FREEFMAIN R, LV=(0), A=(1)			CR17	04375917
SD06220 EQU *			CR17	04376217
L	R3,DMCAP	PTR TO TRACK STACK INFO	11404	04380914
L	R3,0(R3)	PTR TO STACK	11404	04381814
LA	R3,0(R3)	ZERO HI ORDER BYTE	011404	04382714
LTR	R3,R3	TEST IF STACK EXISTS	11404	04383614
BE	SD06225	NO - BRANCH	11404	04384514
*				04385414
LR	R13,R3	PTR TO SAVE AREA IN STACK	11404	04386314
XC	OMPOR(1),OMPOR	ZERO OP CODE FOR PURGE RTN	11404	04387214
LR	R1,R9	PTR TO OMGR PARM AREA	11404	04388114
L	R15,SD06250	ADDR OF STACK PURGE PGM	11404	04389014
HALR	R14,R15	GO TO PURGE+FREE STACK	11404	04389914
*				04390814
LR	R3,R15	SAVE RETURN CODE	11404	04391714
LR	R1,R13	PTR TO RETURNED SAVE AREA	11404	04392614
FREEFMAIN R, LV=72, A=(1)		FREE AREA	11404	04393514
LA	R13,REGSAVE	RESTORE SD062 SAVE AREA	11404	04394414
LTR	R3,R3	TEST RETURN CODE	11404	04395314
BE	SD06225	NORMAL RETURN	11404	04396214
R	SD062010+12	ERROR RETURN	PTM663	04397116
*				04398014
SD06225 EQU *			11404	04398914
*			*	04400000
*****				04420000
* CORE IS GOTTEN FOR A PARAMETER LIST WHICH IS PASSED BETWEEN			1270	04440019
* INITIATOR MODULES AND FOR THE ATTACH PARAMETER LIST WHICH IS			1270	04460019
* INITIALIZED IN IFFSD103 AND USED BY IFFSD263 IN THE LINK			1270	04480019
* PACK AREA TO ATTACH THE SUBTASK.			1270	04500019
*****				04560000
LA	R0,LISTSIZE	SIZE OF PARAMETER LISTS	1270	04600019
LA	R13,LISTSIZE-1	SIZE OF PARM AREA LESS ONE	1270	04640019
LA	R1,TCORF	SUBPOOL 253	1270	04720019
SLL	R1,X24	SP IN HIGH ORDER BYTE	1270	04760019
OR	R0,R1	SP AND LENGTH IN REG 0	1270	04800019
GETMAIN R, LV=(0)			1270	04880019
EX	R13,CLEARCOR	ZERO GOTTEN CORE	1270	04920019
LR	R11,R1	EST ADDRESSABILITY OF PARM ARE	1270	04960019
ST	R4,IFFUSADD	ADDRESS OF USER PARMS	1272	04980019
LA	R0,IFFPSIZE(R11)	ADDRESS OF ATTACH PARM LIST	1270	05000019
ST	R0,IFFREMLS	ADDR OF ATTACH LIST IN PARM LT	1270	05040019
LA	R13,X1(R13)	LENGTH OF GOTTEN CORE	1270	05060019
STC	R13,IFFREMLS	STORE LENGTH OF GOTTEN CORE	1270	05080019
L	R3,12(0,R2)	TIOCLIST POINTER		05120000
ST	R3,IFFTIOA	PLACE IN PARAMETER LIST		05140000
L	R5,LCTIDRTY	GET CSCH BASE REGISTER		05160000
L	R4,LCTTCHAD	ADDRESS OF INIT'S TCB	1272	05180019
ST	R4,IFFTCHAD	PLACE IN PARAMETER LIST		05200000
XC	0(4,R3),TIOTPONT(R4)	THIS SERIES OF EXCLUSIVE OR'S		05220000
XC	TIOTPONT(4,R4),0(R3)	RESULTS IN PUTTING THE JOB STEP		05240000
XC	0(4,R3),TIOTPONT(R4)	TIOI PTR IN THE INITIATOR'S TCB		05260000
*		AND THE INITIATOR TIOI PTR IN		05280000
*		THE TIOI LIST. THIS IS DONE TO		05300000
*		SAVE THE PTR TO THE INITIATOR		05320000
*		TIOI WHILE THE PTR TO THE JOB		05340000
*		STEP TIOI IS PLACED IN THE		05360000

```

*          INITIATOR'S TCM TO BE REFER- 05380000
*          ENCED BY THE OPEN ROUTINE    05400000
*          ST      R8,IEFJCLAD          PLACE LCT POINTER INTO PAK LST 05420000
***** 05440000
*          05460000
* THE FOLLOWING SECTION OF CODE GETS CORE FOR AND OPENS THE JOBLIB
*          STEPLIB, AND/OR FETCH DCB'S AS NECESSARY *0042 05500017
*          05520000
***** 05540000
*          TM      SCTSTAT,X'04'        IS PROTECT KEY TO BE ZERO CR17 05545017
*          HQ      BYPASS                YES, IGNORE JOBLIB AND FETCH A16531 05550019
*          DCBS                                CR17 05555017
*          L        R10,TIOTPOINT(R4)    GET TIOT ADDRESS 05560000
***** 05561019
*          SET UP STAF ENVIRONMENT      * 05562019
***** 05563019
*          LA      R0,STAE LNTH          GET LNTH OF ARFA A16531 05564018
*          LA      R14,TCORE             SUBPOOL NUMBER A A16531 05565018
*          SLL     R14,24                MOVE TO HI ORDER BYTE A16531 05566018
*          OR      R0,R14                LNTH REG A16531 05567018
*          GETMAIN R,LV=(0)              05568018
*          LR      R9,R1                SAVE PARM ADDR A16531 05569018
*          LA      R0,AGAIN              @ STAF EXIT ROUTINE A29871 05571019
*          ST      R0,0(R9)              SAVE EXIT RTN @ A29871 05573019
*          ST      R9,4(R9)              SAVE PARM ARFA @ A29871 05575019
*          STM     R4,R11,16(R9)         SAVE REGISTER'S A29871 05577019
*          SR      R0,R0                ZERO A REGISTER 05580000
* THIS IS THE MACRO WHICH SETS UP THE ENVIRONMENT FOR HANDLING A16531 05580118
* ARENOS AFTER OPEN STMTS. AN ADDRESS OF THE EXIT ROUTINE IS A16531 05580218
* GIVEN HERE TO ROUTE CONTROL THROUGH THE RETRY ROUTINE. A PARM A16531 05580318
* AREA ADDRESS IS GIVRN WHICH HAS SAVED REGS FOR THE RETRY A16531 05580418
*          STAE    MF=(E,(R9))          ISSUE STAE MACRO A29871 05580719
*          TM      SCTSTAT2,SCTSTPLB     IS STEPLIB PRESENT 17052 05581017
*          HZ      IEFJOBLR             NO, CHECK JOBLIB 17052 05582017
IEFSD032 DS      OH                    17052 05583017
*          LA      R0,IEFDCBN-IEFDCHS    DCH LENGTH 17052 05584017
*          LR      R4,R0                SUBTRACT ONE 17052 05585017
*          BCTR    R4,0                 TO EXECUTE MOVE INSTRUCTION 17052 05586017
*          LA      R14,TCORE             SUBPOOL NUMBER 17052 05587017
*          SLL     R14,24                TO HIGH-ORDER BYTE 17052 05588017
*          OR      R0,R14                OF LENGTH FIELD 17052 05589017
*          05590017
*          GETMAIN R,LV=(0)              GET DCB CORE 17052 05591017
*          05592017
*          LR      R3,R1                ADDR TO DCB BASE REGISTER 17052 05593017
*          EX      R4,MOVEDCH            MOVE REMOTE DCB TO GOTTEN CORE 17052 05594017
*          MVC     DCHDDNAM,STEPLIB      PUT STEPLIB DDNAME INTO DCB17052 05595017
*          B       IEFJSOPE              OPEN STEPLIB DCB 17052 05596017
IEFJOBLR TM      JCTJSTAT,JOBLIB        WAS JOBLIB SPECIFIED IN 17052 05606017
*          JOB 05620000
*          HZ      IEFSD030              NO, CHECK FOR FETCH 05640000
*          LA      R0,IEFDCBN-IEFDCHS    NO. LENGTH OF DCB. CR17 05680017
*          LR      R4,R0                LENGTH OF DCB - 1 05720000
*          BCTR    R4,0                 TO EXECUTE 05740000
*          LA      R14,TCORE             SUBPOOL NUMBER 05760000
*          SLL     R14,24                TO HIGH ORDER BYTE 05780000
*          OR      R0,R14                OF LENGTH FIELD 05800000

