SH20-0911-4

Program Product

Information Management System/360, Version 2 System Programming Reference Manual

Program Number 5734-XX6

Information Management System/360 is a processing program (control system) designed to facilitate the implementation of medium to large common data bases in a multiapplication environment. This enviroment is created to accommodate both online message processing and conventional batch processing, either separately or concurrently. The system permits the evolutionary expansion of data processing applications from a batch-only to a teleprocessing environment.

This manual provides system programming personnel with installation considerations and details for generation (definition) of an IMS/360 system for the user's data processing environment. The security maintenance program is included and message editing capabilities discussed. An IMS/360 sample problem is also provided.

Note: IMS/360 operates under several different operating systems (OS/MFT, OS/MVT, OS/VS1, and OS/VS2). For ease of reading, these are collectively referred to as "operating system" in this manual unless some function is peculiar to one of the operating systems.

IBM

Fifth Edition (September 1974)

This edition is a reprint of SH20-0911-3 incorporating changes released in Technical Newsletters SN20-2914 (dated December 4, 1972) and SH20-9004 (dated February 28, 1973).

This edition documents the addition of 3270 Display stations (local and remote), HIDAM/HDAM Distributed Free Space, a Test Driver Program (DFSDDLT0), and a Log Print and Formatting Dump Program to the capabilities of IMS/360 Version 2. Maintenance corrections have also been added to the manual.

This edition applies to Version 2, Modification Level 3, of the program product Information Management System/360 (5734-XX6) and to all subsequent versions and modifications until otherwise indicated in new editions or technical newsletters.

Changes are continually made to the information herein. Therefore, before using this publication, consult the latest IBM System/360 and System/370 Bibliography, GA22-6822, and the technical newsletters that amend that bibliography.

Copies of this and other IBM publications can be obtained through IBM branch offices.

A form has been provided at the back of this publication for readers' comments. If this form has been removed, address comments to: IBM Corporation, Department J04, 1501 California Avenue, Palo Alto, California 94304. Comments become the property of IBM.

CONTENTS

CHAPTER 1. INTRODUCTION

The System Programming Reference Manual is one of a set of manuals associated with the operation of the IBM Information Management System/360 (IMS/360). The accompanying illustration shows the logical relationships among the manuals in the set.

This manual provides system programming personnel with installation considerations and details for generation (definition) of an IMS/360 system for the user's data processing environment. The following functions are included:

- System Distribution, Handling, and Maintenance
- System Definition (IMS/360 System Generation)
- Security Maintenance
- User Modifications and Extensions to the Control Program
- IMS/360 Sample Problem

Other manuals in the set are:

IMS/360 GENERAL INFORMATION MANUAL (GH20-0765)

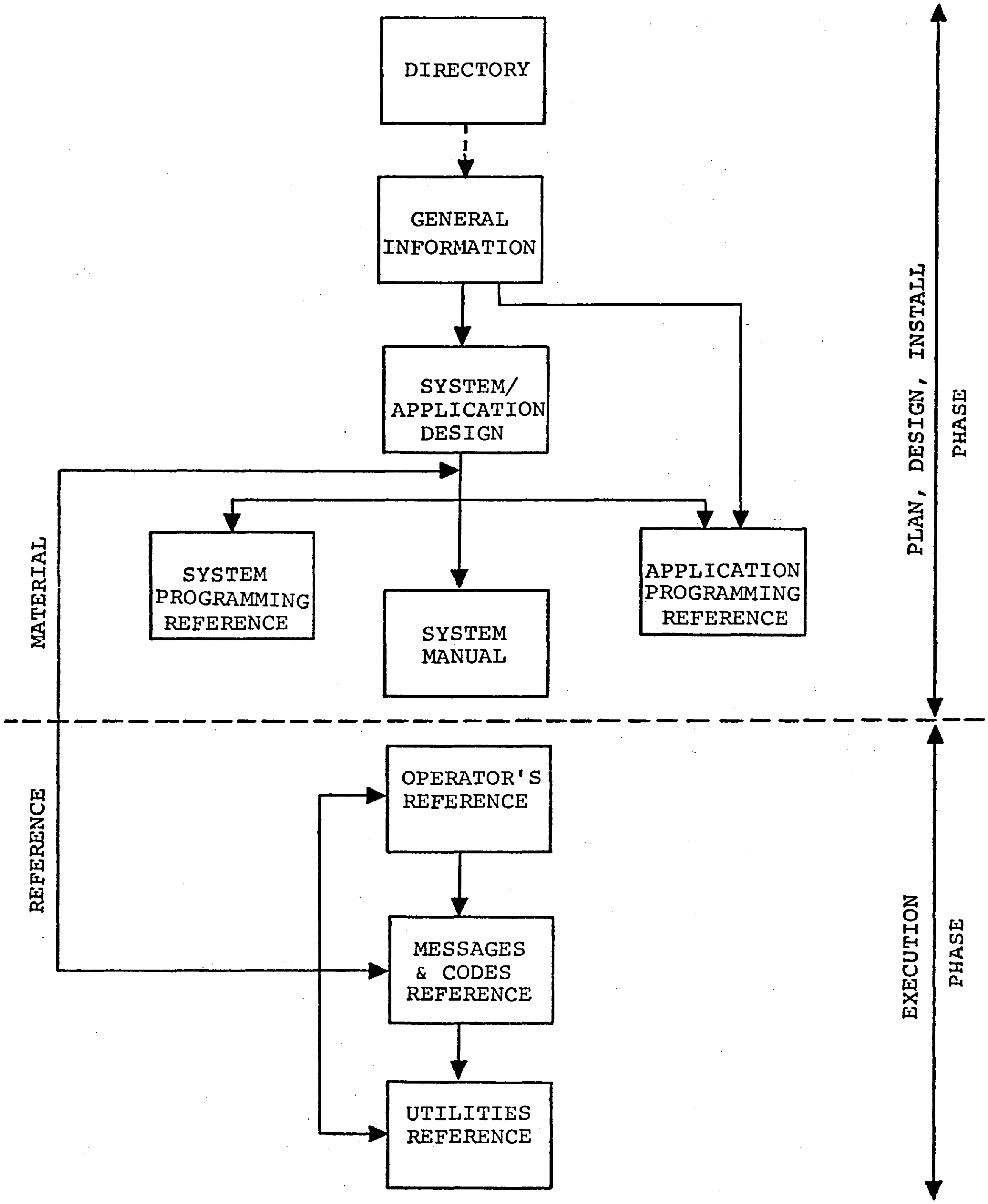
This manual describes the basic concepts and facilities of the IMS/360. This is the first or overview level of documentation in support of the Program Product.

SYSTEM/APPLICATION DESIGN GUIDE (SH20-0910)

This manual is a guide for the application analyst or system analyst. Its contents assist in the design of an IMS/360 system as well as the design of the application programs which operate under IMS/360. The System Programmer will want to be familiar with the information in this manual.

APPLICATION PROGRAMMING REFERENCE MANUAL (SH20-0912)

This manual is a reference manual for an application programmer. It provides information about the coding techniques necessary to implement a designed application under the IMS/360 control system. The System Programmer may want to be acquainted with the information in this manual.



IMS/360 relationship of manuals

UTILITIES REFERENCE MANUAL (SH20-0915)

IMS/360 system utility programs, their operation, and purpose are described in this manual. It tells how to execute these utilities under the operating system. The IMS/360 utilities described are:

- DBD Generation
- PSB Generation
- ACB Generation
- Data Base Reorganization and Recovery
- System Log Analysis (Statistics)
- Message Format/Language Utility

The System Programmer will need this manual as a reference volume.

OPERATOR'S REFERENCE MANUAL (SH20-0913)

The Operator's manual was prepared to provide the information associated with operating IMS/360 once the system has been established and running. The System Programmer may also want this manual for reference.

MESSAGES AND CODES REFERENCE MANUAL (SH20-0914)

This manual lists and explains appropriate responses to the completion codes, status codes, and messages produced by all the IMS-supplied components of the IMS/360 system.

SYSTEM MANUAL

The program logic of IMS/360 is explained in this manual. It consists of four volumes, as follows:

Volume I (LY20-0629) is the text and includes a description of each module comprising the IMS/360 program.

Flowcharts of the modules are contained in Volume II (LY20-0630).

Volume III (LYB0-0631) contains the listings of the IMS/360 Data Base System, furnished as microfiche.

Volume IV (LYB0-0632) contains the listings of the IMS/360 Data Communications System, furnished as microfiche.

The System Programmer will want these volumes for reference.

CHAPTER 2. SYSTEM DISTRIBUTION, HANDLING, AND MAINTENANCE

SYSTEM DISTRIBUTION

IMS/360 Version 2 is available in two configurations:

1. Data Base System

2. Data Base - Data Communication System

The Data Base System provides batch processing only. The Data Base - Data Communication System provides message processing, batch-message processing, and batch processing. If batch-only processing is desired, the user may order the Data Base System alone. If message, batch-message, and/or batch processing is desired, the user may order the Data Base System and the Data Communication Feature. The Data Base - Data Communication System is created by a marriage of the Data Base System modules and the Data Communication Feature modules.

The distribution of IMS/360 is made on unlabeled, nine-track, 800-bpi or 1600-bpi magnetic tape, or unlabeled, seven track, 800-bpi magnetic tape. The seven-track tape distribution requires the Data Conversion Feature. The nine-track tape distribution is recommended, because two nine-track tapes are required for the IMS/360 Data Base - Data Communication System. All IMS/360 distribution data sets are unloaded copies of direct access partitioned data sets. They have been moved to tape using the IBM operating system IEHMOVE Utility program.

DATA BASE SYSTEM

The Data Base System distribution is composed of one tape. This basic distribution tape includes three data sets. These are:

- IMS/360 Data Base Macro-Definition Library (IMS.DBGENLIB)

- IMS/360 Data Base Load Module Library (IMS.DBLOAD)

- IMS/360 Data Base Source Module Library (IMS.DBSOURCE)

When the IMS/360 Data Base System user receives the IMS/360 Data Base System distribution tape, the IEHMOVE program should be employed to copy and rename these data sets to direct access storage (Figure 1). The following job control language statements and utility control cards should assist in the copy execution. While preallocation of data sets is not recommended, the DCB attributes of the IMS2.GENLIB and IMS2.LOAD data sets must correspond to those of SYS1.MACLIB and SYS1.LINKLIB respectively. This may require an additional move/copy (disk to disk) after copying from tape to disk.

Those parameters which are underlined are user-specifiable (for example, 2314 rather than 2311). The REGION parameter is required only for Operating System/360 MVT execution. Generic name 2400-4 is nine track at 800 bpi with DCB=(DEN=2); generic name 2400-2 is seven track with data conversion at 800 bpi with DCB=(DEN=2).

BASIC

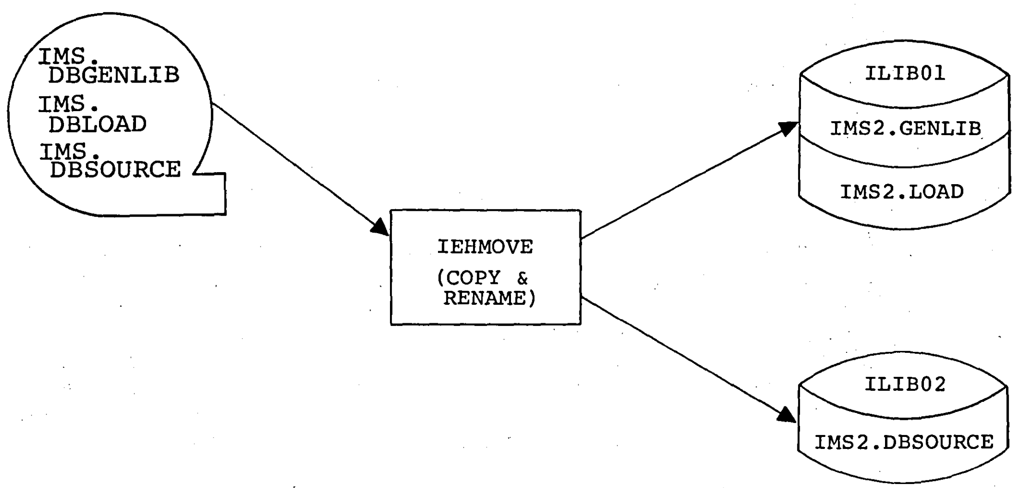


Figure 1. IEHMOVE to direct access storage (Data Base System)

Nine-Track Tape

```
//COPY         JOB      848,NAME,MSGLEVEL=1
//             EXEC     PGM=IEHMOVE,REGION=100K
//SYSPRINT     DD       SYSOUT=A
//SYSUT1       DD       UNIT=2311,DISP=OLD,VOLUME=SER=111111
//TAPE1        DD       UNIT=(2400-4,,DEFER),DISP=OLD,                X
//                      VOLUME=SER=SCRTCH,DCB=(LRECL=80,              X
//                      RECFM=FB,BLKSIZE=800,DEN=2),LABEL=(,NL)
//DISK1        DD       UNIT=2311,DISP=OLD,VOLUME=SER=ILIB01
//DISK2        DD       UNIT=2311,DISP=OLD,VOLUME=SER=ILIB02
//SYSIN        DD       *
               COPY     PDS=IMS.DBGENLIB,                             X
                        FROM=2400-4=(SCRTCH,1),                       X
                        TO=2311=ILIB01,FROMDD=TAPE1,                  X
                        RENAME=IMS2.GENLIB
               COPY     PDS=IMS.DBLOAD,FROM=2400-4=(SCRTCH,2),        X
                        TO=2311=ILIB01,FROMDD=TAPE1,                  X
                        RENAME=IMS2.LOAD
               COPY     PDS=IMS.DBSOURCE,FROM=2400-4=(SCRTCH,3),X
                        TO=2311=ILIB02,FROMDD=TAPE1,                  X
                        RENAME=IMS2.DBSOURCE
/*
```