```

GETMAIN R, LV=(0)	GETMAIN FOR DCB	05820000
LR R3, R1	ADDRESS OF REMOTE DCB	05840000
EX R4, MOVEDCB	MOVE DCB TO GOTTEN CORE	05860000
MVC DCBDDNAM(R), TIOFDDNM	PUT THE DDNAME IN THE DCB	05880000
* THE EXIT LIST CODE USED FOR 213 ARFENDS HAS BEEN REMOVED FROM		
* THIS MODULE. IT IS NO LONGER NEEDED NOW THAT A STATE ENVIRON-		
* MENT HAS BEEN ESTABLISHED THE OPEN STMT IS CHANGED SUCH THAT		
* IT NO LONGER ADDRESSES THE EXIT LIST		
IEFJSOPF LR R1, R2	ALLOCATE PARMLIST	A16531 05890018
STM R2, R3, 8(R9)		A16531 05900018
MVI O(R1), X'80'	INDICATE END OF LIST	A16531 05910018
OPEN ((3), INPUT), MF=(E, (1))	OPEN THE DCB	A16531 05920018
TM DCBDFLGS, X'10'	OPEN SUCCESSFUL	A16531 05930018
RZ IEFSD033A	BRANCH NO	A29871 05960019
ST R3, LCTJDBLB	SAVE JOBLIB DCB POINTER	06000000
TM JCTJSTAT, JOBLIB	WAS JOBLIB SPECIFIED	06020000
RZ IEFSD030		A16531 06040018
IEFSD023 SR R0, R0	ZERO A REGISTER	A16531 06060018
IEFSD025 IC R0, TIOFLNGH		06080000
LTR R0, R0	ANY MORE ENTRIES ?	17052 06086017
RZ IEFSD035	NO, DON'T OPEN FETCH	17052 06092017
AR R10, R0	CHECK FOR	06100000
CLI TIOFDDNM, X'40'	CONCATENATION	06120000
RE IEFSD025	IF CONCATENATED, GET NEXT ENTRY	06140000
IEFSD030 TM SCTSTYPE, SCTGSTP	WAS FETCH SPECIFIED ?	06160000
RZ IEFSD035	NO, DON'T OPEN FETCH	06180000
LA R0, IEFDCBN-IEFDCBS	DCB LENGTH	06200000
LR R4, R0	DECREMENT BY ONE	06220000
HCTR R4, 0	FOR EXECUTING MOVE	17052 06250017
LA R14, TCORE	SUBPOOL NUMBER	06266017
SLL R14, 24	TO HIGH ORDER BYTE	06272017
OR R0, R14	OF LENGTH FIELD	06278017
GETMAIN R, LV=(0)	GET DCB CORE	06284017
LR R3, R1	PLACE ADDRESS IN R3	06290017
STM R2, R3, 8(R9)		06296017
EX R4, MOVEDCB	MOVE DCB TO GOTTEN CORE	06302017
MVC DCBDDNAM, TIOFDDNM	PLACE DDNAME IN DCB	06308017
LR R1, R2	ADDRESS OF REMOTE LIST	A29871 06311019
MVI O(R1), X'80'	INDICATE END OF LIST	06314017
OPEN ((3), INPUT), MF=(E, (1))	OPEN THE DCB	06320017
TM DCBDFLGS, X'10'	OPEN SUCCESSFUL	06350017
RZ IEFSD033	BRANCH NO	06356017
ST R3, IEFDCBAD	DCB ADDRESS IN PARM LIST	06362017
R IEFSD035		A16531 06368018
IEFSD33A DS OH		A16531 06374018
TM SCTSTAT?, SCTSTPLR	STEPLIB PRESENT?	1270 06378019
RZ IEFSD033	NO, JOBLIB	17052 06383017
IEFSD33H DS OH		17052 06403017
CLC TIOFDDNM(8), STEPLIB	YES, DOES R10 POINT TO	17052 06423017
*	STEPLIB ENTRY	17052 06443017
RE IEFSD033	YES, GET OUT	06463017
IC R0, TIOFLNGH	NO, BUMP TO	17052 06523017
AR R10, R0	NEXT ENTRY	17052 06543017
R IEFSD33H	CONTINUE SEARCH	17052 06563017
A7Z EQU *		17052 06583017
BALR R15, 0		06587018
USING *, R15		A16531 06591018
		A16531 06595018

L	R12,STAFREG		A16531	06599018
DROP	R15		A16531	06603018
USING	IEFSD040:12,R12		A16531	06607018
CH	R0,TWELVE	WORKAREA	A16531	06611018
BE	NOGO	BRANCH NO	A16531	06615018
L	R9,0(R1)	GET STAFFARM ADDR	A16531	06619018
FREEMAIN	R,LV=104,A=(1)	FREE WKAREA	A16531	06623018
B	PROCESS	BRANCH TO CONTINUE	A16531	06627018
NOGO	LR	R9,R2	GET STAF PARM ADDR	A16531
PROCESS	LM	R2,R11,8(R9)	RESTORE REGISTER'S	A29871
IEFSD033	OI	LCTINTSW,LCTSD0XX	OVERRIDE USER PROGRAM NAME IN	I270
*		IN ATTACH PARM LIST.	I270	06671019
LA	R0,IEFDCBN-IEFDCHS	LENGTH OF DCH		06680000
LA	R4,TCORE	SUBPOOL NUMBER		06700000
SLL	R4,24	TO HIGH ORDER BYTE OF		06720000
OR	R0,R4	LENGTH REGISTER		06740000
LR	R1,R3	ADDRESS OF REMOTE DCH		06760000
FREEMAIN	R,LV=(0),A=(1)	FREE DCH AND EXIT LIST		06780000
L	R1,IEFUSADD	ADDRESS OF USER PARAMETERS		06800000
L	R4,4(R1)	LENGTH OF USER PARAMS	18428	06820017
LA	R0,8(R4)		18428	06840017
LA	R4,TCORE	SUBPOOL NUMBER		06860000
SLL	R4,24	TO HIGH ORDER BYTE OF		06880000
OR	R0,R4	LENGTH REGISTER		06900000
FREEMAIN	R,LV=(0),A=(1)	FREE USER PARAM LIST		06920000
LA	R4,8	GET LENGTH OF USER PARAMS		06940000
LA	R0,8(R4)	PLUS 8 FOR GETMAIN	18428	06960017
LA	R1,TCORE	SUBPOOL NUMBER	M2434	06980018
SLL	R1,24	TO HIGH ORDER BYTE OF	M2434	07000018
OR	R0,R1	LENGTH REGISTER	M2434	07020018
GETMAIN	R,LV=(0)	GETMAIN FOR USER'S PARAMS		07040000
ST	R1,IEFUSADD	SAVE NEW ADDRESS OF USER		07060000
ST	R4,4(0,R1)	LENGTH TO LIST	M2434	07080018
LA	R0,6(0,R1)	PLACE ADDRESS OF LENGTH FIELD	23021	07100018
ST	R0,0(0,R1)	IN THE GOTTEN LIST AND		07120000
OI	0(R1),128	SET HI-ORDER BIT TO 1		07140000
MVC	R(8,R1),TIDEDDNM	MOVE IN DDNAME	18428	07180017
IEFSD035	EQU	*		07220000
*	FREE THE	STAF PARMAREA GOTTEN FOR THE RETRY	A27679	07220019
LA	R0,STAELENTH	GET PARM LENGTH	A27679	07224019
LA	R14,TCORE	SUBPOOL NUMBER	A27679	07226019
SLL	R14,24	MOVE TO HI ORDER BYTE	A27679	07228019
OR	R0,R14	LENGTH REGISTER	A27679	07230019
FREEMAIN	R,LV=(0),A=(R9)	FREE THE AREA	A27679	07232019
BYPASS	EQU	*	A16531	07236019
L	R4,IEFTCRAD	GET ADDRESS OF INITIATOR'S TCB		07240000
*		IN R4		07260000
XC	LCTPARM1(X4),LCTPARM1	ZERO FIELD	I272	07262019
TM	SCTSTAT,X'04'	IS PROTECTION KEY TO BE CH	CR17	07264017
BZ	IEFSD040	NO.	CR17	07266017
MVC	LCTPARM1(1),TCBPKF(R4)	YES.SAVE OLD PROTECTION KE	CR17	07268017
TCHPKF	EQU	28	CR17	07270017
XC	TCHPKF(1,R4),TCHPKF(R4)	ZERO OUT PROTECTION KEY IN	CR17	07272017
IEFSD040	EQU	*	CR17	07274017
L	R3,IEFTIOTA	GET ADDRESS OF TIOT LIST IN R3		07280000
*				07300000
XC	0(4,R3),TIOTPOINT(R4)	THIS SERIES OF EXCLUSIVE OR'S		07320000

```

XC      TIOTPONT(4,R4),0(R3)      PUTS BACK THE JOB STEP AND      07340000
XC      0(4,R3),TIOTPONT(R4)      INITIATOR TIOT PIRS INTO THE  07360000
*                                     TIOT LIST AND THE INITIATOR'S  07380000
*                                     ICR RESPECTIVELY. (SEE ABOVE  07400000
*                                     WHERE THEY WERE SWAPPED TO  07420000
*                                     ALLOW OPEN TO REFERENCE THE JOB  07440000
*                                     STEP TIOT FROM INITIATOR'S TCB)  07460000
* RESTORE THE JSCH ADDRESSES WHICH WERE SWITCHED BEFORE ALLOCATION 1272 07540019
L        R0,TCRJSCH(R4)           JOB STEP JSCH
L        R3,LCTJSCH               INIT'S JSCH
ST       R0,LCTJSCH
ST       R3,TCRJSCH(R4)           JOB STEP JSCH IN LCT
LA       R1,IEFFCHLT              ADDRESS OF ECR LIST
ST       R1,IEFFCHLA              STORE IN PARAMETER LIST
LA       R3,IEFAFCH              ADDRESS OF ATTACH ECH
ST       R3,0(0,R1)              PLACE IN ECHLIST
LA       R4,CHCECH
ST       R4,4(0,R1)              PLACE IN FCHLIST
OI       4(R1),128              TURN ON HIGH ORDER BIT OF LIST
LR       R1,R2                   ADDRESS OF INPUT LIST
FREEMAIN R,LV=24,A=11,SP=TCORF   08000000
* THIS IS THE MACRO WHICH CANCELS THE STAE ENVIRONMENT
STAE     0                       08200000
LR       R1,R11                  CANCEL STAE
L        R15,IEF63VCN            POINT TO PARAMETER LIST
RR       R15                     ADDRESS OF ATTACH MODULE
* THIS IS THE STAE EXIT ROUTINE. HERE IS WHERE WE SPECIFY VIA 08240000
* A RETURN CODE IN R15 THAT A RETRY IS TO BE ATTEMPTED. WE 08260000
* RETURN TO THE STAE INTERFACE WITH THE RETRY ADDRS IN R0 08280000
AGAIN    EQU *
BALR     R15,0
USING    *,R15
L        R12,STAEREG
DRPP     R15                     RE-ESTABLISH RASEREG
USING    IEFSD062+2,R12
LA       R15,4
LA       R0,A7Z
RR       R14
TWELVE   DC H'12'
STAEREG  DC A(IEFSD062+2)
STAELENTH EQU 60
CFO      DC F'0'
MAINLGTH DC A(DATACELL-ICR0B)   WORD OF ZEROS FOR COMPARES
SUBLGTH  DC A(L'DATACELL)
IEFRDER  DC CL8'IEFRDER'
HOURS24  DC F'8640000'
STEPLIR  DC CL8'STEPLIR'
IEFOMGRV DC V(IEFOMRAW)
SYSCMD   ENQ (IEFSQCHS,IEF07DCH,E,2,SYSTEM),MF=L
IEF07DCH DC C'07'              MINOR FOR SYSTEM TASK
IEFSQCHS DC C'SYSIEFSO'        MAJOR
ACSTARTG DC C'STARTING'
BLANKFW  DC C' '
IEFMOVE1 MVC 8(0,R4),4(R5)      MOVE USER PARAMS FROM BUFFER
CLEARCOR XC XO(X0,R1),XO(R1)
SETRITS  OI JCTRSW2,X'00'      PUT C/R BITS IN JCT
TIOTPONT EQU 12