Seven-Track Tape

```
//COPY2       JOB     848,NAME,MSGLEVEL=1
//            EXEC    PGM=IEHMOVE,REGION=100K
//SYSPRINT    DD      SYSOUT=A
//SYSUT1      DD      UNIT=2311,DISP=OLD,VOLUME=SER=111111
//TAPE1       DD      UNIT=(2400-2,,DEFER),DISP=OLD,              X
//                    LABEL=(,NL),                                X
//                    VOLUME=SER=SCRTCH,DCB=(LRECL=80,            X
//                    RECFM=FB,BLKSIZE=800,DEN=2,TRTCH=C)
//DISK1       DD      UNIT=2311,DISP=OLD,VOLUME=SER=ILIB01
//DISK2       DD      UNIT=2311,DISP=OLD,VOLUME=SER=ILIB02
//SYSIN       DD      *
              COPY    PDS=IMS.DBGENLIB,TO=2311=IBIL01,            X
                      FROM=2400-2=(SCRTCH,1),FROMDD=TAPE1,        X
                      RENAME=IMS2.GENLIB
              COPY    PDS=IMS.DBLOAD,TO=2311=ILIB01,              X
                      FROM=2400-2=(SCRTCH,2),FROMDD=TAPE1,        X
                      RENAME=IMS2.LOAD
              COPY    PDS=IMS.DBSOURCE,TO=2311=ILIB02,            X
                      FROM=2400-2=(SCRTCH,3),FROMDD=TAPE1,        X
                      RENAME=IMS2.DBSOURCE
/*
```

DATA BASE - DATA COMMUNICATION SYSTEM

The Data Base - Data Communication System is created by merging the data sets in the Data Base System distribution with the data sets in the Data Communication Feature. This is accomplished through the IEHMOVE program. The basic Data Communication Feature distribution tape includes three data sets. These are:

- IMS/360 Data Communication Feature Macro-Definition Library (IMS.DCGENLIB)
- IMS/360 Data Communication Feature Load Module Library (IMS.DCLOAD)
- IMS/360 Data Communication Feature Source Module Library (IMS.DCSOURCE)

When the IMS/360 Data Base - Data Communication System user receives the IMS/360 Data Base System and Data Communication Feature distribution tape, the IEHMOVE program should be used to copy and rename these data sets to direct access storage (Figures 1 and 2). The job control language statements and utility control cards (in Figures 1 and 2) should assist in the copy execution. While preallocation of data sets is not recommended, the DCB attributes of the IMS2.GENLIB and IMS2.LOAD data sets must correspond to those of SYS1.MACLIB and SYS1.LINKLIB respectively. This may require an additional move/copy (disk to disk) after copying from tape to disk.

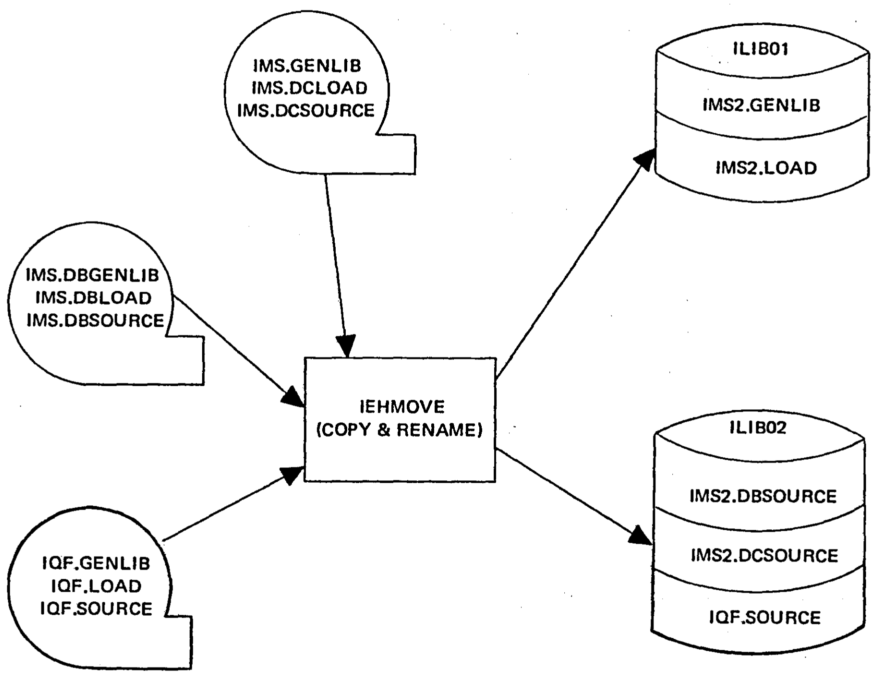


Figure 2. IEHMOVE to direct access storage (DB/DC System)

Nine-Track Tape

```
//COPY          JOB      848,NAME,MSGLEVEL=1
//              EXEC     PGM=IEHMOVE,REGION=100K
//SYSPRINT      DD       SYSOUT=A
//SYSUT1        DD       UNIT=2311,DISP=OLD,VOLUME=SER=111111
//TAPE1         DD       UNIT=(2400-4,,DEFER),DISP=OLD,                X
//                       VOLUME=SER=SCRTCH,DCB=(LRECL=80,              X
//                       RECFM=FB,BLKSIZE=800,DEN=2),LABEL=(,NL)
//DISK1         DD       UNIT=2311,DISP=OLD,VOLUME=SER=ILIB01
//DISK2         DD       UNIT=2311,DISP=OLD,VOLUME=SER=ILIB02
//SYSIN         DD       *
                COPY     PDS=IMS.DCGENLIB,                             X
                         FROM=2400-4=(SCRTCH,1),                       X
                         TO=2311=ILIB01,FROMDD=TAPE1,                  X
                         RENAME=IMS2.GENLIB
                COPY     PDS=IMS.DCLOAD,FROM=2400-4=(SCRTCH,2),        X
                         TO=2311=ILIB01,FROMDD=TAPE1,                  X
                         RENAME=IMS2.LOAD
                COPY     PDS=IMS.DCSOURCE,FROM=2400-4=(SCRTCH,3),X
                         TO=2311=ILIB02,FROMDD=TAPE1,                  X
                         RENAME=IMS2.DCSOURCE
/*
```

Seven-Track Tape

```
//COPY2        JOB     848,NAME,MSGLEVEL=1
//             EXEC    PGM=IEHMOVE,REGION=100K
//SYSPRINT     DD      SYSOUT=A
//SYSUT1       DD      UNIT=2311,DISP=OLD,VOLUME=SER=111111
//TAPE1        DD      UNIT=(2400-2,,DEFER),DISP=OLD,               X
//                     LABEL=(,NL),                                 X
//                     VOLUME=SER=SCRTCH,DCB=(LRECL=80,             X
//                     RECFM=FB,BLKSIZE=800,DEN=2,TRTCH=C)
//DISK1        DD      UNIT=2311,DISP=OLD,VOLUME=SER=ILIB01
//DISK2        DD      UNIT=2311,DISP=OLD,VOLUME=SER=ILIB02
//SYSIN        DD      *
               COPY    PDS=IMS.DCGENLIB,TO=2311=IBIL01,             X
                       FROM=2400-2=(SCRTCH,1),FROMDD=TAPE1,         X
                       RENAME=IMS2.GENLIB
               COPY    PDS=IMS.DCLOAD,TO=2311=ILIB01,               X
                       FROM=2400-2=(SCRTCH,2),FROMDD=TAPE1,         X
                       RENAME=IMS2.LOAD
               COPY    PDS=IMS.DCSOURCE,TO=2311=ILIB02,             X
                       FROM=2400-2=(SCRTCH,3),FROMDD=TAPE1          X
                       RENAME=IMS2.DCSOURCE
/*
```

Those parameters which are underlined are user-specifiable (for example, 2314 rather than 2311). The REGION parameter is required only for OS/MVT or /VS2 execution. Generic name 2400-4 is nine track at 800 bpi with DCB=(DEN=2); 2400-2 is seven track with data conversion at 800 bpi with DCB=(DEN=2).

INTERACTIVE QUERY FACILITY (IQF)

The IQF/IMS system is created by merging the data sets in the Data Base/Data Communication system distribution with the data sets in the IQF feature. This is accomplished through the IEHMOVE program.

The IQF/IMS System distribution tape includes the following three data sets:

- IQF/IMS Macro-Definition Library (IQF.GENLIB)
- IQF/IMS Load Module Library (IQF.LOAD)
- IQF/IMS Source Module Library (IQF.SOURCE)

When the IMS/360 user receives his IQF feature distribution tape, the operating system utility program IEHMOVE should be employed to copy and rename these data sets to direct access storage (Figure 2). The following job control language statements and utility control cards should assist in the copy execution. While preallocation of data sets is not recommended, the DCB attributes of the IQF.GENLIB and IQF.LOAD data sets must correspond to those of SYS1.MACLIB and SYS1.LINKLIB respectively. This may require an additional move/copy (disk to disk) after copying from tape to disk.

Nine-Track Tape for IQF

```
//COPY      JOB     848,NAME,MSGLEVEL=1
//          EXEC    PGM=IEHMOVE,REGION=100K
//SYSPRINT DD       SYSOUT=A
//SYSUT1    DD      UNIT=2311,DISP=OLD,VOLUME=SER=111111
//TAPE1     DD      UNIT=(2400-4,,DEFER),DISP=OLD,                      X
//                  VOLUME=SER=SCRTCH,DCB=(LRECL=80,                    X
//                  RECFM=FB,BLKSIZE=800,DEN=2),LABEL=(,NL)
//DISK1     DD      UNIT=2311,DISP=OLD,VOLUME=SER=ILIB01
//DISK2     DD      UNIT=2311,DISP=OLD,VOLUME=SER=ILIB02
//SYSIN     DD      *
            COPY    PDS=IQF.GENLIB,
                    FROM=2400-4=(SCRTCH,1),                             X
                    TO=2311=ILIB01,FROMDD=TAPE1,RENAME=IMS2.GENLIB
            COPY    PDS=IQF.LOAD,FROM=2400-4=(SCRTCH,2),                X
                    TO=2311=ILIB01,FROMDD=TAPE1,RENAME=IMS2.LOAD
            COPY    PDS=IQF.SOURCE,FROM=2400-4=(SCRTCH,3),              X
                    TO=2311=ILIB02,FROMDD=TAPE1,RENAME=IQF.SOURCE
/*
```

Seven-Track Tape for IQF

```
//COPY      JOB     848,NAME,MSGLEVEL=1
//          EXEC    PGM=IEHMOVE,REGION-100K
//SYSPRINT DD       SYSOUT=A
//SYSUT1    DD      UNIT=2311,DISP=OLD,VOLUME=SER=111111
//TAPE1     DD      UNIT=(2400-2,,DEFER),DISP=OLD,          X
//                  VOLUME=SER=SCRTCH,DCB=(LRECL=80,        X
//                  RECFM=FB,BLKSIZE=800,DEN=2),LABEL-(,NL)
//DISK1     DD      UNIT=2311,DISP=OLD,VOLUME=SER=ILIB01
//DISK2     DD      UNIT-2311,DISP=OLD,VOLUME=SER=ILIB02
//SYSIN     DD      *
            COPY    PDS=IQF.GENLIB,
                    FROM=2400-2=(SCRTCH,1),                 X
                    TO 2311=ILIB01,FROMDD=TAPE1,RENAME=IMS2.GENLIB
            COPY    PDS-IQF.LOAD,FROM=2400-2=(SCRTCH,2),    X
                    TO=2311=ILIB01,FROMDD=TAPE1,RENAME=IMS2.LOAD
            COPY    PDS=IQF.SOURCE,FROM=2400-2=(SCRTCH,3),  X
                    TO=2311=ILIB02,FROMDD=TAPE1,RENAME=IQF.SOURCE
/*
```

Those parameters whice are underlined are user-specifiable (for example, 2314 rather than 2311). The REGION parameter is required only for MVT execution. Generic Name 2400-4 is nine track at 800 bpi with DCB=(DEN=2); 2400-2 is seven track with data conversion at 800 bpi with DCB=(DEN=2).

SYSTEM HANDLING

Once the IMS/360 libraries have been copied from the distribution tape to direct access storage, the user is able to begin to tailor IMS/360 to his data processing environment. The tailoring of IMS/360 to a particular user's data processing environment is accomplished with the IMS/360 system definition macro-instructions which are contained within IMS2.GENLIB. Because the IMS/360 system executes with a collection of control blocks, which describe the user's data processing environment, system definition is required. These control blocks, which describe application programs, data bases, communication lines and terminals, and other IMS/360 resources, are constructed by the system definition process.

The IMS/360 user must prepare a control card input deck for IMS/360 system definition. The control card types and formats are described later in this manual. Once the control card deck has been prepared, it is appended to a package of job control language for the macro-instruction assembly of system definition. System definition is required if either an online message processing and batch processing (combined Data Communication - Data Base) or a batch-only processing (Data Base System) system is desired (Figure 3).