```

IEF63VCN DC	V(IEFSD063)		08440000
IEFSD65V DC	V(IEFSD065)		08460000
IEFSDST1 DC	V(IEFSD110)		08480000
SD514RTE DC	V(IEFSD514)	ADDR OF STACK INIT. PGM	
SD06250 DC	V(IEFSD112)	TIOT WRITE ROUTINE.	CR17 08485017
IEFSD161 DC	V(IEFSD061)	LINKAGE TO IEFSD161	11404 08490014
RESTART DC	CLR'IEFRSTRT'	STEP NAME FOR RESTARTING STEP	099 08490719
*			099 08491419
* * * * *			099 08492119
* ENQ/DEQ FOR DIRECT SYSOUT CONTROL BLOCKS (DSOCH)			099 08492819
DSOCHNO END (IEFSOCHS,DSOCHNOR,S,05,SYSTEM),MF=L			099 08493519
*			099 08494219
* * * ENQ/DEQ Q AND R NAMES			099 08494919
*			099 08495619
* USING MAJOR NAME IN ENQ 'SYSCMD'			099 08496319
DSOCHNOR DC C'DSOCH' R - MINOR NAME			099 08497019
*			099 08497719
MOVEDCH MVC	0(0,R3),IEFDCRS	MOVE DCR TO GUTTM CORE	099 08498419
TCORE EQU	253		08500000
N0 EQU	0		08520000
N1 EQU	1		099 08521019
N2 EQU	2		099 08522019
N3 EQU	3		099 08523019
N4 EQU	4		099 08524019
N7 EQU	7		099 08525019
N8 EQU	8		099 08526019
EIGHT EQU	8		099 08527019
NC EQU	12		099 08528019
N72 EQU	72		099 08529019
X104 EQU	104		099 08530019
N255 EQU	255		099 08531019
RSTRCK DC	CLR'IEFDSORP'		099 08532019
DSMSG DC	C'IEF387I DIRECT SYSOUT='		099 08533019
PERIOD EQU	C'.'		099 08534019
CONMA EQU	C'.'		099 08535019
JOBLLH EQU	X'80'		099 08536019
TCBJOBL EQU	40		08540000
X0 EQU	0		08560000
X1 EQU	1		1272 08560119
X3 EQU	3		1272 08560219
X4 EQU	4		1272 08560319
X7 EQU	7		1272 08560419
X8 EQU	8		1272 08560519
X24 EQU	24	DISPLACEMENT	1272 08560619
X96 EQU	96		1272 08560719
X184 EQU	184		1272 08560819
X255 EQU	255	USED IN MASK	1272 08560919
X256 EQU	256		1272 08561019
BLANK EQU	C' '		1272 08561119
IELRTNCD EQU	2		1272 08561219
NOSIOTS EQU	X'00'		1272 08561319
ONEREC EQU	X'01'		1272 08561419
READCODE EQU	X'04'		1272 08561519
MAINUCB EQU	X'FF'		1272 08561619
POOL246 EQU	246		1272 08561719
RTNCODE4 EQU	X'04'		1272 08561819
IOFLAG EQU	X'81'		1272 08561919
			1272 08562019

SPRGRAM EQU X'04'	VERB CODE FOR STARTED PROGRAM	1272	08562119
TGRTIO EQU 12	TIOI DISPLACEMENT IN TCB	1272	08562219
TGRTSCH EQU 180		1272	08562619
EJECT			08563017
IEFDCHS DS OF			08580000
DCH DSORG=PO,MACRF=E			08600000
EJECT			08610017
IEFDCHM EQU *			08640000
DCHM DSORG=(PO,XF)			08660000
EJECT			08670017
IEFDCSB DSECT			08680000
IEFCHAIN			08700000
EJECT			08710017
IEFONNGR			08720000
OMCAP EQU IOPARAMS+24	PTR IN OMPA TO TRK STACK INFO		08740000
EJECT			08750017
*****			08760000
*****			08780000
* THE FOLLOWING REPRESENTS A BLOCK OF MAIN STORAGE WHICH		*	08800000
* EXISTS WHILE THE JOB STEP IS RUNNING. IT IS GOTTEN		*	08820000
* BY THIS MODULE AND IS FREED BY IFFSD164		*	08840000
*		*	08860000
*		*	08880000
IEFPARAM DSECT			08900000
IEFUSADD DS F	ADDRESS OF USER PARAMETERS		08920000
IEFCHLA DS F	ADDRESS OF FCB LIST		08940000
IEFERMLS DS F	ADDRESS OF SUPERVISOR L LIST		08960000
IEFLCTAD DS F	ADDRESS OF LCT		08980000
IEFTCHAD DS F	ADDRESS OF INITIATOR'S TCB		09000000
IEFTIOTA DS F	ADDRESS OF TIOI		09020000
IEFDCHAD DS F	ADDRESS OF DCH	1270	09030019
IEFECHLT DS 2F	FCB LIST		09040000
IEFAFCB DS F	ATA		09060000
IEFPLEND EQU *		1270	09070019
IEFPSIZE EQU IEFPLEND-IEFPARAM		1270	09080019
EJECT			09090019
IEFZATTCH		1270	09100019
LISTSIZE EQU IEFPSIZE+ATLSTSI2		1270	09105019
EJECT			09110017
*****			09120000
*		*	09140000
* THE FOLLOWING REPRESENTS A BLOCK OF MAIN STORAGE WHICH IS		*	09160000
* ALLOCATED FOR THE LIFE OF		*	09180000
* OF THIS INITIATOR. IT CONTAINS:		*	09200000
*		*	09220000
* (1) THE LCT (LINKAGE CONTROL TABLE)		*	09240000
* (2) A 2 LEVEL REGISTER SAVE AREA FOLLOWED BY		*	09260000
* THE QUEUE MANAGER PARAMETER AREA		*	09280000
* (3) AN ALTERNATE QUEUE MANAGER PARAMETER AREA		*	09300000
*		*	09320000
*****			09340000
IEFLCT DSECT			09360000
IEFALLCT			09380000
EJECT			09390017
LCTBATMN DS F	*****TO BE ADDED TO LCT		09400000
LCTSOOMP DS F	*****TO BE ADDED TO LCT(VMS ONLY) POINTER TO		09420000
*	SYSOUT QUEUE MANAGER PARAMETER AREA		09440000

LCTRTN	DS	F	*****TO BE ADDED TO LCT(VMS ONLY) INITIATOR'S	09460000
*			RETURN POINT	09480000
LCTINTSW	DS	0C	INITIATOR INTERNAL SWITCHES	1272 09482019
LCTINHLR	EQU	128	RUN IN HIERARCHY ONE	1272 09484019
LCTSDOXX	EQU	32	ATTACH IEFSDOXX	1272 09486019
LCTMINRG	EQU	16	JOB FLUSH - USE MINPAR	1272 09488019
LCTSTART	EQU	8	TASKNAME NOT ON COMMAND	1272 09490019
LCTSTOP	EQU	4	INITIATOR INTERNAL STOP	1272 09492019
LCTCSCH	DS	F	*****TO BE ADDED TO LCT(VMS ONLY) INITIATOR CSCH	09500000
*			ADDRESS	09520000
LCTTMRK	DS	4F	*****TO BE ADDED TO LCT(VMS ONLY) TIMER WORK AREA	09540000
LCTJHLR	DS	F	*****TO BE ADDED TO LCT(VMS ONLY) JHLR DCR PTR	09560000
LCTATLST	DS	F	*****TO BE ADDED TO LCT(VMS ONLY) POINTER TO	09580000
*			ALLOCATE/TERMINATE PARAMETER LISTS	09600000
REGSAVE	DS	36F	REGISTER AVE AREA	09620000
QMGR1	DS	9F	QUEUE MANAGER PARAMETER AREA	09640000
QMGR2	DS	9F	ALTERNATE QUEUE MANAGER AREA	09660000
TRSTKINF	DS	2F	NEEDED FOR TRACK STACKING AND QUEUE	09680000
*			BREAK INFORMATION. 1ST BYTE TO CON-	09700000
*			TAIN NUMBER OF BUFFERS, NEXT 3 BYTES	09720000
*			FOR STACK ADDR, NEXT 4 BYTES FOR	09740000
*			QUEUE BREAK INFORMATION	09760000
ECBLIST	DS	1F	PTR	0C42 09764017
LCTIDENT	DS	2F	IDENTIFIER	0C42 09768017
LCTFORCE	DS	CL8	POSSIBLE FORCE VALUES	1241 09770018
LCTLIMIT	DS	C	LIMIT VALUE	1241 09772018
FRCPRTY	DS	C	FORCE PRTY HOLDER	1241 09774018
INITPRTY	DS	C	INITIATOR'S PRTY	1241 09776018
*			THESE FIELDS ARE NEEDED FOR L-SHAPE/INIT MERGE	1272 09776219
DS	OF			1272 09776419
LCTOPSW1	DS	0C	INITIATOR OPTIONS BYTE 1	1272 09776619
LCTPKEYF	EQU	128	DONT GET PROTECT KEY	1272 09776819
LCTDWF	EQU	64	DONT PROCESS DEDICATED WORKF	1272 09777019
LCTSTMODF	EQU	32	DONT PROCESS STOP/MODIFY	1272 09777219
LCTMINPF	EQU	16	GET REGION SIZE SPECIFIED	1272 09777419
LCTCANF	EQU	8	ALLOW CANCEL ONLY AT ALLOC	1272 09777619
LCTONEJF	EQU	4	PROCESS ONLY ONE JOB	1272 09777819
LCTICMODF	EQU	2	DONT PROCESS INITIATOR CMDS	1272 09778019
LCTEXIT	DS	F	ADDRESS INITIATOR EXIT LIST	1272 09778219
LCTOPSW2	DS	0C	INITIATOR OPTIONS BYTE 2	1272 09778419
LCTTIMEF	EQU	128	DONT TIME THIS JOB	1272 09778619
LCTCRF	EQU	64	DONT ALLOW CHECKPT/RESTART	1272 09778819
LCTDSOF	EQU	32	DONT PROCESS DSO	1272 09779019
LCTINTH0	EQU	16	INIT IN HIERARCHY ZERO	1272 09779219
LCTINTH1	EQU	8	INIT IN HIERARCHY ONE	1272 09779419
LCTCOM	DS	F	COMMUNICATIONS PARM AREA PTR	1272 09779619
LCTJSCB	DS	F	ADDRESS OF JSCB	1272 09779819
IEFEND	EQU	*	END OF LIFE-OF-TASK BLOCK	09780000
EJECT				09790017
IEFJCT	DSECT			09800000
IEFAJCTR				09820000
SMBHDR	EQU	SMRMSGLN		099 09820519
HEX00	EQU	X'00'	HEX EQUATE	09821018
DEC8	EQU	8	NUMERIC EQUATE	09822018
DEC4	EQU	4	NUMERIC EQUATE	09823018
DECO	EQU	0	NUMERIC EQUATE	09824018
HEX80	EQU	X'80'	HEX EQUATE	09825018

	EJECT		09830017
IEFSECT	DSECT		09840000
IEFASCTR			09860000
	EJECT		09870017
IEFTIOT	DSECT		09880000
IEFTIOT1			09900000
UCK	DSECT	1272	09903019
	IEFUCKOR	1272	09906019
	EJECT		09910017
SIOT	DSECT	1272	09915019
	IEFASIOI	1272	09920019
	EJECT		09925019
	IEZJSCB	1272	09930019
WTPCKPTR	FOU JSCKWTP-IEZJSCB		09935019
	EJECT		09941017
GETPTWT	DSECT		09942017
GPSIZEA	DS 1F	CONTAIN THE ADDRESS OF THE REGION SIZE LI	LS17 09943017
GPADOLT	DS 1F	CONTAIN THE ADDR OF ADDR LIST FOR THE HIF	LS17 09944017
GPCODE	DS X'0'	CONTAIN 80 TO INDICATE UNCONDITIONAL REGU	LS17 09945017
GPSUBP	DS X'0'	CONTAIN SUBPOOL 247 OR 246	LS17 09946017
	DS H'0'		LS17 09947017
GPFIARCO	DS OF	CONTAINS 00	LS17 09948017
GPADDOH	DS 1F	CONTAINS ADDRESS(SPECIFIC) ,ZERO(NON-SPEC	LS17 09949017
GPFIARCI	DS OF	CONTAINS 01	LS17 09950017
GPADOH1	DS 1F	CONTAINS ADDRESS(SPECIFIC) ,ZERO(NON-SPEC	LS17 09951017
GPSIZEHO	DS 1F	SIZE OF HO	LS17 09952017
GPSIZEH1	DS 1F	SIZE OF H1	LS17 09953017
GPSIZEPP	DS 1F	SIZE OF REGION FOR P/P	1272 09953319
GPSIZEFT	DS 1F	SIZE OF REGION FOR TERMINATE	1272 09953619
GPADDOH	DS 1F	ORIGINAL ADDRESS	CR17 09954017
GPMINPAR	DS 1F	247 + MINIMUM REGION SIZE	CR17 09955017
IEFDSOCH	DSECT		09956019
IEFDSOCH			09957019
	EJECT		09958019
	IEFHASEA		09959019
	IEFWTPCH	1272	09959519
	END		09960000