If the user has an existing IMS/360 system which is not to be altered except for the inclusion of the Interactive Query Facility feature (IQF), an online system definition is required as shown in Figure 3 and described in the discussion of the IMSCTRL macro-instruction in Chapter 3 of this manual. The IQF user must prepare a control card input deck (macro-instructions) to specify the data bases accessible to IQF as well as the transaction codes to invoke a query. These statements must be added to the decks previously used to perform the system definition. See Chapter 3 of this manual.

If the user is installing IMS/360 for the first time, the macro-instructions to define data bases and transaction codes for IQF can be incorporated in the initial system definition.

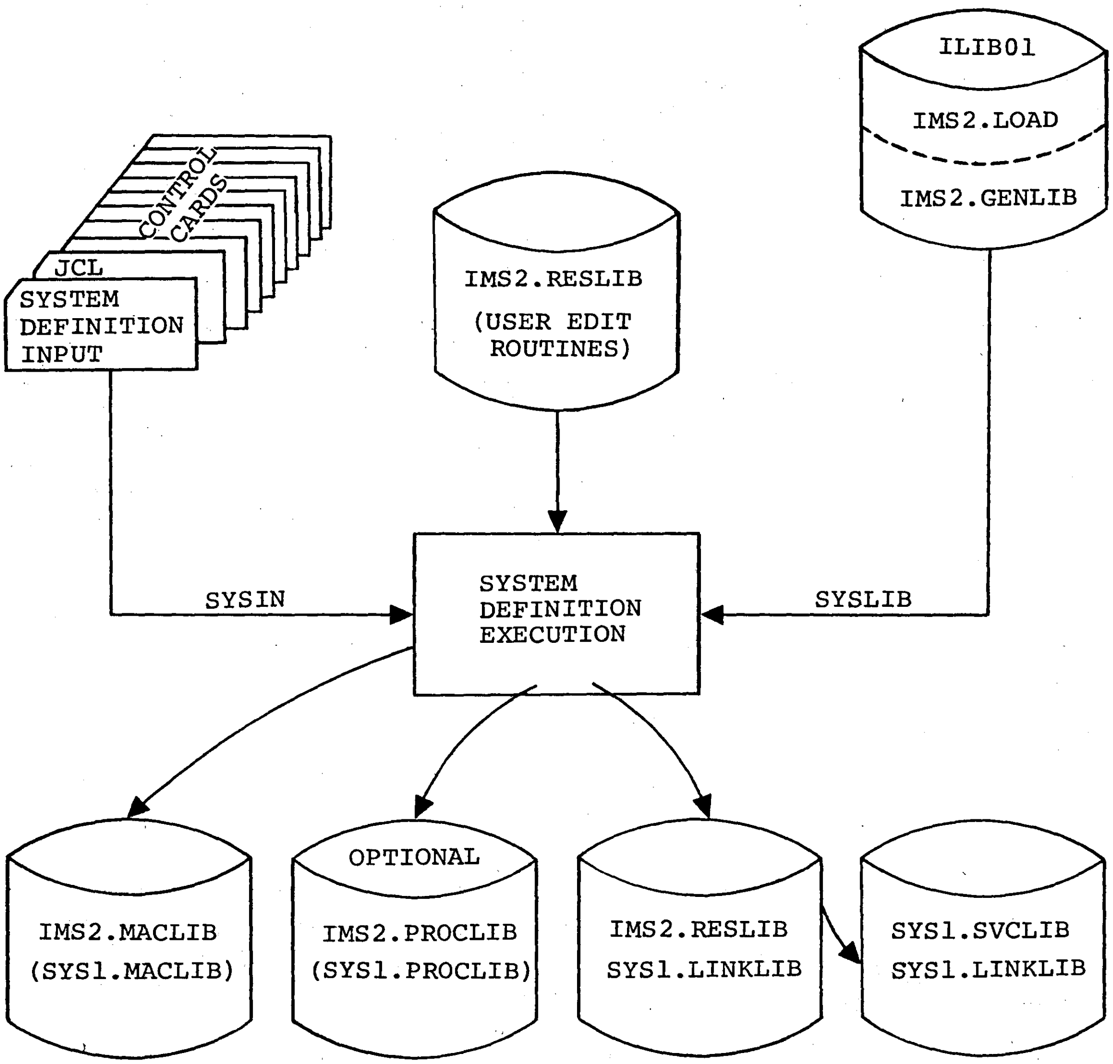


Figure 3. System definition handling

The output from IMS/360 system definition may include:

- Generation and placement of IMS/360 control program control blocks into the IMS2.RESLIB library
- Generation of the IMS/360 control program nucleus into the IMS2.RESLIB library, with the optional inclusion of user message edit routines
- Generation and placement into the IMS2.RESLIB library of the Data Language/I batch processing nucleus
- The linkage edit of three user supervisor calls (SVCs), one of which is used for interregion communication, one for online system operation, and one for OSAM multivolume execution. These are placed in the IMS2.RESLIB library. The user can specify the desired SVC numbers. The OSAM SVC modules must be link edited into the SYS1.SVCLIB data set. The other two SVC modules must be link edited with the OS nucleus.
- The naming and creation of the OSAM channel-end appendage module in the IMS2.RESLIB library. The user can specify the module name and must link edit the module to SYS1.SVCLIB.

- The moving of procedures to a user procedure library or SYS1.PROCLIB. These procedures are used for data base description (DBD) and program specification block (PSB) generation, ACB generation and maintenance, IMS/360 execution, message region execution, batch region execution, etc.

- The naming and creation of the 7770-3 channel/abnormal end appendage module in the IMS2.RESLIB library if the system contains 7770-3 lines. The user can specify the module name and must link edit the module to SYS1.SVCLIB.

- The link edit of an additional load of the Type 4 SVC to the IMS2.RESLIB library if the system contains 7770-3 lines. The module must be link edited into SYS1.SVCLIB with the OSAM SVC modules.

Once IMS/360 system definition has been performed, a Type 1 SVC for interregion communication and one Type 2 SVC routine must be link edited with the operating system nucleus. The OSAM Type 4 SVC routines must be link edited into SYS1.SVCLIB.

The IMS/360 user must have provided space for one Type 1 user SVC routine, one Type 2 user SVC routine, and one Type 4 user SVC routine in his operating system system generation. If the system definition is for batch Data Base System execution only, the Type 4 user SVC is all that is required. This SVC is used by OSAM. The procedure for relink-editing the operating system nucleus with the user SVC routines is specified in Chapter 3 of this manual.

Once system definition, the SVC-operating system nucleus link-edit, and the OSAM SVC link-edit to SYS1.SVCLIB are performed, the user must allocate direct access space for the DBD, PSB, ACB, and application program libraries. In addition, if online processing is desired, space should be allocated for message queue, conversation scratchpad, and log restart data sets. All these data sets must be cataloged. In addition, the operating system privileged name table module in SYS1.LINKLIB must be modified to incluude the name DFSRRC00 and the procedures IMS and IMSRDR must be moved to SYS1.PROCLIB. This allows for START and STOP IMS commands which initiate and terminate the IMS/360 online control region to be executed. Chapter 3 of this manual describes the execution of these functions.

Finally, the user is ready to create DBDs, PSBs, ACBs, and application programs. Before any message processing may be performed, the required data bases must be created in the batch region environment.

SYSTEM MAINTENANCE

Permanent modifications and corrections to problems encountered with the IMS/360 system are provided with complete library replacements through distribution of new modification levels of all IMS/360 libraries. These "mod-level" distributions are made on a periodic basis.

Between releases of modification levels of IMS/360, users receive corrections to the system on PTF (Program Temporary Fix) tapes. These PTF tapes are automatically distributed to licensed users. They are distributed based on the number of problems discovered and fixed since either the most recent modification of the program product or of the PTF tape. The PTF tape contains symbolic updates for the IMS2.DBSOURCE, IMS2.DCSOURCE, and IMS2.GENLIB libraries. It also contains replacement load modules for the IMS2.LOAD library. These load modules correspond to the SOURCE library updates.

Between the releases of PTF tapes, fixes for problems reported by users are distributed in two ways: (1) The user who reports the problem receives a response letter describing the source symbolic updates necessary to correct the problem. (2) A summary of the problem and its solution are entered into an IBM internal information system through which Field Engineering representatives can acquire problem summaries and solutions for all users.

The solutions to individual problems provided between PTF tapes and modification levels are source symbolic updates. If the updates are for the IMS2.DBSOURCE or the IMS2.DCSOURCE library, the user is instructed to perform a "temporary" update, assemble, and link edit the object module into either IMS2.LOAD or IMS2.RESLIB. If the symbolic change affects IMS2.GENLIB, it may be necessary to perform a new IMS/360 system definition. If this is so, the instructions accompanying the change will specify that a new system definition is necessary. Typically, individual fixes only require assembly of updated source members and reprocessing of the affected IMS/360 load modules to incorporate the new load modules created during the assembly process.

For users of the Interactive Query Facility (IQF), permanent modifications and corrections to problems encountered with IQF are provided from time to time with complete library replacements through a mod-level distribution of all IQF libraries. Whenever possible, IQF mod-level distributions will be coincident with IMS/360 mod-level distributions. (This does not mean, however, that the number of the mod-level will always match; for example, IQF Modification Level 3 could be the same as IMS/360 Modification Level 2, etc.)

Between releases of modification levels of IQF, users receive corrections to the system on PTF (Program Temporary Fix) tapes. These PTF tapes are automatically distributed to licensed users. They are distributed based on the number of problems discovered and fixed since either the most recent modification of the program product or of the PTF tape. The PTF tape contains symbolic updates for the IQF.SOURCE and IQF.GENLIB libraries. It also contains replacement load modules for the IMS2.LOAD library. These load modules correspond to the SOURCE library updates.

Between the releases of PTF tapes, fixes for problems reported by IQF users are distributed in the same manner as described for IMS/360 in the preceding discussion.

The solutions to individual problems provided between PTF tapes and modification levels are source symbolic updates. If the updates are for the IQF.SOURCE library, the user is instructed to perform a "temporary" update, assemble, and link edit the object module into either IMS2.LOAD or IMS2.RESLIB. If the symbolic change affects IQF.GENLIB, it may be necessary to perform a new IMS/360 system definition. If this is so, the instructions accompanying the change will specify that a new system definition is necessary. Typically, individual fixes only require assembly of updated source members and reprocessing of the affected IQF load modules to incorporate the new load modules created during the assembly process.

For user changes such as the addition or deletion of transaction codes or data bases, a system definition for a CTLBLKS type of generation is required. (See discussion of IMSCTRL macro-instruction in Chapter 3 of this manual.)

CHAPTER 3. IMS/360 SYSTEM DEFINITION

Through system definition a user structures IMS/360 to serve his data processing environment. This structuring includes a definition of communication line groups, lines, physical terminals, logical terminals, pools, and subpools. Pools and subpools are IMS/360 resources associated with switched communication lines. System definition also includes the definition of transactions, application programs, data bases, and various operating system interfaces. The subject of system definition is discussed in detail in this chapter.

Security maintenance is a supplementary process associated with system definition. Through use of the security maintenance utility a user defines the terminal and password security characteristics of a defined IMS/360 system. The subject of security maintenance is discussed in detail in Chapter 4 of this manual.

IMS/360 SYSTEM STRUCTURING CONSIDERATIONS

Before structuring the IMS/360 system, the user must consider the requirements and capabilities of IMS/360 in relation to his own requirements for the most expeditious operating environment. Consideration must be given to such things as the amount of main and secondary storage to be dedicated to IMS/360, the number of application programs to be run, and how many of these programs are to be run concurrently. A determination must be made of the transaction codes which are going to initiate the various application message processing programs and how many transaction codes are necessary. Each transaction code is assigned a class identifier, and each message processing region is assigned up to four class identifiers. These identifiers then determine which transactions are scheduled into which region. Region class assignments can be made according to application program size, response requirements, or other variables. The user must decide which transaction codes are to be of the response-type and which of the nonresponse-type. Decisions must be made concerning how many transaction codes cause data base updates, and how many are restricted to entry from a particular terminal. Consideration must also be given to how many communication lines and terminals of each type are to operate with the IMS/360 system.

In supplying answers to these questions, the user should consider the possible impact of his decisions on the operating capability of the system and the efficiency of its operation. The _IMS/360 System/Application Design Guide_ (SH20-0910) provides design information for evaluation of the various system capabilities and requirements.

IMS/360 allows the user to immediately process or batch online transactions. The user would be wise to consider whether some of the transaction codes he chooses can be queued up and can wait for processing on an as-required basis (batched). Time accounting is an example of the type which may fall into this category. Attendance reporting is another. Transaction codes of these types can be readily batched, because there is no necessity for an immediate-type response.

"Response-type" and "nonresponse-type" messages should not be confused with true "message types". Whenever a response-type message is entered from a terminal, the user should always be aware that his terminal locks and he must _wait for_ a response before he can enter another message from that terminal. The nonresponse-type message is entered and competes with other messages, on a priority basis, for system resources, but the terminal and communication line are always available for further message input until response.

The processing limit count feature of IMS/360 allows consideration of the number of messages which a reusable application program can process in one load of the program.

The scheduling limit count feature, in conjunction with the scheduling limit priority, does not say that a program will never be processed if there are always higher priority messages. It does say that, if the particular message is not called for execution by the time a certain number of messages have been received and queued, the scheduling priority is changed to a higher one. If there are messages with higher selection priorities in the system, of course, this message may still have to wait.

The total IMS/360 system must be considered by the user when structuring his system. The user must consider what the various types of transactions mean to the system, what the responses are, how many there are, etc.

Again, the user must consider the number of programs he wishes to be operating concurrently; how large these programs are and how many transaction codes they are operating against; and with how many terminals. These considerations affect the amount of core which is dedicated to IMS/360. Each application program, and the system, at any given time, may require additional amounts of space. Even the number of terminals concurrently being transmitted to has an effect upon the amount of core buffer space which should be allocated.

For example, assume a message is entered from a terminal. The application program for processing this message may send messages to each of six different terminals. This means that IMS/360 may require core buffer space for one line of the message output to each terminal. If the system is executing three application programs concurrently and trying to transmit to those six terminals, it would require allocation for 3 (number of application programs) times 6 (number of terminals) buffers in addition to the core required to hold the three application programs (in message processing regions).

The I/O units on which a system user chooses to place his message queues have special significance on system operations. If disk instead of drum is chosen, as an example, there is an effect on the number of messages run and consequently how many are processed. Since drum access is faster than disk, a greater number of messages can come in and go out of the system faster than when disk storage is used. Of course, there is more storage available on disk, but this is part of the tradeoff analysis to be made while structuring the system.

A number of factors must be considered when the Interactive Query Facility (IQF) feature is incorporated into the IMS/360 environment. These factors include the installation's requirements for security and privacy, the need to limit data base processing intent, the requirements for multiple processing of queries, etc. The IMS/360 Version 2 System Application and Design Guide (SH20-0910) provides further information on structuring the system environment to include IQF.

DEFINING THE IMS/360 SYSTEM

So far this discussion has centered on what the IMS/360 system user should consider in structuring the environment in which IMS/360 will execute. Now to be considered is how the IMS/360 system is tailored to the user's needs. This is accomplished with the IMS/360 system definition macro-instructions.

The IMS/360 requirements have been mentioned, and it is obvious that there must be modifications to make IMS/360 compatible with the operating system. These are accomplished through the use of three supervisor calls (SVCs) which must be made a part of the operating system. This is a simple matter for the system programmer to accomplish.

The System/360 used for the IMS/360 two-stage definition process must be at least a Model 40, with the "F" assembler and at least 128K storage. If an OS/VS1 or VS/2 system is used, the generation can be accomplished in a virtual region. The IMS/360 system definition must be run using the same version of the operating system under which the generated system will execute. The one Type 1 interregion communication SVC and the one Type 2 control SVC must be placed by the system user into (link-edited with) the Operating System nucleus of the system under which IMS/360 execution is to occur. In addition, the OSAM SVC modules for OPEN/CLOSE and the OSAM channel-end appendage must be moved to SYS1.SVCLIB (MFT, MVT, and VS1) or SYS1.LPALIB (VS2). If the complete Data Base - Data Communication System is to be executed, the OS privileged name table module in SYS1.LINKLIB must be modified to include the name DFSRRC00. This allows the online control program to be initiated through a START IMS command. (Use of the START, STOP, and MODIFY commands is explained in the Executing IMS/360 section of Chapter 4 of the <u>IMS/360 Operator's Reference Manual</u>, SH20-0913.) If the OS/VS1 or OS/VS2 system is to be used for either DB/DC or DB only, the privileged name table must have the name DFSRRC00 added to the list. All IMS/360 system data sets must be cataloged. If the complete Data Base/Data Communication System is to be executed and contains 7770-3 lines, the 7770-3 channel/abnormal end appendage and DEB builder (load 12 of the OSAM Type 4 SVC) modules must be link edited into SYS1.SVCLIB (MFT, MVT, and VS1) or SYS1.LPALIB (VS2).

The character set used by IMS/360 is the Assembler Language character set as defined in the SRL <u>IBM System/360 Operating System: Assembler Language</u> (GC28-6514).

IMS/360 SYSTEM DEFINITION MACRO-INSTRUCTIONS

The function of the IMS/360 system definition utility is to provide a user the capability of creating an IMS/360 system tailored to a specific environment.

System definition is a two-stage utility written primarily in the operating system assembler macro language. The first stage of the utility processes the user's specifications and checks them for validity, consistency, and compatibility. If the checks are successful (that is, if no errors are encountered), an operating system job stream (Stage II) is produced. When this job stream is executed, it produces an executable IMS/360 system, capable of operating in the environment and with the capabilities described in Stage 2 of IMS/360 system definition.

The user's input to the IMS/360 system definition utility contains four categories of information: (1) Data Base (Data Language/I) and Application, (2) Data Communications, (3) Interactive Query Facility (IQF), and (4) General System environment specifications. If a complete Data Base - Data Communication System is not desired, only the input describing the general system environment is required. This batch-only Data Base System allows Data Language/I data base operation in a batch processing region environment.

The following paragraphs describe the information required in each category:

<u>Data Base - Data Language/I</u>

This category of system definition includes information about application programs, transaction codes, and data bases used in the combined data communication - data base environment. Only application programs used for message processing or batch processing with data bases concurrently used for message processing need be defined. Only data bases used by application programs for message processing need be defined.

Application programs, as well as information concerning their operating characteristics, are defined and associated with the transaction codes they are to process.

Data base specifications name all Data Language/I data bases to be under control of the generated system and select logging and control blocks residency options.

Transaction codes must be defined. This includes transaction code name and scheduling characteristics. If a transaction is used for terminal conversation, conversational scratchpad area (SPA) specifications are required. The SPA is maintained in main storage or on disk as specified.

<u>Data Communications</u>

The system definition utility provides the specification of all supported telecommunication devices. This includes telecommunication line group, line, terminal, and terminal component specifications. Facility is also provided for incorporation of user-written modules to edit input and output messages.

If terminal conversation capabilities are desired, the number of concurrent conversations to be allowed must be specified. The user must also specify the number of conversational scratchpad areas (SPAs) to be maintained by the system, a subset of which may be defined as core resident.

<u>Interactive Query Facility (IQF) Feature</u>

Provided as an additive feature for IMS/360 users with the full Data Base/Data Communication system, IQF operates in a mode similar to a user's message processing application program. At message scheduling time, IQF is united with an input transaction (query) and a data base for execution in a message processing region. The association of transaction code, program and data base must be specified at system definition. This is accomplished through use of the system definition APPLCTN and TRANSACT macro-instructions described later in this chapter. One or more TRANSACT macro-instructions is required for each logical data base (or group of logical data bases) to be queried by a given transaction code. Each group of logical data bases accessible to IQF through a transaction code is described by a PSB (Program Specification Block).

If the optional IQF Index (QINDEX) capability is employed, the user must describe the data base(s) by the DATABASE macro-instruction, also described later in this chapter. It is not necessary to define either the IQF System Data Base or the IQF Phrase Data Base through use of the DATABASE macro. These data bases are automatically defined to IMS at system definition time.

General System Specifications

The user supplies information determining the IMS/360 - operating system interfaces such as SVC numbers, appendage names, etc. as well as:

- Library and message queue data sets
- Message processing region information, including size, number, and class
- Number and size of single and multiple line buffers
- Size of DMB, PSB, data base buffer, and various teleprocessing pools
- Job and SYSOUT class of the Stage 2 job stream and number of JOB cards produced by Stage 2

The control blocks as well as the executable code modules included in the generated system are only those necessary for execution. For example, if no 1050 terminals are specified, neither 1050 control blocks nor the modules for the 1050 are included in the user's system residence library (IMS2.RESLIB). Only those modules which are not part of the resident nucleus are moved or linked into IMS2.RESLIB.

Some IMS/360 system definition macro-instructions appear only once in the Stage 1 input stream, while others may be used multiple times in a hierarchical set arrangement to describe related user requirements. Figure 4 lists which macro-instructions are required and which may be used more than once. The end of this chapter provides examples of IMS/360 system definition.

Macro Instruction	Type of Definition: Data Communication - Data Base System	Type of Definition: Data Base Only (Batch) System
1 IMSCTRL	Required 1	Required 1
2 IMSCTF	Optional 1	Optional 1
3 SPAREA	Optional 1	N/A *
4 BUFPOOLS	Required 1	N/A
5 MSGQUEUE	Required 1	N/A
6 IMSGEN	Required 1	Required 1
7 DATABASE	Required n	N/A
8 APPLCTN	Required n	N/A
9 TRANSACT	Required n	N/A
10 LINEGRP	Required n	N/A
11 LINE	Required n	N/A
12 CTLUNIT	Optional n	N/A
13 STATION	Optional n	N/A
14 TERMINAL	Required n	N/A
15 NAME	Required n	N/A
16 POOL	Optional n	N/A
17 SUBPOOL	Optional n	N/A

* N/A - Not Allowable

Figure 4. Complete IMS/360 system definition macro-instruction

Three groups of macro-instructions form hierarchical sets which are required for the description of user resources. One group (Figure 5) describes application programs and their related transaction codes. The other two describe communication line groups, communication lines, and associated physical and logical terminals (Figures 6 and 6A).

Note: All macro-instruction positional or keyword operand values which are names must start with alphabetic characters. When defining PSB names, logical terminal names, and transaction codes, care should be taken in the use of keywords, synonyms, and null words as defined in the IMS/360 Operator's Reference Manual. Note also that during control blocks or nucleus generation only the IMSCTRL card statement need be changed.

Hierarchical set requirements are shown in the following:

Macro-Instruction	Number Per Set	Purpose
APPLCTN	1	Names application program. Delimits this set of macro-instructions.
TRANSACT	n	Names transaction codes which are to be processed by the above application program.

Figure 5. Application description macro-instruction set

Each application program described with an APPLCTN macro-instruction should be used for message or batch-message processing. Application programs used only for batch processing are not to be described by APPLCTN macro-instructions.

Macro-Instruction	Number Per Set	Purpose
LINEGRP	1	Names collection of terminals with like attributes. Delimits this set of macro-instructions.
LINE	n	Provides address of line and delimits terminals on same line.
CTLUNIT	n	Provides terminal control unit address and attributes. Delimits terminals on same control unit.
TERMINAL	n	Provides physical terminal data and delimits logical terminal name.
NAME	n	Provides logical terminal names.

Figure 6. Communication terminal description macro-instruction set. Note: When the TERMINAL statement is used to describe a switched physical terminal, no NAME statement may be associated with it.

Macro-Instruction	Number Per Set	Purpose
POOL	n	Defines a pool of switched terminals and delimits pool descriptions
SUBPOOL	n	Describes a set of logical terminals and delimits subpools
NAME	n	Provides logical terminal name

Figure 6A. Switched communications terminal set

During the system definition process, all resources, except communication lines and message-processing regions, are generated in a 'started' mode; that is, a /START command is not required to make the resource operational.

IMSCTRL Macro

The IMSCTRL macro-instruction is used to describe the basic IMS/360 control program options and the operating system environment under which IMS/360 will operate. The IMSCTRL macro-instruction is always required. For the batch Data Base System definition, the MAXIO and MAXREGN operands need not be specified. This macro-instruction should be the first statement in the system definition control card deck which is input to system definition Stage 1 and can only occur once.

```
         IMSCTRL    SYSTEM=( {VS/1 }  {CTLBLKS} )
                           ( {VS/2 }  {NUCLEUS} )
                           ( {MFT  }, {BATCH  } )
                           ( {MVT  }  {ON-LINE} )
                           ( {M65MP}  {ALL    } )

                    [,MAXIO=number]

                    [,MAXREGN= (number, size, job class,
                              msgclass) ]

                    [,MCS=(number[,number,...]) ]

                    [,DESC=number]

                    [,MAXCLAS=number]
```

Note: The only other macro-instructions needed for a batch Data Base System are the IMSCTF and IMSGEN macro-instructions.

- Operand Field:

SYSTEM=

specifies the OS programming system configuration and the type of IMS/360 system to be generated. The programming system configuration under which IMS/360 will operate may be multiprogramming with a fixed number of tasks (MFT), multiprogramming with a variable number of tasks (MVT), multiprogramming with a fixed number of tasks in a virtual system (VS/1), or multiprogramming with a variable number of tasks in a virtual system (VS/2). The M65MP version of MVT is specified as "M65MP". If programming system configuration is not specified, the value "MVT" is assumed.

The following values specify the type IMS/360 generation to be performed:

CTLBLKS - Generates only a composite control block module for use within an IMS/360 nucleus in the IMS/360 control program region. This function is useful if an alternate set of IMS/360 control blocks is desired. The Composite Control Blocks module (DFSIBLKx) resulting from this option must be link edited with an existing control program nucleus (DFSCNUCx).

NUCLEUS - Generates a composite IMS/360 nucleus for the IMS/360 control program region as well as a composite control block module which is included in the generated nucleus. This parameter is useful when generating an alternate nucleus including new system features, such as additional terminal support or conversational support.

BATCH - Moves required modules from the IMS/360 distribution libraries to the user's libraries; generates system procedures and a batch Data Base System. This value must be specified if the user has not ordered the IMS/360 Data Communication feature.

ON-LINE - Moves required modules from the IMS/360 distribution libraries to the user's libraries, generates system procedures, and generates those modules comprising the CTLBLKS and NUCLEUS options for a Data Communication - Data Base System.

ALL - Combines the BATCH and ON-LINE options.

MAXIO=

specifies the maximum number of terminal I/O requests, message queue requests, and Data Language/I data base requests which may be in process in the IMS/360 control program region at any one time. A recommended minimum number is the sum of the values specified in the MAXREGN parameter and the number of communication lines divided by 2. The value should never be less than the value specified in the MAXREGN parameter. The operand may be omitted, thus causing a value to be calculated by the formula:

$$MAXIO = \left(\frac{\text{number of communication lines}}{2} + MAXREGN + 2\right)$$

MAXREGN=

specifies the maximum number of regions or partitions which IMS/360 is expected to support at any one time. This value includes batch message processing regions as well as message processing regions. Default value is 3 (regions). This operand is also used to specify region size (default value is 52K), region job class (default value is Class A), and job message class (default value is Class A). Note that region size (must be expressed in terms of xxK), region job class, and job message class must conform to the operating system job control language specifications. This operand is ignored for BATCH system definitions.