TITLE 'IEFSD263-MS/1 ATTACH MODULE'	00020000
IEFSD263 CSECT	00080014
* 039200-039800	M3767 00080119
	A24887 00080219
	A29871 00080519
* 029210-029220	M3441 00081019
	M0432 00082018
	M409 00085018
	M4160 00087019
	M2695 00088019
	M2696 00089019
*0054021400-021600	1272 00090014
*2286024200,024400	18428 00092017
*0642	MTS0 00095016
*019400-020200,020600,027200-027400,035200-035400,036200,036400	1272 00096019
*037600,040000	1272 00097019
	M4373 00098019
*027600-027800,034800-035000	M4005 00099019
*****	00100000
	* 00120000
INITIATOR ATTACH MODULE	* 00140000
	* 00160000
*****	00180000
	* 00200000
* STATUS - CHANGE LEVEL 000	* 00220000
	* 00240000
* FUNCTION - THE FUNCTION OF IEFSD263 IS	00260000
	00280000
(1) TO ATTACH THE JOB STEP. NOTE THAT THIS INVOLVES	00300000
SAVING THE CONTENTS OF THE FIELD IN THE INITIATOR'S	00320000
TCH WHICH POINTS TO THE INITIATOR'S TIOT, THEN	00340000
REPLACING THAT FIELD WITH THE POINTER TO THE JOB STEP	00360000
TIOT, THEN ISSUING THE ATTACH, THEN PUTTING THE	00380000
POINTER TO THE INITIATOR'S TIOT BACK IN ITS TCH.	00400000
	00420000
(2) TO DETERMINE WHETHER OR NOT THE JOB STEP IS TO BE	00440000
TIMED. IF SO TO ISSUE THE STIMER MACRO. IF NOT TO	00460000
BYPASS IT.	00480000
	00500000
(3) TO ISSUE A WAIT AGAINST THE POSTING OF A NORMAL STEP	00520000
TERMINATION ECH OR A STEP CANCEL ECH	00540000
	* 00560000
(4) TO INTERFACE WITH ABTERM IF THE JOB STEP IS	1272 00563019
CANCELED OR IF THE TIME LIMIT HAS EXPIRED	1272 00566019
	1272 00569019
(5) TO DETACH THE JOB STEP	1272 00572019
	1272 00575019
* ENTRY POINT: IEFSD263 FROM IEFSD103	* 00580000
	* 00600000
* INPUT - REGISTER ONE POINTING TO AN INPUT PARAMETER LIST CONTAINING -	* 00620000
(1) USER PARAMS ADDR	* 00640000
(2) ECH LIST POINTER	* 00660000
(3) ATTACH PARAMETER LIST ADDR	* 00680000
(4) LCT POINTER	* 00700000
(5) INITIATOR TCH POINTER	* 00720000
(6) TIOT LIST POINTER	* 00740000
(7) DCR ADDRESS	1270 00750019

```

*
* OUTPUT - REGISTER ONE POINTING TO THE INPUT PARM LIST CONTAINING - * 00760000
* (1) USER PARAMS ADDR * 00780000
* (2) PCB LIST POINTER * 00800000
* (3) ATTACH PARAMETER LIST ADDR * 00820000
* (4) LCT POINTER * 00840000
* (5) ADDRESS OF PARM AREA FOR SYSTEM TASKS. 1272 00860000
* ZERO FOR NON SYSTEM TASKS 1272 00880019
* (6) TINT LIST POINTER * 00900000
* (7) DCH ADDRESS 1270 00910019
* LCTPARM1-LCTPARM4 OR THE 16 BYTE PARAMETER AREA WHEN THE LCT IS *M4160 00911019
* WRITTEN TO THE JOB QUEUE HAS THE FOLLOWING FORMAT ON EXIT *M4160 00912019
* FROM IEFS0263 *M4160 00913019
* *M4160 00914019
* TCBFLG * TCT ADDRESS *M4160 00915019
* T T R OF LCT * LCTINTSW *M2696 00916019
* TCBDSP * CSCB ADDRESS *M4160 00917019
* TCBTCC (TASK COMPLETION CODE) *M4160 00918019
* *M4160 00919019
* * 00920000
* EXTERNAL REFERENCES- IEFSMFAT SMF1 00940018
* * 00960000
* EXIT IEFS0104 1272 00980019
* * 01000000
* TABLES/WORK AREAS: INPUT PARAMETER, LIFE-OF-TASK BLOCK * 01020000
* * 01040000
* ATTRIBUTES - REENTRANT * 01060000
* * 01080000
***** 01100000
R0 EQU 0 01120000
R1 EQU 1 01140000
R2 EQU 2 01160000
R3 EQU 3 01180000
R4 EQU 4 01200000
R5 EQU 5 01220000
R6 EQU 6 01240000
R7 EQU 7 01260000
R8 EQU 8 01280000
R9 EQU 9 01300000
R10 EQU 10 01320000
R11 EQU 11 01340000
R12 EQU 12 01360000
R13 EQU 13 01380000
R14 EQU 14 01400000
R15 EQU 15 01420000
Q0 EQU 0 ZERO DISPLACEMENT VALUE MP65 01422018
CVTPTR EQU 16 POINTER TO CVT MP65 01424018
CVTDCH EQU X'74' SYSTEM FLAGS BYTE IN CVT MP65 01426018
MP65FLAG EQU X'04' M65MP MULTIPROCESSOR SYSTEM MP65 01428018
STATUS EQU 79 STATUS SVC NUMBER MP65 01430018
STATFLAG EQU 5 TCB PARM TO STATUS SVC MP65 01432018
STMFLAG EQU X'02' STEP MUST COMPLETE BIT MP65 01434018
BALR R12,0 SET BASE REGISTER 01440000
USING *,R12 01460000
USING IEFPARAM,R8 01480000
USING IEFL0T,R4 01500000
USING GETPTWT,R11 1272 01506019

```

USING ATTCHLST,R3

01512019

```
*****
* THIS SECTION OF CODE IS FOR PROVIDING A METHOD OF IDENTIFYING
* THIS MODULE IN A MEMORY DUMP
*
B      FIRST
DC     X'CCCC'
DC     C'IEFSD263'
DC     X'02231970'
DC     X'CCCC'
FIRST  DS     OH
*****
```

```
*****
EJECT
LR     R8,R1          SAVE REGISTER ONE
L      R9,ALLCORE     PURGE MODULES FROM SUBPOOL 252
*                               WHICH HAVE A USER COUNT OF ZERO
GETMAIN EC,LV=(9),A=A
LM     R3,R4,IEFREMLS ADDRESSABILITY FOR ATTACH LIST
*                               AND THE LCT
L      R11,LCTPARM3   ADDRESS OF GWT
LA     R11,X0(R11)    ZERO HIGH ORDER BYTE
LTR    R11,R11        DOES GWT EXIST
RE     SD06305        BRANCH GWT DOES NOT EXIST
* IEFSD062 ISSUES OPENS FOR JOBLIB,STEPLIB OR FETCH AND
* SETS UP A STAE ENVIRONMENT TO PREVENT A 213 ABEND IN OPEN
* FROM BRINGING DOWN THE INITIATOR. OPEN LEAVES BEHIND SOME
* CORE IN SUBPOOL 252 THAT WOULD NOT MATTER IF A TASK ABENDS
* NORMALLY, BUT SINCE WE ARE INTERCEPTING THE 213 WE MUST
* FREE THIS CORE IF WE ARE GETTING A NEW PARTITION.
```

```
L      R14,LCTTCBAD   GET INITIATOR'S TCB.
L      R14,12(R14)    BUMP TO THE TIOT.
CL1    0(R14),C'F'    FIXED INITIATOR?
BE     SD06355        BRANCH TO KEEP CURRENT REGION.
```

```
LA     R0,252        SUBPOOL ID TO HI-ORD BYTE
SLL    R0,24         LENGTH AND ADDR BOTH ZERO
SR     R1,R1         INDICATES FREE ENTIRE SP
FREEMAIN R,LV=(0),A=(1) FREE SUBPOOL 252
LR     R1,R11        PARM LIST REPLACE PART
GETMAIN ,MF=(E,(1))  REPLACE PARTITION
L      R0,GPSIZET    REGION SIZE FOR TERMINATE
LTR    R0,R0         TEST FOR NEW REGION
BNE    SD06305       KEEP GWT FOR TERMINATE
SD06355 LR     R1,R11 ADDRESS OF GWT TO BE FREED.
DROP   R11
L      R0,LCTPARM4   LOAD SUBPOOL + LENGTH
FREEMAIN R,LV=(0),A=(1) FREE ITS CORE
XC     LCTPARM3+X1(X3),LCTPARM3+X1 ZERO PTR TO GWT
SD06305 EQU     *
L      R13,LCTCMCBA  PROVIDE SAVE AREA
LM     R0,R1,IEFTCBAD TCR ADDRESS IN REG 0, PTR TO
*                               TIOT PTR IN REG ONE
L      R15,SMFAT
BALR   R14,R15
LTR    R9,R15        GO TO SMF TO BUILD TCTIOT
*                               IF THERE WAS SMF PRESENT
*                               FOR THIS JOB, R9 NOW
```

```

*                               POINTS TO TCT                               SMF1 02324018
USING SMFCT,R9                                                         02332018
M/ SD63007 *                                                         02340018
ST R0,TCTWLMT ST JOB WAIT LIMIT IN R0 SMF1 02348018
MVC TCTSW(1),LCTTMWRK+R MOVE INDICATOR SET BY SMF1 02356018
* IEFSD162 OF WHETHER SMF1 02364018
* TIME LIMIT IS JOB OR SMF1 02372018
* STEP SMF1 02380018
SD63007 EQU * * 02388018
L R7,IEFUSADD MOVE USER PARAMETER AREA FROM 02400000
L R4,4(R7) SUBPOOL 253 TO SUBPOOL 0 18428 02420017
LA R4,8(R4) 18428 02440017
LR R0,R4 02460000
GETMAIN R,LV=(0) CORE FROM SUBPOOL ZERO 02400000
LR R0,R4 RESTORE LENGTH VALUE M409 02490018
HCTR R4,R0 SUBTRACT ONE FROM R4 02500000
EX R4,IEFUSEP MOVE USER PARM 02520000
LA R4,6(0,R1) SET UP USER PARM 02540017
ST R4,0(0,R1) AREA 02560000
DI 0(R1),128 SET HIGH ORDER BIT TO ONE 02580000
ST R1,IEFUSADD UPDATE USER PARM ADDR 02600000
LR R1,R7 LAAD ADDR OF USER PARM AREA 02620000
L R7,SP00L253 FREE SUBPOOL 253 M3441 02650019
OR R0,R7 02680000
FREEMAIN R,LV=(0),A=(1) FREE USER PARM IN 253 02700000
SR R9,R9 CLEAR REG 1241 02702018
LH R0,ATOPMOD LOAD DPMOD 1270 02704019
LTR R0,R0 IS DPMOD NEGATIVE 1241 02706018
BL SD06310 YES, BRANCH AND CONTINUE 1241 02708018
CHAP (0) NO, CHAP INIT UP 1241 02710018
LNR R9,R0 LOAD NEGATIVE VALUE IN REG TO 1241 02712018
SAVE ACROSS ATTACH 1241 02714018
SD06310 LM R1,R6,IEFPARAM PARMS INTO REGS 1272 02734019
L R7,TCHTINT(0,R5) SAVE THE POINTER TO THE 02800000
* INITIATOR'S TIT 02820000
MVC TCHTINT(4,R5),0(R6) STORE PTR TO THE JOB STEP TIT 02840000
* INTO INITIATOR'S TCB FOR ATTACH 02860000
* (R6 POINTS TO TIT LIST) 02880000
SR R11,R11 CLEAR REGISTER 02882016
TM 8(R3),X'80' NO, BRANCH 02884016
BC 8,AGG1 02886016
L R11,8(R3) SET R11 NEGATIVE 02888016
XI 8(R3),X'80' TURN OFF BIT IN ATTACH LIST (FCB) 02890016
AGG1 EQU * 02892016
LR R15,R3 ADDRESS OF SUPERVISOR L LIST 02900000
DROP R3 M4005 02900119
L R3,TCBJHL(R5) SAVE OUR JOBLIB PTR M4005 02900219
MVC TCBJHL(X4,R5),LCTJOBLB PUT NEW PTR FOR ATTACH M4005 02900319
***** 02900519
* 02901019
* SINCE THIS MODULE RESIDES IN THE LINK PACK AREA, THE ATTACH * 02901519
* PARAMETER LIST IS GOTTEN IN IEFSD162 AND INITIALIZED IN * 02902019
* IEFSD103, BOTH OF WHICH RUN IN THE INITIATOR REGION. THE * 02902519
* ADDRESS OF THE COMPLETED PARM LIST IS PART OF THE INPUT TO * 02903019
* THIS MODULE. THE FIELDS INITIALIZED AND OPTIONS SELECTED FOR * 02903519
* THE SUBTASK ARE OUTLINED HERE. * 02904019
* 02904519