MCS=

specifies the routing code to be assigned to the IMS/360 system console if multiple console support (MCS) has been included in the operating system. If omitted, the master system console will be used.

DESC=

specifies the message descriptor code to be assigned to the IMS/360 system console messages if MCS support has been included in the OS generations. If omitted, no descriptor will be assigned.

See the WTO macro and Appendix C of Operating System Supervisor and Data Management Macro SRL, Form No. GC28-6647, for a detailed description of the above two operands. Parameters should be defined as required for the ROUTCDE and DESC keywords for the WTO macro.

MAXCLAS=

specifies the maximum number of unique transaction code classes which are to be supported by Message Class Scheduling. Any transaction code class specified on either the APPLCTN or TRANSACT macro statement must not exceed this value. Valid specification of this operand is a decimal number from 1 to 255. Default value is 1.

Example of IMSCTRL macro-instruction operands when system option is MVT, 50 communication lines and 3 message regions, message region job class and msgclass are A, and both an online and a batch system are to be defined.

```
IMSCTRL SYSTEM=(MVT,ALL),MAXIO=27,MAXREGN=(3,52K)
```

The job class and msg class for all message regions are A. A message region size of 52K is used.

If system is MP65, an alternate online nucleus is to be built (normally assuming a previous IMS/360 online system definition), and MAXREGN class information is to be class B, the following operands would be defined:

```
IMSCTRL SYSTEM=(M65MP,NUCLEUS),MAXREGN=(3,,B,B)
```

The MAXIO value is calculated to 27.

IMSCTF Macro

The IMSCTF macro-instruction defines parameters to the IMS/360 control facility.

```
IMSCTF    [SVCNO=(type1, type2, type4)]

          [,APNDG=([osam suffix][,7770 suffix])]

          [,CPLOG=checkpoint log frequency]

          [,CORE=(minimum,maximum)]
```

- Operand Field:

SVCNO=

specifies operating system SVC numbers reserved for use by the generated IMS/360 system. Entered values may range from 128 to 255 and must be mutually exclusive. Default value is 253 for Type 1, 254 for Type 2, and 255 for Type 4. The Type 1 SVC is used for IMS/360 interregion communication. The Type 2 SVC is used for exclusive control handling of IMS/360 resources. The Type 4 SVC is used for OSAM. Independent of the operands of the IMSCTRL macro, the user must specify values for all SVC numbers.

APNDG=

The first value specifies the OSAM channel-end appendage suffix value used by the IMS/360 OSAM access method. The value may range from A8 to Z9. Default value is Z9. The second operand is the channel/abnormal end appendage suffix for the 7770-3 if 7770-3 lines are part of the generated system. The value specified may range from A8 to Z9 and must not be the same as the OSAM suffix. The default value is Z8. The complete appendage name(s) will be IGG019xx.

CPLOG=

specifies the number of IMS/360 system log operations between automatic IMS/360 checkpoints. The permitted value ranges from 500 to 32767. Default value is 1000.

CORE=

defines the amount of main storage available to the IMS/360 control facility ENQ/DEQ routines via GETMAIN from subpool 252. The first parameter states a minimum main storage availability; the second states maximum main storage availability. Values may range from 0 to 32K. The default values are 500 bytes minimum and 1000 bytes maximum. For a detailed calculation of this value, the reader is referred to the _IMS/360 System/Application Design Guide_ (SH20-0910).

Example of IMSCTF macro-instructions, when Type 1 SVC is 241, Type 2 SVC is 254, and Type 4 SVC is 235, OSAM channel end appendage is IGG019Z6, 7770 appendage is IGG019Z0, and CPLOG=2000, is the following:

```
IMSCTF  SVCNO=(241,254,235),APNDG=(Z6,Z0),CPLOG=2000
```

SPAREA Macro

The SPAREA macro-instruction defines the maximum number and maximum size of conversational scratchpad areas (SPAs) to be maintained by the IMS/360 system. The total number of disk and main storage scratchpads can not exceed 4000. When the user defines a conversational transaction through the TRANSACT macro, the SPA for that transaction will normally be smaller than the maximum defined here. However, it can never exceed the maximum specified in this macro-instruction. This macro-instruction should be omitted if no conversational transactions are subsequently specified through TRANSACT macro-instructions. A warning message is generated if transactions are subsequently defined in this system definition using SPAs (conversational transactions) and this macro-instruction is omitted.

SPAREA	[CORE=	(number, size)]
	[,DASD=	(number, size)]

• Operand Field:

CORE=

specifies the number and size of the main storage SPAs. If this operand is omitted, no main storage SPAs are maintained. The number of main storage SPAs determines the maximum number of terminal conversations in which incore SPAs are used. The size of a main storage SPA can not exceed 32767 bytes.

DASD=

specifies the number and size of the disk-maintained SPAs. The size of a single disk SPA cannot exceed the track length of the device on which the SPA resides. One disk SPA must exist for each concurrent conversation where data associated with each conversation is to be maintained on a disk SPA.

If the user wishes to allow 20 concurrent conversations, with 5 maintained in main storage the maximum size of which is 200 bytes, and with 15 maintained on direct access the maximum size of which is 1000 bytes, the following operands are appropriate:

```
SPAREA  CORE=(5,200),DASD=(15,1000)
```

If the Interactive Query Facility (IQF) has been incorporated into the user's system, the SPA for IQF should be large enough to hold the maximum IQF query which may require a full file search; an additional 34 bytes are required for system usage. (See discussion of SPA for the TRANSACT macro-instruction.)

BUFPOOLS Macro

With the BUFPOOLS macro-instruction, the user specifies default main storage buffer pool sizes for the online control program system region. The sizes and options specified are used unless otherwise expressly stated for that buffer or pool at online system control program execution time through the PARM field.

```
BUFPOOLS  [PSB= (size, {RETAIN | FLUSH})]
          [,DMB= (size, {RETAIN | FLUSH})]
          [,DBASE=  size]
          [,GENERAL=  size]
          [,FORMAT=  size]
          [,FRE=  number]
          [,COMM=  size]
```

- Operand Field:

PSB=

specifies the size of the PSB control block pool. The functions provided by the RETAIN and FLUSH parameters are no longer required. They are retained only for compatibility with previous user system definition source decks. Default value is 10000 (bytes). Maximum allowable specification is 65535 bytes.

DMB=

specifies the size of the DMB control block pool. The functions provided by the RETAIN and FLUSH parameters are no longer required. They are retained only for compatibility with previous user system definition source decks. Default value is 10000 (bytes). Maximum allowable specification is 65535 bytes.

DBASE=

specifies the size of the common data base buffer pool. This pool supplies buffers for all data bases used in the combined data base - data communication environment in the online control program system region. Default size for DBASE is 10000 (bytes).. Maximum allowable specification is 65535 bytes.

GENERAL=

specifies the size of the general buffer pool area used dynamically by the IMS/360 control program and its various action modules for producing system messages in response to communication activity. Default size for GENERAL is 5000 (bytes). GENERAL pool size must be large enough to accommodate main storage SPAs, one disk SPA plus 2000. Maximum allowable specification is 65535 bytes.

FORMAT=

specifies the size of the message format block pool. Minimum specification is 2000 bytes; maximum specification is 16000000 bytes. Default value is 10000 plus the number of lines requiring format services times 1000 (10000+(LINES*1000)) bytes. This keyword parameter is ignored if no unit types requiring format services are included in the defined system.

FRE=

specifies the number of fetch request elements that are to be used for loading message format blocks into the message format block pool. The number specified must be in the range of 10 to 65535. Default value is 10 plus the number of lines requiring format services: maximum default is 20.

COMM=

specifies any <u>additional</u> space to be added to the value calculated for the communications line buffer pool. This size defaults to 0 unless 3270 is included in the system. If 3270 is included, this size defaults to 1000 bytes plus 1000 bytes for each 3270 line. Maximum allowable specification is 65535 bytes.

The reader is referred to the <u>IMS/360 System/Aplication Design Guide</u> (SH20-0910) for the details of calculation for the various buffer pool sizes.

In addition to those buffer pools which the system user explicitly defines, an additional buffer pool is created by IMS/360 system definition for communication device line control operations. The buffer requirements for each communication line, as defined by the communication terminal type and model attached to that line, are used in summation to define the communication line buffer pool size.

An example of the BUFPOOLS macro-instruction operands with a PSB pool of 10,000 bytes and the FLUSH option, with a DMB pool of 10,000 bytes and the RETAIN option, with a data base buffer pool of 20,000 bytes, and a general pool of 2,000 bytes, is:

```
BUFPOOLS     PSB=10000,DMB=(10000,RETAIN),DBASE=20000,     X
             GENERAL=2000
```

MSGQUEUE Macro

The MSGQUEUE macro-instruction defines the characteristics of the three message queue data sets desired by the user.

```
MSGQUEUE     DSETS=  ( [ {2301}   {2301}   {2301} ] )
                         {2303}   {2303}   {2303}
                         {2311} , {2311} , {2311}
                         {2314}   {2314}   {2314}
                         {2305}   {2305}   {2305}
                         {3330}   {3330}   {3330}

             [ ,RECLNG=   ( {size1} , {size2} ) ]
                            { 192 }   { 576 }

             [ ,BUFFERS=  ( {nbr}   [,size3] ]
                            { n }

             [ ,SHUTDWN=    {nbr} ]
                            { 30}
```

- Operand Field:

DSETS=

specifies the device types on which the three message queue data sets will reside (IMS2.QBLKS, IMS2.SHMG, and IMS2.LGMSG, respectively). If all three data sets will reside on the same device type, only the first parameter need be specified.

RECLNG=

specifies the logical record lengths for the short and long message queue data sets, respectively. The minimum value for size1 and size2 is 192. Size2 must be equal to or greater than size1. If either size1 or size2 is not specified, default values of 192 and 576, respectively, will be assigned.

SIZE2 minus 48 is the maximum size allowed for segments processed by the message format service facility.

BUFFERS=

specifies the number of buffers to allocate for message queue management and the block size to be assigned for all three message queue data sets. If 'nbr' is not specified, a default value equal to four plus one-tenth the sum of the number of logical terminal names plus the number of transaction codes is used. If size3 is not specified, it will be calculated by the following formula. If the defaults of size1 (192) and size2 (576) are used, then size3 will equal 576. Maximum allowable number specification is 255.

SIZE3 = ((SIZE2 + SIZE1 - 1)/SIZE1) * ((SIZE1 + 47) /48) * 48

This calculation can leave null space at the end of the short and long message blocks.

SHUTDWN=

specifies the number of records to be reserved in each data set to allow the system to automatically shut down if the data set becomes filled with unprocessed messages.

All sizes specified will be rounded up to a multiple of 4.

An example of the MSGQUEUE macro-instruction with all data sets on 2314 is:

```
MSGQUEUE     DSETS=(2314),BUFFERS=(9,1152),SHUTDWN=200
```

This example indicates that the default sizes for short and long messages are adequate, but that two long message records should share a block. With the number of buffers indicated, a message flow of 6 long messages and 11 short messages may be concurrently processed against 23 destinations without requiring I/O operations against the queue data sets. (See Message Queue Space Allocation in Chapter 3.)

IMSGEN Macro

The IMSGEN macro-instruction is used to specify the data sets, volumes, and I/O devices required for the definition process, the system definition output options.

The IMSGEN macro-instruction must be the last macro-instruction in the Stage 1 input stream. It must be followed immediately by an assembler END statement.

```
IMSGEN   [ASM= {F}]
               {H}

         [,ASMPRT= {ON }]
                   {OFF}

         [,JOBCTL=(nbr,jobclass,output class,
                   job accounting,step accounting)]

         [,MACLIB= {ALL    }]
                   {UTILITY}

         [,OBJDSET= name]   [,USERLIB=name ]

         [,PROCLIB=({NO } [n ])]
                    {YES}

         [,PAGE=     {YES}]
                     {NO }

         [,PRTY=]

         [,SUFFIX= character]

         [,LKPRT= ({MAP }    ,  LIST )]
                   {XREF}

         [,LKSIZE=( value1 [,value2] )]

         [,LKRGN= size ]
```

• Operand Field:

ASM=

defines whether OS component Assembler F or OS program product Assembler H Job Control Language is to be produced for the Stage II assembly steps. Default value is F.

ASMPRT=

specifies the assembler print options for those assembler job steps produced by IMS/360 system definition. ON specifies that assembly listings are to be generated; OFF, that assembly listings are not to be generated. OFF is the default value.

JOBCTL=

nbr

specifies the number of steps per job card to be produced by system definition Stage 1 for execution of Stage 2. Maximum allowable value is 10; default value is 5. Regardless of the value specified, a job card will be produced for the beginning of the link edit steps and for the nucleus link edit step (if applicable).

jobclass

specifies the jobclass to be generated on the Stage 2 job card. Default value is A.

output class

specifies the output class to be generated for the Stage 2 JCL. Default value is A.

job accounting and/or step accounting

specifies job and/or step accounting data to be placed in the Stage 2 JCL. The length of the accounting data may not exceed 50 bytes. If job accounting data is specified, a programmer name of IMS is provided.

MACLIB=

specifies the macro requirements of the generated system. This operand is used by system definition to determine the macro-instructions to be moved into the IMS2.MACLIB data set. This operand is inoperative if the IMSCTRL macro SYSTEM operand specifies CTLBLKS or NUCLEUS. Default value is UTILITY.

UTILITY - Only those macros necessary for PSB and DBD generation are copied to the IMS2.MACLIB data set.

ALL - All IMS/360 macros are copied to the IMS2.MACLIB data set, except those necessary for an IMS/360 system definition.

OBJDSET=

specifies the name of a cataloged partitioned data set into which assembler object modules are placed during Stage 2 of IMS/360 system definition. If not supplied, these modules are placed in IMS2.OBJDSET.

USERLIB=

should be supplied if any user-furnished routines, such as message edit routines, are to be included in the generated IMS/360 nucleus. If this operand is omitted and user edit routines are indicated, the library containing the routines is assumed to be IMS2.RESLIB. The data set must be a cataloged data set.

PROCLIB=

specifies whether system procedures are to be generated. If PROCLIB= NO, no system procedures are generated. Default Value is YES. The second parameter (n) defines data base buffer size for a DL/I batch region. The entered value is expressed in 1K increments.

When a number (n) is provided:

1. Place the entered value as a default buffer requirement on the DLIBATCH, IMSCOBGO, and IMSPLIGO cataloged procedures.

2. Place the entered value in the batch nucleus to be used by DL/I as the buffer requirement, when none is provided by the application program execution.

If a number is <u>not</u> provided, the value (n) representing 7K is placed in the batch nucleus.

PAGE=

defines whether (PAGE=YES) or not (PAGE=NO) the terminal paging feature is to be included in the defined system.

PAGE=YES is ineffective if no 2260 or 2265 terminals are included in the defined system. If 3270 devices are defined, the paging capability is included.

If the Interactive Query Facility has been incorporated into the user's system, the user must include video-paging if the answer to his query will generate more than one page to a video-device.

PRTY=

specifies the priority placed on the job cards for IMS/360 system definition Stage 2 jobs. Default is priority zero.

SUFFIX=

specifies the suffix character appended to the generated composite control blocks, nucleus, and security directory module names upon placing them into the IMS2.RESLIB data set. The IMS/360 online nucleus name always starts with DFSCNUC. This suffix character supplies the eighth character of the nucleus name. If the suffix character equals zero, the nucleus name is DFSCNUC0. This concept applies for composite control blocks and security directory module names as well. This concept allows the system user to generate multiple IMS/360 systems for use in one environment where the characteristics of each system vary. Default value is zero.

LKPRT=

specifies linkage editor print options for those linkage editor job steps produced by IMS/360 system definition.

Value	Print Option
XREF	Cross-reference table (XREF includes the MAP option)
LIST	List of control statements in card image format
MAP	Module map

If this parameter is omitted, only linkage editor error messages, if any, are printed. For a more detailed description of these options, see the publication IBM System/360 Operating System: Linkage Editor, Form GC28-6538.

LKSIZE=

specifies the value(s) to be placed in the SIZE parameter for use by the level F linkage editor.

where:

value1

specifies the maximum number of bytes of main storage available to the linkage editor. This value can be specified in the form n (where n represents the actual number of bytes of main storage, not to exceed 99999) or nk (where n represents the number of 1K blocks of main storage, not to exceed 9999K).

value2

specifies the maximum amount of value1 that is to be used as the load module buffer and is expressed in the form n (where n may not exceed 9999) or nk (which should not exceed 100K).

Default values are not supplied for the SIZE parameter specifications. If value1 and value2 are omitted, the SIZE parameter is not specified. For a more detailed description of the SIZE option, see the publication IBM System/360 Operating System: Linkage Editor (GC28-6538).

LKRGN=

specifies a region size value to be placed on the generated EXEC card for execution of the level F linkage editor. This parameter may be specified as a decimal number (not to exceed 999999) or in the form nk (not to exceed 99999K). The default value is 130K.

DATABASE Macro

The DATABASE macro-instruction defines all data bases to be used under control of the IMS/360 online control program. One DATABASE macro-instruction must be specified for each HSAM, HISAM, and HDAM data base to be used under control of the IMS/360 online control program. Two DATABASE macro-instructions are required for a HIDAM data base, one for the INDEX DBD, and one for the HIDAM DBD.

	DATABASE	[INDEX ,] DBD=name

The positional operand, INDEX, indicates that this is a DATABASE statement for the INDEX DBD of a HIDAM data base.

- Operand Field:

DBD=

specifies the name of the data base description (DBD). At execution time, the DBD must have been processed by the block builder utility program and must exist as a member in the partitioned data set named in the IMSACB DD statement (IMS2.ACBLIB). The first character of the name must be a valid alphabetic character. If the DBD has not been processed by the block builder utility program, and is, therefore, not present in the ACBLIB data set, then the data base will be locked at execution time. This operand is required.

If the Interactive Query Facility (IQF) has been incorporated into the user's system and the QINDEX capability is employed, two or more DATABASE macro-instructions must be added.

APPLCTN Macro

The APPLCTN macro-instruction describes the program resource requirements for application programs which run under the control of the IMS/360 online control program system region. When combined with one or more TRANSACT macro-instructions, the set defines the scheduling and resource requirements for an application program. The APPLCTN macro-instruction describes only programs which operate in message processing regions or batch message processing regions. Application programs which operate in batch processing regions are not to be described by the APPLCTN macro-instruction.

```
|        | APPLCTN  | [DOPT,]
|        |          |
|        |          | PSB = name of PSB
|        |          |
|        |          |  [            {BATCH}                 {class} ]
|        |          |  [,PGMTYPE=   { TP  }   [,OVLY]      ,{  1   } ]
|        |          |
|        |          |  [      {YES} ]
|        |          |  [,IQF= {NO } ]
```

(In the format box, the defaults TP, 1 and NO are underlined.)

• Operand Field:

DOPT

specifies that this PSB is to be located dynamically.

If this parameter is specified, the following actions are taken during the execution of the IMS control region:

1. Initialization does not perform a BLDL on this PSB. Thus the PSB need not be present in any data set defined by the ACBLIB DD card until it is actually required to process a transaction.
2. Each time the program called by this PSB is scheduled, a BLDL will be performed, thus locating the latest copy of the PSB.
3. When the program terminates, the PSB will be deleted from the PSB pool as part of the termination process.

In order to provide greater flexibility in the user's control of the library containing the dynamic PSBs, the following restriction is imposed: The PSB must reside in a library other than IMS2.ACBLIB and be concatenated to it. If the BLDL performed at the scheduling of the PSB determines that it resides in the first concatenation of the IMSACB DD card set, the PSB will be stopped and an error message sent to the master terminal. The PSB will not be scheduled.

PSB=

specifies the name of the PSB. The first character of the name must be a letter. If PGMTYPE is TP, this must also be the program name.

PGMTYPE=

defines application program characteristics. TP implies that IMS/360 schedules the program when messages exist in the system which are processed by the program. It is also the default value. A BATCH program may utilize Data Language/I in the IMS/360 control program system region and may reference the message queues. If BATCH is coded, all TRANSACT macro-statements which follow are assigned a normal and limit priority value of zero. The OVLY value indicates that the application program uses overlay design. If OVLY is specified for application programs which do not use overlay design, it results in unnecessary processing overhead in handling the overlay supervisor. If OVLY is not specified when required, it will cause unnecessary core storage to be used in the message processing region and may eventually cause the message region control program to be abnormally terminated.

The third parameter of the PGMTYPE= keyword specifies the class to which the transaction codes specified in the following TRANSACT macro statement(s) are to be assigned. This parameter must be a decimal number from 1 to 255. The default value is 1. If the transaction code class is to be specified in the individual TRANSACT macro statement(s), this parameter need not be coded. If the transaction code class is specified in both the APPLCTN and TRANSACT macro-statements, the APPLCTN macro specification is ignored for the transaction code for which specification is made on the TRANSACT macro.

If IQF=YES is specified, TP is assumed for PGMTYPE.

IQF=

specifies whether or not the program invoked by transaction codes associated with this transaction is the Interactive Query Facility (IQF). YES must be specified for IQF.

TRANSACT Macro

The TRANSACT macro-instruction may be used one or more times with each APPLCTN macro-instruction. It specifies the transaction codes which cause the application program named in the preceding APPLCTN macro-instruction to be scheduled for execution in an IMS/360 message processing region. It also provides the IMS/360 control program with information which influences the application program scheduling algorithm and may define a message editing routine.

```
| TRANSACT |   CODE = transaction code
|          | [ ,PRTY = (normal, limit, limit
|          |           count) ]
|          | [ ,MSGTYPE=( {MULTSEG, NONRESPONSE}, class) ]
|          |             {SNGLSEG, RESPONSE   }
|          |   [ ,PROCLIM = (count, seconds) ]
|          | [ ,INQUIRY = {NO } or INQ=  {NO } ]
|          |              {YES}          {YES}
|          | [ ,MODE=   {SNGL} ]
|          |            {MULT}
|          | [ ,EDIT= ( {UC }, name ) ]
|          |            {ULC}
|          | [ ,SPA=( size[, {CORE}] ) ]
|          |                 {DASD}
```

- Operand Field:

CODE=

specifies the one- to eight-character alphameric transaction code. The first character of transaction codes and logical terminal names must be either a letter or a digit. Transaction codes and NAME macro-instructions must comprise a set of values, each of which is unique in the system. That is, transaction codes and logical terminal names collectively may not contain duplicates. The CODE operand is required.

PRTY=

specifies the priorities at which this transaction code contends for scheduling selection with other transaction codes being processed by the system. The normal and limit values may range from 0 through 14 and are coded as one or two numeric digits. The limit count value may range from 1 through 65535. The normal field is the priority assigned to this transaction when the number of input transactions enqueued and waiting to be processed is less than the value specified in the limit count field. The limit priority field is the priority to which this transaction code is raised when the enqueued count of waiting input messages is equal to or exceeds the value specified in the limit count field. Once the priority of this transaction has been raised to the limit priority, it is not reduced to the normal priority until all enqueued messages for this transaction code have been processed by the program specified in the preceding APPLCTN macro-instruction, that is, the input queue is empty. If the limit priority feature is not desired for this transaction code, the normal and limit values should be equal and the limit count value 65535. Default values for normal, limit, and limit count are (1,1,65535).

MSGTYPE=

defines the transaction code as being single segment (SNGLSEG), or multiple segment (MULTSEG), and whether the communication line from which the transaction was entered will be held until a response is received (RESPONSE) or not (NONRESPONSE). It also specifies the time at which an incoming message is considered complete and available to be routed to an application program for subsequent processing. MULTSEG means that the incoming message may be more than one line in length and is not to become eligible for scheduling to an application program until an end-of-message indication is received. SNGLSEG specifies that the incoming message is always only one line in length and becomes eligible for scheduling when the terminal operator depresses the EOB key (carriage return if the Auto EOB feature is present) or equivalent. NONRESPONSE specifies that, upon completion of the input message, single or multiple segment, the terminal is to accept further input without waiting for the completed input message to be processed. RESPONSE specifies that, upon completion of the input message, single or multiple segment, the terminal and the communication line to which it is attached are to accept no further input until the program specified in the APPLCTN macro-instruction has been scheduled, has processed the input message, and has sent an output message to the input terminal. The third parameter of the MSGTYPE= keyword specifies the class to which this transaction code is to be assigned. This parameter must be a decimal number from 1 to 255. The default value is 1. If the transaction code class is specified in the APPLCTN macro statement, this parameter need not be specified. If the transaction code class is specified in both the APPLCTN and TRANSACT macro statements, the APPLCTN macro statement specification is ignored for this transaction. The default value is (MULTSEG, NONRESPONSE,1). _Note_: An input transaction from a 7770-3 line, 2741 line, or 2740 non-Station Control will be treated as a RESPONSE-type transaction without regard to the type specified for this operand. Note also that MSGTYPE=RESPONSE is ignored during online processing for all terminals except 2740 Model 1 with Station Control feature, and 1050.

If IQF=YES is specified in the associated APPLCTN macro, MULTSEG and NONRESPONSE are assumed. If the user specifies otherwise, it is ignored.

PROCLIM=

specifies the number of messages of this transaction code a program can process in a single scheduling and the amount of time (in seconds) allowable to process a single transaction (or message). The seconds field specifies a numeric value in seconds which may range from 1 to 65535 and represents the maximum CPU time allowed for each message to be processed in the message processing region. The count field specifies the maximum number of messages which are provided to the application program by the IMS/360 control program for processing without reloading the application program. The count field value may range from 1 through 65534. Code the count field value at 65535 if no limit is to be placed upon the number of messages processed at a single program load. Default value for the PROCLIM operand is (65535,65535).

The seconds value assigned is used for the purpose of controlling application program erroneous looping. No attempt need be made to optimize the seconds value for program-transaction execution time. However, the seconds time value assigned should not be less than the expected per-transaction execution time. If the scheduled application program exceeds the product of seconds and count, the application program will be terminated abnormally.

The count value assigned is used to determine how many messages an application program is allowed to process in a single scheduling cycle, that is, program load. When the application program has requested and received the number of messages indicated in the count value, it will receive a 'no more messages' indicator upon any subsequent requests from the IMS/360 control program. IMS/360 may, in fact, have other messages enqueued for the application program. Upon receiving the indication that no more messages are available, it is the responsibility of the message processing application program to terminate, thus making available for rescheduling the region which it occupied. This feature makes it possible for IMS/360 to allow scheduling of higher priority transactions which may have entered the system while the previous transactions were in process. In addition, if any equal priority transactions are enqueued, they will become eligible for scheduling on a first in, first out (FIFO) basis.