```

*	KEYWORD SATISFIED	INFORMATION PROVIDED	*	02905019	
*			*	02905519	
*	EP	ENTRY POINT NAME	*	02906019	
*	ROLL	ROLLOUT CRITERIA	*	02906519	
*	DCH	JOBLIK DCH ADDRESS	*	02907019	
*	ECB	ECB WAITED ON BY INITIATOR	*	02907519	
*	LPMOD	LIMIT PRIORITY	*	02908019	
*	DPMOD	DISPATCHING PRIORITY	*	02908519	
*	HIARCHY	HIERARCHY INFORMATION	*	02909019	
*			*	02909519	
*	IN ADDITION THE PARAMETER LIST HAS BEEN INITIALIZED TO CAUSE			*	02910019
*	THE FOLLOWING PROCESSING BY THE ATTACH SERVICE ROUTINE.			*	02910519
*			*	02911019	
*	• TCB IS INITIALIZED AS A JOB STEP TCB.		*	02911519	
*			*	02912019	
*	• THE JOB PACK AREA QUEUE POINTER IS PROPAGATED FROM THE		*	02912519	
*	INITIATOR'S TCB		*	02913019	
*			*	02913519	
*	• A 72 BYTE REGISTER SAVE AREA IS PROVIDED FOR THE		*	02914019	
*	INITIAL ROUTINE OF THE JOB STEP		*	02914519	
*			*	02915019	
*	• THE PROTECT KEY IS PROPAGATED FROM THE INITIATOR'S TCB		*	02915519	
*			*	02916019	
*	• THE TASK IS SET TO RUN IN PROBLEM PROGRAM MODE.		*	02916519	
*			*	02917019	
*	• SUBPOOL ZERO IS SHARED WITH THE PROBLEM PROGRAM.		*	02917519	
*			*	02918019	
*	• THE JSCB ADDRESS IS PLACED IN THE SUBTASK'S TCB		*	02918519	
*			*	02919019	
*****				02919519	
	ATTACH MF=(E,(1)),SF=(F,(15))			02920000	
	ST R3,TCBJBL(R5)	RESTORE JOBLIK POINTER	M4005	02920719	
	L R3,LCTQDRTY	ADDRESS OF CSCR	1272	02921019	
	USING IFFCSCR,R3	CSCR ADDRESSABILITY	1272	02922019	
	ST R1,CHSPH	STORE TCB ADDRESS IN CSCR	1272	02922519	
	LTR R11,R11	IS TCB BIT TO BE SET FOR GRAPHICS		02924016	
	BZ AGG2	BRANCH IF NO		02924016	
	BI 20(R1),X'20'	SET TCBGRPH BIT IN TCB		02932017	
AGG2	EQU *			02936016	
	LR R11,R1	SAVE ATTACHED JOBSTEP TCB ADDR		02940000	
	ST R7,TCBTIOT(0,R5)	RESTORE THE POINTER TO THE		02960000	
		INITIATOR'S TIOT		02980000	
	L R1,CVTPTR	GET CVT ADDRESS	MP65	02982018	
	TM CVTDCB(R1),MP65FLAG	TEST IF MULTIPROCESSOR	MP65	02984018	
	BZ MPNDBYPS	NO JOBSTEP NOT RUNNING	MP65	02986018	
	L R0,MPNDFLGS	PARM FOR STATUS SVC	MP65	02988018	
	LA R1,Q0(R11)	PASS JOBSTEP TCB ADDRESS	MP65	02990018	
	SVC STATUS	SET TCB NONDISPATCHABLE	MP65	02992018	
MPNDBYPS	EQU *		MP65	02994018	
	LA R0,LCTTMWRK+4	ADDRESS OF TIME INTERVAL		03000000	
	LR R10,R0	PUT ADDRESS IN R10		03020000	
	CLC HOURS24(4),0(R10)	CHECK FOR INDICATION THAT STEP		03040000	
	BH SETTIME	IS NOT TO BE TIMED		03060000	
	LA R7,1	SET SWITCH TO INDICATE TIMER		03080000	
*		WAS NOT SET		03100000	
	MVC 4(4,R10),TUHRS24	SET TIME REMAINING FIELD OF		03120000	
*		TIMER WORK AREA TO 1440 MIN		03140000	