The PROCLIM parameter is ignored for transactions processed by a batch-message processing (BMP) program. BMP transactions are assigned a count of 65535 and a time of 24 hours at program scheduling time.

INQUIRY= or INQ=

This operand should be specified only for those transactions which when entered will not cause a change to any data base. Programs are prohibited from issuing ISRT, DLET, or REPL calls to a data base when scheduled to process a transaction defined as INQUIRY=YES.

Since switched terminals which have signed on for the INQUIRY LTERM or 2741, 7770, or 33/35 TWX terminals are not allowed to enter transactions which update a data base, the INQUIRY=YES parameter must be specified for transactions which will be entered by these terminals.

If an attempt to enter a transaction from one of these terminals is made that has not specified INQUIRY=YES, the transaction is rejected and a message 'DFS067 Terminal Security Violations' will be sent to the inputting terminal.

To prohibit the use of DEFINE or DELETE (phrase definition functions) for a transaction code, INQ=YES should be coded. Otherwise, if DEFINE and DELETE functions are desired, INQ=NO should be coded (or defaulted to by omitting INQ=).

MODE=

determines whether or not data base buffers will be written to direct access (flushed) upon each request for a new message (SNGL) by the processing program or upon program termination (MULT). Default value is MULT. This operand affects emergency restart. If MODE=SNGL, emergency restart will only reprocess the last message, even if one or more messages were scheduled and processed by a single load of an application program. Otherwise, emergency restart will reprocess all the messages that were scheduled and processed by the single load of an application program. Converstationl transactions must be defined as SNGL.

SNGL is forced for IQF applications.

EDIT=

is the name of a user-supplied transaction input edit routine to edit messages prior to the program receiving the message. This name must begin with a letter. The specified edit (load module) must reside on the USERLIB data set prior to IMS/360 system definition Stage 2 execution. The first parameter of this operand defines whether the transaction is to be translated to uppercase (UC) or uppercase/lowercase (ULC) as entered from the terminal prior to presentation to the processing program. The default is UC.

The EDIT operand is ignored for IQF since a default edit routine is provided for all IQF transactions.

SPA=

specifies where the conversational scratchpad will be kept (CORE or DASD), as well as the size of the scratchpad area (SPA) required for this transaction. Default value is DASD. The inclusion of this operand defines this transaction as a conversational transaction and assumes the presence of the SPAREA macro-instruction in the system definition.

If the user's system includes the Interactive Query Facility (IQF), a scratchpad must be specified for all IQF transactions. The user may specify either CORE or DASD. The size of the scratchpad is 34 bytes plus the length of the longest query for which a full file search may be required.

LINEGRP Macro

The LINEGRP macro-instruction defines the beginning of a set of communication line, communication terminal control unit, physical terminal, logical terminal pool, logical terminal subpool, and logical terminal description macro-instructions which include LINE, CTLUNIT, TERMINAL, POOL, SUBPOOL, and NAME. These sets are used to describe the user's telecommunications system. The LINEGRP macro-instruction is used to begin a description of one or more lines of the same type, over which the same type of terminal communicates.

		ALL LINE GROUPS
	LINEGRP	DDNAME= name or list of names ,UNITYPE= { 3270 [,LOCAL] | TWX | 1030 | 1050 | 2260 [,LOCAL] | 2265 | 2740 [,NOSTACTL] | 2741 | 2770 | 2780 | 2980 | 7770 | DISK | PUNCH | PRINTER | READER | TAPE | SPOOL } [,EDIT= module name]
		ALL NONSWITCHED LINE GROUPS ONLY
		[,FEAT=([{NONSWITCHED | SWITCHED}] [{,POLL | AUTOPOLL}])]
		ALL SWITCHED LINE GROUPS ONLY
		,FEAT=(SWITCHED)
		2780 LINE GROUPS ONLY

```
[          {EBCDIC }  ]
[ ,CODE=   {USASCII}  ]
[          {TRANSC }  ]

[ ,MODE=   ( {A} , {A} ) ]
[          ( {B}   {B} ) ]
```

2741 NONSWITCHED LINE GROUPS ONLY

```
[          {EBCDIC} ]
[ ,CODE =  {BCD   } ]
[          {CORRS } ]
```

(CORRS is underlined.)

2741 SWITCHED LINE GROUPS ONLY

```
[        ( {EBCDIC} [   {EBCDIC} ] ) ]
[ ,CODE =( {BCD   } [ , {BCD   } ] ) ]
[        ( {CORRS } [   {CORRS } ] ) ]
```

(The first CORRS is underlined.)

2770 LINE GROUPS ONLY

```
         {EBCDIC }
 ,CODE = {USASCII}
```

7770 LINE GROUPS ONLY

```
 ,OUTTBL = module
```

3270 REMOTE LINE GROUPS ONLY

```
         {EBCDIC }
 ,CODE = {USASCII}
```

- Operand Field:

DDNAME=

specifies a one- to eight-character name which associates the generated DCB for this line group within the IMS/360 nucleus control block to the generated JCL DD cards identifying the lines within the line group. If the line group being defined is a SPOOL line group, a list of up to 20 names may be specified. Each SPOOL ddname is associated, in the order in which it was specified, with the sequentially generated SPOOL data set names produced for the SPOOL print procedures which are generated and named sequentially for each SPOOL line group. Refer to the end of the LINEGRP macro operand field discussions for an example of SPOOL ddname association.

The name(s) must begin with an alphabetic character. The operand is required. The following DDNAMES cannot be used as LINEGRP ddnames: IMS, IMSCSP, IMSDILIB, IMSLOG, IMSLOGR, JOBLIB, SHMSG, IMSACB, LGMSG, QBLKS, STEPLIB, SYSABEND, and SYSUDUMP.

FEAT=

determines whether this group of telecommunication lines is leased (NONSWITCHED) or not (SWITCHED).

when

$$\text{FEAT = SWITCHED, UNITYPE = } \left\{\begin{array}{l}\text{1050}\\\text{2740}\\\text{2741}\\\text{7770}\\\text{TWX}\end{array}\right\} \text{ are the only allowable values}$$

for the UNITYPE = keyword.

when

$$\text{FEAT = (NONSWITCHED,} \left\{\begin{array}{l}\text{AUTOPOLL}\\\underline{\text{POLL}}\end{array}\right\}\text{),UNITYPE = } \left\{\begin{array}{l}\text{1030}\\\text{1050}\\\text{2740}\\\text{2780}\end{array}\right\}$$

__Note__: Polling features are ignored for READERs, PRINTERs, PUNCHs, TAPEs, DISKs, SPOOLs, 2770s, 2980s, and 3270s. The AUTOPOLL polling feature is invalid for 2260s, 3270s (LOCAL), 2265s, 2741s, and 2740 (NOSTACTL). 2260 remote lines use conventional polling (POLL). No polling list is generated for 2260s (LOCAL), 3270s (LOCAL), 2740s (NOSTACTL), 2741s (NONSWITCHED), READERs, PRINTERs, PUNCHs, TAPEs, DISKs, or SPOOLs. A BTAM AUTOWLST is generated for 2770s, 2980s, 3270s, and multi-dropped 2780s. No polling list is generated for single-drop 2780s unless specifically requested by specifying AUTOPOLL.

Direct SYSOUT specifications (PRINTER, PUNCH, TAPE, or DISK) do not require that specific device type to be assigned at IMS execution time. All specifications except PRINTER affect only default BUFSIZE or generated JCL. A specification of PRINTER will result in execution time translation to the 48 character set, lowercase to uppercase and all other codes translated to periods(.). If a specification other than PRINTER is used, no translation occurs. If a line not generated as a PRINTER is allocated to a printer having the universal character set feature (UCS) and fold mode operation is used, unprintable characters will print as extraneous alphamerics.

UNITYPE=

specifies the terminal device type contained in this line group. UNITYPE=2260 must be specified when defining a line group containing both 2260 remote and 2265 terminals. If UNITYPE=(3270,LOCAL) is specified, then only one line statement may exist in the linegroup.

CODE=

specifies translate options for 2770 line groups, 2780 line groups, 2741 line groups, and 3270 remote line groups only.

For 2770 and 3270 line groups, specifies translate table EBCDIC/EBCDIC or RASA/SASA. The operand is required.

For 2780 line groups, specifies translate table EBCDIC/EBCDIC, RASA/SASA, or RC80/SD80. The operand is required.

For nonswitched 2741 line groups, specifies Extended Binary Decimal (EBCDIC), Binary Coded Decimal (BCD), or Correspondence Code (CORRS). Default value is CORRS.

For switched 2741 line groups, specifies Correspondence Code (CORRS) and/or Binary Coded Decimal (BCD). The combination of Extended Binary Decimal (EBCDIC) and Binary Coded Decimal (BCD) is invalid. Default value is CORRS.

MODE=

The first parameter specifies that communications are to be through the 270x Data Adapter Unit's Dual Communication Interface A or B.

The second parameter specifies the use of the transmission code designated by Code (A or B) for the 270x Data Adapter Unit Dual Code Feature. This feature must be present on the 270x. See the BTAM Manual, GC30-2004, BTAM DCBs.

EDIT=

A user-supplied physical terminal output edit routine for the terminal type in this line group. In addition to the edit routine specified here, if the LINEGRP refers to 1030 or 7770-3 terminals, an edit routine(s) must exist in IMS2.RESLIB before system definition Stage 2 is executed. See Chapter 5 of this manual.

OUTTBL=

is specified only for 7770-3 line groups. The operand represents the load module name which contains the output translate table for this line group. The control section (CSECT) name is assumed to be the same as the load module name. See 7770-3 Output Edit Routine in Chapter 5 of this manual for special considerations for this module.

An example of the LINEGRP macro for a 2260 remote line group with an output edit routine is:

```
   LINEGRP      DDNAME=DD2260,UNITYPE=2260,EDIT=OUTEDIT1
```

An example of the LINEGRP macro for a 2780 line group with no output edit routine and EBCDIC translation is:

```
   LINEGRP      DDNAME=DD2780,UNITYPE=2780,CODE=EBCDIC
```

An example of the LINEGRP macro for a switched 7770 line group with an output edit routine is:

```
   LINEGRP      DDNAME=ARU7770,UNITYPE=7770,
                EDIT=EDIT7770,FEAT=(SWITCHED),
                OUTTBL=OUT7770
```

An example of the LINEGRP macro and the resulting ddname/dsname and print procedure associations for SPOOL line groups is:

```
   LINEGRP      DDNAME=(NAME1,NAME2),UNITYPE=SPOOL

   LINEGRP      DDNAME=(NAMEA,NAMEB),UNITYPE=SPOOL
```

the SPOOL print procedure for line group one would be named IMSWT000. The data set names produced for this print procedure would be IMS2.SYSO1 and IMS2.SYSO2.

the SPOOL print procedure for line group two would be named IMSWT001. The data set names produced for this print procedure would be IMS2.SYSO3 and IMS2.SYSO4.

the corresponding JCL generated for the IMS online execution procedure would be as follows:

```
   //NAME1      DD      DISP=SHR,DSNAME=IMS2.SYSO1
   //NAME2      DD      DISP=SHR,DSNAME=IMS2.SYSO2
   //NAMEA      DD      DISP=SHR,DSNAME=IMS2.SYSO3
   //NAMEB      DD      DISP=SHR,DSNAME=IMS2.SYSO4
```

LINE Macro

The LINE macro-instruction describes both switched and nonswitched communication lines. If the line described has only one terminal attached, only one TERMINAL macro-instruction appears after the LINE macro-instruction. Multiple TERMINAL macro-instructions would appear if the description were for a multidrop line. Multiple NAME macro-instructions, each of which describes a logical terminal, may appear after each TERMINAL macro-instruction which follows a LINE macro-instruction. Each LINE macro-instruction must be followed by at least one TERMINAL macro-instruction. If the terminal is the 2260-local mode, LINE macro-instruction is required with no operands. Only one LINE statement per line group is allowed if UNITYPE=(3270,LOCAL) is specified on the LINEGRP macro statement.

		ALL LINES EXCEPT SPOOL 2260 LOCAL, AND 3270 LOCAL
	LINE	ADDR = 3 hex digits
		2740 LINES ONLY
		$\left[\text{,MODEL=}\left\{\begin{matrix}\underline{1}\\2\end{matrix}\right\}\text{,}\quad\left\{\begin{matrix}\underline{120}\\248\\440\end{matrix}\right\}\text{,control}\right]$
		1030, 7770-3, 3270 (LOCAL), DISK, TAPE, and SPOOL LINES ONLY
		[,BUFSIZE=]
		7770-3 LINES ONLY
		$\left[\text{,FEAT=}\left(\left[\left\{\begin{matrix}\underline{\text{ABC}}\\\text{ABB'}\end{matrix}\right\}\right]\left[\text{,}\left\{\begin{matrix}\underline{\text{NOREPEAT}}\\\text{REPEAT}\end{matrix}\right\}\right]\right)\right]$
		2770 LINES ONLY
		$\left[\text{,FEAT=}\left\{\begin{matrix}\text{BUFEXP}\\\text{BUFEXP1}\end{matrix}\right\}\right]$

		2980 LINES ONLY
		[,FEAT= BUFEXP]

• Operand Field:

ADDR=

specifies the address of the communication line as defined in the 270X Communication Transmission Control Unit. The address value is three hexadecimal digits in the range from 000 through 6FF. All line addresses specified in the system definition must be unique values. This operand is used only to generate IMS DD cards for cataloged procedures. It is not allowed if terminal is a 2260 local mode or 3270 local. This operand is not required for SPOOL line groups.