*		SO THAT IF THIS IS A CATALOGED	03160000
*		PROCEDURE THE REMAINING STEPS	03180000
*		WILL ALSO NOT BE TIMED	03200000
SETTIME	R	BYPASS	03220000
*	SR	R7,R7	03240000
		BRANCH AROUND STIMER MACRO	03260000
		SET SWITCH TO INDICATE TIMER	03280000
		WAS SET	03300000
		SET TIMER	03300000
BYPASS	STIMER TASK,EXIT,BINTVL=(0)		
	EQU	*	
	L	R6,LCTPARM1	1272 03301019
	LA	R6,X0(R6)	1272 03302019
	LTR	R6,R6	1272 03303019
	HE	NONSYSYK	1272 03304019
	MVC	X0(X16,R6),LCTPARM1	1272 03304519
	L	R1,SP00L253	M3441 03305519
	LA	R0,IEFEND-IEFLOT	1272 03307019
	OR	R0,R1	1272 03308019
	FREEMAIN R,LV=(0),A=(4)	FREE LCT	1272 03309019
	LR	R13,R6	1272 03310019
	B	GOWAIT	1272 03311019
NONSYSYK	LA	R13,LCTPARM1	1272 03312019
GOWAIT	EQU	*	1272 03313019
	LR	R1,R2	03320000
	L	R2,4(0,R1)	03340000
	NI	TCBPKF+5(R11),255-STMCLAG	MP65 03350018
	WAIT	ECBLIST=(1)	03360000
	LTR	R0,R9	1241 03360418
	BZ	SD06320	1241 03360818
	CHAP	(0)	1241 03361218
SD06320	MVC	TCBPQE(X4,R5),TCBPQE(R11)	1272 03362219
	CLI	X0(R13),PKEZERO	1272 03364619
	BZ	SD63020	1272 03367619
	MVC	TCBPKF(X1,R5),X0(R13)	1272 03370619
SD63020	EQU	*	CR17 03374017
	LTR	R7,R7	03380000
	HC	2,NOTEST	03400000
	TTIMER	CANCEL	03420000
	L	R9,TCBTCT(R5)	1272 03421019
	LA	R9,0(R9)	SMF 03423018
	LTR	R9,R9	SMF1 03424018
	BZ	NOSMF	03425018
	LM	R14,R15,TCTSTOF	SMF1 03426018
	XC	TCTSTOF(R),TCTSTOF	SMF1 03427018
	D	R14,CON385	SMF1 03428018
	A	R15,0(R10)	SMF1 03429018
	ST	R15,0(R10)	SMF1 03430018
NOSMF	EQU	*	03431018
	ST	R0,4(R10)	03440000
NOTEST	EQU	*	03460000
	TM	0(R2),POSTBIT	03580000
	BZ	SD63010	03600000
	CLI	CHSPC+X1,SYSCNCL	1272 03604019
	HE	CODE422	1272 03606019
	TM	X0(R2),X32	1272 03608019
	BZ	SD06710	1272 03610019
	MVC	CHSPC(X4),CODE1	1272 03612019
	R	SD06730	1272 03614019
SD06710	EQU	*	1272 03616019

	TM	X0(R2),X16	WAS JOB CANCELLED WITH DUMP	1272	03618019
	RZ	SD06720	NO	1272	03620019
	MVC	CHSPC(X4),CODE2	INDICATE DUMP	1272	03622019
	R	SD06730	CONTINUE	1272	03624019
SD06720	EQU	*		1272	03626019
	MVC	CHSFC(X4),CODE3	TIMER EXPIRED - DUMP	1272	03628019
	R	SD06730	CONTINUE	1272	03630019
CODE422	EQU	*		1272	03632019
	MVC	CHSPC(X4),CODE4	MOVE SYSTEM CANCEL CODE	1272	03634019
SD06730	EQU	*		1272	03636019
	OI	TCHFLGS+X1(R11),TCBSTI	INDIC NO STAF OPERATIONS	1272	03638019
	OI	CHSTS,CHABTERM	INDICATE ABTERM FOR SVC 34	1272	03640019
	LA	R1,X0(R3)	ADDRESS OF CSCB	1272	03642019
	MGR	(1),CHAIN	GO TO ABTERM VIA MASTER	1272	03644019
	LA	R1,IEFAECH	ADDRESS OF ATTACH ECB	1272	03646019
	WAIT	ECB=(1)	WAIT ON ATTACH ECB	1272	03648019
SD63010	SR	R0,R0	CLEAR REGISTERS TO INDICATE		03653019
	SR	R1,R1	FREE ALL OF SUBPOOL ZERO		03658019
	FREE	MAIN R,LV=(0),A=(1)	SUBPOOL ZERO		03663019
	L	R10,X12(R13)	SAVE SP AND LENGTH OF GWT	1272	03668019
	L	R7,X8(R13)	ADDRESS OF GWT	M4160	03673019
	LA	R7,X0(R7)	ZERO HIGH ORDER BYTE	M4160	03678019
	ST	R3,X8(R13)	STORE CSCB ADDRESS	M4160	03683019
	MVC	X8(X1,R13),TCBDSR(R11)	MOVE DISPATCHING PRIORITY	M4160	03688019
	MVC	CHPKE(X1),TCBPKF(R11)	SAVE PROTECT KEY IN CSCB	M3767	03700719
	MVC	X0(X1,R13),TCHFLGS(R11)	SAVE TCHFLGS	1272	03701019
	MVC	X1(X3,R13),TCBTCT+X1(R11)	SAVE TCT PTR	1272	03701519
	MVC	X12(X4,R13),TCBTCC(R11)	SAVE TASK COMPLETION CODE	1272	03702019
	XC	TCBFSA+X1(X3,R11),TCBFSA+X1(R11)		1272	03702519
			ZERO PGM SAVE AREA PTR	1272	03703019
	LA	R1,CHSPB	ADDRESS OF SUBTASK TCB	1272	03703519
	DETACH	(1)		1272	03704019
	USING	GETPTWT,R7		M4160	03704619
	LTR	R7,R7	ONES GWT STILL EXIST		03705219
	BE	SD06330	BRANCH NO GWT	1272	03706019
	L	R14,LCTTCBAD	GET INITIATOR'S TCB.		
	L	R14,12(R14)	BUMP TO THE TLOT.		
	CLI	O(R14),C'F'	FIXED INITIATOR?		
	BE	SD06330	BRANCH TO KEEP CURRENT REGION.		
	FREE	MAIN R,SP=247	FREE REGION	1272	03707519
	MVI	GPSUBP,POOL247	INDICATE GET REGION	1272	03708019
	XC	GPADDHO(X8),GPADDHO	ZERO SPECIFIC ADDRESS	1272	03708519
	TM	X7(R13),LCTIHIER	TEST HIERARCHY	M2696	03709019
	BO	SD06328	BRANCH HIERARCHY 1	1272	03709519
	MVC	GPSIZEHO(X4),GPSIZET	MOVE REGION SIZE	1272	03710019
	MVI	GPSIZEHO,ENDLIST	INDICATE END OF LIST	1272	03710519
	B	SD06329		1272	03711019
SD06328	MVC	GPSIZEH1+X1(X3),GPSIZET+X1		1272	03711519
SD06329	LR	R1,R7	ADDRESS OF GWT IN REG ONE	M2695	03712019
	GET	MAIN ,MF=(E,(1))	GET NEW REGION	1272	03712519
	LR	R1,R7	ADDRESS OF GWT TO BE FREED	M4160	03713019
	LR	R0,R10	SP AND LENGTH	1272	03713519
	FREE	MAIN R,LV=(0),A=(1)		1272	03714019
SD06330	EQU	*		1272	03714519
	ST	R6,X16(R8)	STORE ADDRESS OF PARM AREA FOR	1272	03715519
*			SYSTEM TASKS. R6 WILL BE ZERO	1272	03716019
*			FOR NON SYSTEM TASKS.	1272	03716519

	LR	R1,R8	PARAMETER LIST POINTER		03720000
	XCTL	EP=IEFSD100,ME=(F,(1))			03740000
EXIT	EQU	*	EXIT RTN USED WHEN TIMER EXP		03760019
	LR	R1,R2	CANCEL ECB		03780000
	LR	R7,R14	SAVE REG 14		03800000
	POST	(1)	POST CANCEL ECB		03820000
	LR	R14,R7	RESTORE REGISTER 14		03840000
	RETURN				03860000
HOURS24	DC	F'8640000'	1440 MINUTES (IN HUNDREDS OF SECONDS)		03880000
					03900000
TIMERS24	DC	X'C5'	1440 MINUTES IN TIMER UNITS	A24887	03920019
	DC	X'C1'	IF SPECIFIED ON EXECUTE CARD	A24887	03940019
	DC	X'00'	THE JOB IS NOT TO BE	A24887	03960019
	DC	X'00'	TIMED	A24887	03990019
A	DS	F			04020000
ALLCORE	DC	X'00FFFFFF'			04040000
SPDOL253	DC	X'FD000000'		M3441	04041019
MPNDFLGS	DC	AL2(STMCLAG),AL2(STATFLAG)	STATUS SVC PARMS	MP65	04043018
CONBR5	DC	F'385'	TO TO HUND SEC	SMF1	04046018
SMFAT	DC	V(IEFSMFAT)	SMF TCT BUILDER	SMF1	04052018
X0	EQU	0		1272	04052419
X1	EQU	1		1272	04052819
X3	EQU	3		1272	04053219
X4	EQU	4		1272	04053619
X8	EQU	8		1272	04054019
X16	EQU	16	DISPLACEMENT	1272	04054419
X32	EQU	32	DISPLACEMENT	1272	04054819
SYSCNCL	EQU	X'42'	SYSTEM CANCEL INDICATOR	1272	04055219
PDOL247	EQU	247		1272	04055619
ENDLIST	EQU	X'80'		1272	04056019
PPAREND	EQU	X'80'		1272	04056419
TCBFSA	EQU	112			04056819
TCBFLGS	EQU	29		1272	04057219
TCBPQE	EQU	X'98'		1272	04057419
TCBSTI	EQU	X'40'		1272	04057619
CODE1	DC	X'00222000'	JOB CANCELLED - NO DUMP	1272	04058019
CODE2	DC	X'80122000'	JOB CANCELLED - DUMP	1272	04058419
CODE3	DC	X'80322000'	TIMER EXPIRED - DUMP	1272	04058819
CODE4	DC	X'00422000'	EXCEEDED SPEC QUEUE SPACE	1272	04059219
POSTBIT	EQU	X'40'			04060000
PKFZERO	EQU	X'00'		1272	04062019
TC3TCC	EQU	X'10'		1272	04064019
TCBPKE	EQU	28		1272	04066019
TCBOSP	EQU	X'23'		M4160	04067019
X6	EQU	6		1272	04068019
X7	EQU	7		M2696	04069019
X12	EQU	12		1272	04070019
X24	EQU	24		1272	04072019
TCORE	EQU	253		1272	04074019
IEFUSEP	MVC	0(0,R1),0(R7)	MOVE USER PARM FROM 253 TO ZERO		04080000
TCBJBL	EQU	40	JOB LIB POINTER DISPLACEMENT IN		04100000
*			TCB		04120000
TCBTIOT	EQU	12	TIOT PTR DISPLACEMENT IN TCB		04140000
TCBTCT	EQU	164	*	SMF1	04150018
IEFPARAM	DSECT				04160000
IEFUSADD	DS	F	ADDRESS OF USER PARAMETERS		04180000
IEFECHLA	DS	F	ADDRESS OF ECB LIST		04200000

IEFRFMLS DS	F	ADDRESS OF SUPERVISOR L LIST	04220000
IEFLCTAD DS	F	ADDRESS OF LCT	04240000
IEFTCRAD DS	F	ADDRESS OF TCR	04260000
IEFTIOTA DS	F	ADDRESS OF TIOI	04280000
IEFDCBAD DS	F	ADDRESS OF DCR	1270 04240019
IEFECHLI DS	2F	FCR LIST	04300000
IEFAECH DS	F	ATTACH FCR	04320000
IEFPLEND EQU	*		04328019
IEFPSIZE EQU	IEFPLEND-IEFPARAM		04336019
IEFTCT			04344018
DEC246 EQU	246	NUMERIC EQUATE	04348018
DEC24 EQU	24	NUMERIC EQUATE	04352018
DECO EQU	0	NUMERIC EQUATE	04356018
*****			04380000
*			* 04400000
*	THE FOLLOWING REPRESENTS A BLOCK OF MAIN STORAGE WHICH IS		* 04420000
*	ALLOCATED FOR THE LIFE OF		* 04440000
*	OF THIS INITIATOR. IT CONTAINS:		* 04460000
*			* 04480000
*	(1) THE LCT (LINKAGE CONTROL TABLE)		* 04500000
*	(2) A 2 LEVEL REGISTER SAVE AREA FOLLOWED BY THE		* 04520000
*	THE QUEUE MANAGER PARAMETER AREA		* 04540000
*	(3) AN ALTERNATE QUEUE MANAGER PARAMETER AREA		* 04560000
*			* 04580000
*****			04600000
IEFLOT DSECT			04620000
IEFALLCT			04640000
LCTHATMN DS	F	*****TO BE ADDED TO LCT	04660000
LCTSDOMP DS	F	*****TO BE ADDED TO LCT(VMS ONLY) POINTER TO	04680000
*		SYSOUT QUEUE MANAGER PARAMETER AREA	04700000
LCTRTN DS	F	*****TO BE ADDED TO LCT(VMS ONLY) INITIATOR'S	04720000
*		RETURN POINT	04740000
LCTINTSW DS	0F	INITIATOR INTERNAL SWITCHES	1272 04742019
LCTIHIER EQU	128	RUN IN HIERARCHY ONE	1272 04744019
LCTSDOXX EQU	32	ATTACH IEFSDOXX	1272 04746019
LCTMINRG EQU	16	JOB FLUSH - USE MINPAR	1272 04748019
LCTSTART EQU	8	TASKNAME NOT ON COMMAND	1272 04750019
LCTSTOP EQU	4	INITIATOR INTERNAL STOP	1272 04752019
LCTCSCB DS	F	*****TO BE ADDED TO LCT(VMS ONLY) INITIATOR CSCB	04760000
*		ADDRESS	04780000
LCTTMWRK DS	4F	*****TO BE ADDED TO LCT(VMS ONLY) TIMER WORK AREA	04800000
LCTJOBRLB DS	F	*****TO BE ADDED TO LCT(VMS ONLY) JOBLIB DCR PTR	04820000
LCTATLST DS	F	*****TO BE ADDED TO LCT(VMS ONLY) POINTER TO	04840000
*		ALLOCATE/TERMINATE PARAMETER LISTS	04860000
REGSAVE DS	36F	REGISTER AVE AREA	04880000
QMGR1 DS	9F	QUEUE MANAGER PARAMETER AREA	04900000
QMGR2 DS	9F	ALTERNATE QUEUE MANAGER AREA	04920000
DS	8F		1272 04920919
* THESE FIELDS ARE NEEDED FOR L-SHAPE/INIT MERGE			* 1272 04921819
DS	0F		1272 04922719
LCTOPSW1 DS	0C	INITIATOR OPTIONS BYTE 1	1272 04923619
LCTPKYF EQU	128	DONT GET PROTECT KEY	1272 04924519
LCTDWFF EQU	64	DONT PROCESS DEDICATED WORKF	1272 04925419
LCTSTMDF EQU	32	DONT PROCESS STOP/MODIFY	1272 04926319
LCTMINPF EQU	16	GET REGION SIZE SPECIFIED	1272 04927219
LCTCANF EQU	8	ALLOW CANCEL ONLY AT ALLNO	1272 04928119
LCTONEJF EQU	4	PROCESS ONLY ONE JOB	1272 04929019

-104-

LCTICMDF	EQU	2	DONT PROCESS INITIATOR CMDS	1272	04929919
LCTEXIT	DS	F	ADDRESS INITIATOR EXIT LIST	1272	04930819
LCTOPSW2	DS	0C	INITIATOR OPTIONS BYTE 2	1272	04931719
LCTTIMEF	EQU	128	DONT TIME THIS JOB	1272	04932619
LCTCRF	EQU	64	DONT ALLOW CHECKPT/RESTART	1272	04933519
LCTDSOF	EQU	32	DONT PROCESS DSO	1272	04934419
LCTINTHO	EQU	16	EXECUTE INITIATOR IN HIER 0	1272	04935319
LCTINTH1	EQU	8	EXECUTE INITIATOR IN HIER 1	1272	04936219
LCTCOM	DS	F	COMMUNICATIONS PARM AREA PTR	1272	04937119
LCTJSCR	DS	F	ADDRESS OF JSCR		04938019
IEFEND	FQU	*	END OF LIFE-OF-TASK BLOCK		04940000
*			ORDER BYTE OF R15		04960000
IEFCSCB	DSECT			1272	08960019
IEECHAIN				1272	12960019
GETPTWT	DSECT				16960019
GPSIZEA	DS	1F	CONTAIN THE ADDRESS OF THE REGION SIZE LI	LS17	20960019
GPADLT	DS	1F	CONTAIN THE ADDR OF ADDR LIST FOR THE HIE	LS17	24960019
GPCODE	DS	X'0'	CONTAIN 80 TO INDICATE UNCONDITIONAL REGU	LS17	28960019
GPSUBP	DS	X'0'	CONTAIN SUBPOOL 247 OR 246	LS17	32960019
	DS	H'0'		LS17	36960019
GPFIARCO	DS	OF	CONTAINS 00	LS17	40960019
GPADDO	DS	1F	CONTAINS ADDRESS(SPECIFIC) ,ZERO(NON-SPEC	LS17	44960019
GPFIARCI	DS	OF	CONTAINS 01	LS17	48960019
GPADDOH1	DS	1F	CONTAINS ADDRESS(SPECIFIC) ,ZERO(NON-SPEC	LS17	52960019
GPSIZEHO	DS	1F	SIZE OF HO	LS17	56960019
GPSIZEH1	DS	1F	SIZE OF H1	LS17	60960019
GPSIZEPP	DS	1F	REGION SIZE FOR P/P	1272	64960019
GPSIZET	DS	1F	REGION SIZE FOR TERMINATE	1272	68960019
GPADDOH0	DS	1F	ORIGINAL ADDRESS	CR17	72960019
GPMINPAR	DS	1F	247 + MINIMUM REGION SIZE	CR17	76960019
IEZATTCH					80960019
END					84960019