MODEL=

is required for 2740 LINEGRP and invalid for all other LINEGRP types. This operand specifies whether the LINE has 2740 Model 1 or Model 2 terminals attached. MODEL=2 is invalid if line group FEAT=SWITCHED. If MODEL=1 (2740 Model 1), all other parameters of the MODEL= operand must be omitted.

The second parameter specifies the maximum input buffer size for all 2740 Model 2 terminals on this line.

The third parameter specifies the header control feature for 2740 Model 2 terminals on this line and number of characters reserved for header control. The default value is 0. Entered value for header control must not exceed 28, and must be a multiple of 4.

BUFSIZE=

this operand is valid for 1030, 3270 (LOCAL), 7770, DISK, TAPE, and SPOOL lines only.

For 1030 lines, this operand is used to specify the maximum size of an input message on this line. The operand is required. The minimum acceptable value is 2; the maximum value is 182.

For 3270 local lines, this operand is used to specify the maximum size of an input message on this line. The operand is required for 3270 local lines containing 3277 terminals and is not allowed for lines containing 3284 or 3286 printers. The minimum acceptable value is 300; the maximum value is 3850.

For DISK, TAPE, or SPOOL lines, this operand is used to specify the maximum output buffer size to be used for the line. The operand is required. The minimum acceptable value is 16; the maximum value is 32767.

For 7770 lines, this operand is used to specify the input and output buffer sizes to be used for the line. The first operand is the input buffer size; the second operand is the output buffer size. Default value is (50,50). Maximum value for either operand is 246; minimum value is 1.

FEAT=

For 7770-3 lines only: The first parameter is used to specify the line code feature on the 7770. The second parameter is used to specify the message repeat capabilities of this line.

Note: The Touch-Tone Telephone* operates on either ABC or ABB' lines.

See Chapter 5 of this manual for additional information on repeat operations, under the topic 7770-3 Input Edit Routine.

For 2770 lines only: Specifies the largest terminal input buffer (2772 line buffer) size which will reside on this line. The omission of this keyword specifies an input buffer size of 128 bytes; the specification of the Buffer Expansion (BUFEXP) or additional Buffer Expansion (BUFEXP1) terminal feature will allow for an input buffer size of 256 bytes or 512 bytes, respectively.

For 2980 lines only: Specifies the largest terminal input buffer size which will reside on this line. The omission of this keyword specifies an input buffer size of 48 bytes; the specification of an expanded buffer (BUFEXP) will allow for an input buffer size of 96 bytes.

An example of the LINE and LINEGRP macros used in combination to define a 2740 Model 2 with maximum input buffer of 248 bytes, line address of 024, and no header control or edit routine is:

```
LINEGRP     DDNAME=D27402,UNITYPE=2740
LINE        ADDR=024,MODEL=(2,248)
```

An example for a 1030 environment with a maximum of 50 characters for input messages is:

```
LINEGRP     DDNAME=DD1030,UNITYPE=1030
LINE        ADDR=021,BUFSIZE=50
```

Although no edit routine is specified on the LINEGRP statement, the 1030 edit routine named DFS10300 must exist in the USERLIB library. See Chapter 5 of this manual.

* Registered Trademark of the American Telephone & Telegraph Co.

CTLUNIT Macro

The CTLUNIT macro-instruction specifies 2848, 2972, and 3271 control unit characteristics. For 2848 control units, CTLUNIT specifies control unit address, write line addressing (WLA) feature, and 2848 model number. For 2972 control units, CTLUNIT specifies control unit address and buffer size. For 3271 control units, CTLUNIT specifies control unit address and model number. CTLUNIT is valid only for 2260, 2980, and 3270 remote line groups. For 2260 line groups containing both 2260 remote and 2265 terminals, the CTLUNIT statement must precede any definition of any 2260 terminal, and 2265 terminal statements must be defined before any CTLUNIT statements. All TERMINAL macro statements defining 3275 display stations which are to be included in a 3270 line group with 3271/3277 control unit/display station clusters must be defined prior to any CTLUNIT macro statement which defines a 3271 control unit.

```
|      | CTLUNIT |  [ADDR=          hex byte]
|      |         |
|      |         | [ ,WLA=     ( { YES }, [1053] ) ]
|      |         | [           ( { NO  }         ) ]
|      |         |
|      |         | [ ,MODEL=     { 1 } ]
|      |         | [             { 2 } ]
|      |         | [             { 3 } ]
|      |         |
|      |         | [ ,BUFSIZE=   { 48 } ]
|      |         | [             { 96 } ]
```

(2260 line groups must specify 2848 control unit features.)

- Operand Field:

ADDR=

specifies the two-digit hexadecimal address of either (1) the 2848 control unit associated with the 2260 remote mode or (2) the two-digit hexadecimal polling address of the 3271 associated with the remote 3277, or (3) the two-digit hexadecimal address of the 2972 control unit. These addresses are those assigned by the customer engineer upon installation of the control unit. Note that for the 3271, the customer engineer will assign the selection address, which must be converted to the polling address for specification here. This operand is not allowed if 2260 local mode is being specified. To be acceptable as a valid 2972 control unit address, this operand must be the hexadecimal representation of a character from A through Z.

WLA=

specifies the existence of the write line addressing and 1053 features on this 2848 control unit. The 1053 parameter can not be specified for 2260 local mode, but write line addressing can be specified. Specification of this keyword is invalid for 2980 line groups.

MODEL=

specifies the 2848 or 3271 control unit model number. For 3271 control units only, the value 1 or 2 may be specified. Default value is 2. For 2848 control units, the entered value will determine the input buffer size for terminals attached to this control unit. Specification of this keyword is invalid for 2980 line groups.

2848 Control Units (default value is 3):

Model	Buffer Size
1	240
2	480
3	960

BUFSIZE=

specifies the 2972 control unit buffer size. The entered value will determine the output buffer size for terminals attached to this control unit and may not exceed the value implied with the FEAT= keyword of the LINE macro instruction. This keyword is invalid for 2260 and 3270 line groups.

TERMINAL Macro

The TERMINAL macro-instruction describes a physical terminal on a nonswitched line or the representation to BTAM of a physical terminal on a switched line. NAME macro-instructions which follow the TERMINAL macro-instruction supply the logical terminal name(s) associated with the physical terminal at system definition time. Within the definitions and restrictions of terminal security, the first NAME macro-instruction encountered following a TERMINAL macro-instruction becomes the response or input/output logical terminal. Each TERMINAL macro-instruction defining a terminal which connects to a switched communication line must not be followed by a NAME macro-instruction. If the terminal is a 2741, PRINTER, PUNCH, TAPE, DISK, or 2740 Model 1 Non-Station-Control, the TERMINAL macro-instruction may be specified with no operands.

[label]	TERMINAL	ALL TERMINAL TYPES
		[ADDR= hex byte(s) or address list] [,EDIT= {YES | NO}] [,LTERM=name]
		2260 AND 3270 REMOTE OR LOCAL LINE GROUPS
		[,PAGDEL= {YES | NO}]
		2260 REMOTE LINE GROUPS
		[,UNIT= {1053 | 2845 | 2260}]
		2265 LINE GROUPS and TERMINALS
		[,FEAT= ({1053 | 2265},{2 | 1}[, WLA])] [,PAGDEL={YES | NO}]
		3270 LOCAL LINE GROUPS

$$\left[\text{,UNIT=}\quad\left\{\begin{matrix}3284\\3286\\\underline{3277}\end{matrix}\right\}\right]$$

$$\left[\text{,FEAT=}\left(\left\{\begin{matrix}\underline{1}\\2\end{matrix}\right\}\text{,[NOCOPY],}\left[\left\{\begin{matrix}\text{DEKYBD}\\\underline{\text{PFK}}\\\text{NOPFK}\end{matrix}\right\}\right]\text{,}\left[\left\{\begin{matrix}\underline{\text{CARD}}\\\text{NOCD}\end{matrix}\right\}\right]\text{,}\left[\left\{\begin{matrix}\underline{\text{PEN}}\\\text{NOPEN}\end{matrix}\right\}\right]\right)\right]$$

$$\left[\text{,BUFSIZE=}\left\{\begin{matrix}132\\126\\\underline{120}\end{matrix}\right\}\right]$$

3270 REMOTE LINE GROUPS

$$\left[\text{,UNIT=}\quad\left\{\begin{matrix}3284\\3286\\3275\\\underline{3277}\end{matrix}\right\}\right]$$

$$\left[\text{,FEAT=}\left(\left\{\begin{matrix}\underline{1}\\2\end{matrix}\right\}\text{,}\left[\left\{\begin{matrix}\text{NOCOPY}\\\underline{\text{COPY}}\end{matrix}\right\}\right]\text{,}\left[\left\{\begin{matrix}\text{DEKYBD}\\\underline{\text{PFK}}\\\text{NOPFK}\end{matrix}\right\}\right]\text{,}\left[\left\{\begin{matrix}\underline{\text{CARD}}\\\text{NOCD}\end{matrix}\right\}\right]\text{,}\left[\left\{\begin{matrix}\underline{\text{PEN}}\\\text{NOPEN}\end{matrix}\right\}\right]\right)\right]$$

$$\left[\text{,BUFSIZE=}\left\{\begin{matrix}132\\126\\\underline{120}\end{matrix}\right\}\right]\qquad\text{[,COMPT= PTR1]}$$

1030 LINE GROUPS

$$\left[\text{,UNIT=}\quad\left\{\begin{matrix}1033\\\underline{1031}\end{matrix}\right\}\right]$$

2740 MODEL 2 LINES

$$\left[\text{,BUFSIZE=}\quad\left\{\begin{array}{c}\underline{\text{120}}\\ \text{248}\\ \text{440}\end{array}\right\}\right]$$

[,HEADCTL=]

1050 LINE GROUPS

$$\left[\text{,FEAT=}\left(\left\{\begin{array}{l}\text{INPUT}\\ \text{1053}\\ \underline{\text{1052}}\end{array}\right\},\left\{\begin{array}{l}\text{CARD}\\ \underline{\text{TAPE}}\end{array}\right\},\left\{\begin{array}{l}\text{TAPE}\\ \underline{\text{CARD}}\end{array}\right\}\right)\right]$$

$$\left[\text{,COMPT=}\left(\left\{\begin{array}{c}\text{PTR1}\\ \text{1}\end{array}\right\},\left\{\begin{array}{c}\text{PTR2}\\ \text{2}\end{array}\right\},\left\{\begin{array}{c}\text{PCH1}\\ \text{3}\end{array}\right\},\left\{\begin{array}{c}\text{PCH2}\\ \text{4}\end{array}\right\}\right)\right]$$

2780 LINE GROUPS

$$\left[\text{,FEAT=}\left(\left\{\begin{array}{l}\text{MULTIPLE}\\ \underline{\text{STANDARD}}\end{array}\right\}\begin{array}{l}\text{or}\\ \text{or}\end{array}\left\{\begin{array}{l}\text{M}\\ \underline{\text{S}}\end{array}\right\},\left\{\begin{array}{r}\text{144}\\ \text{120}\\ \underline{\text{80}}\end{array}\right\},\left\{\begin{array}{c}\underline{\text{1}}\\ \text{2}\\ \text{3}\\ \text{4}\end{array}\right\}\right)\right]$$

$$\left[\text{,COMPT=}\left(\left\{\begin{array}{l}\text{PTR1}\\ \text{1}\end{array}\right\},\left\{\begin{array}{l}\text{PCH1}\\ \text{2}\end{array}\right\}\right)\right]$$

2741 LINE GROUPS

$$\left[\text{,FEAT=}\left\{\begin{array}{l}\text{INTERUPT}\\ \underline{\text{NOINTRPT}}\end{array}\right\}\right]$$

TWX LINE GROUPS

[,FEAT = AUTOSIGN]

2770 LINE GROUPS

```
[,FEAT= ({1} [, {BUFEXP }])]
        ({2} [  {BUFEXP1}])

                 [ {CARD  } ] [ {CARD  } ] [ {CARD  } ]
                 [ {READER} ] [ {READER} ] [ {READER} ]
                 [ {VIDEO } ] [ {VIDEO } ] [ {VIDEO } ]
         (PTR    [ {PTR   } ] [ {PTR   } ] [ {PTR   } ])
, COMPT= (KYBD  ,[,{PT    } ] [,{PT    } ] [,{PT    } ])
                 [ {PTRDR } ] [ {PTRDR } ] [ {PTRDR } ]
                 [ {MDI   } ] [ {MDI   } ] [ {MDI   } ]
                 [ {MICR  } ] [ {MICR  } ] [ {MICR  } ]
```

if PT or PTRDR component specified, then;

[,PTSEG=]

if MDI component specified, then;

```
[         {IGNORE}          {NOCHK }]
[,ERROPT= {ACCEPT}  ,VERCHK={VOKCHK}]
[         {REJECT}                  ]

[          {19            }]
[,REPLACE= {two hex digits}]
```

[,MDISEG=]

if VIDEO component specified then;

```
[,PAGDEL = {YES}]
[          {NO }]
```

Defaults (underlined): REJECT, VOKCHK, 19, YES.

		2980 LINE GROUPS
		[,FEAT=([{1 / 2 / 4}] [,{0 / decimal number}])]
		SPOOL LINE GROUPS
		,FEAT= {AUTOSCH / NOAUTOSCH}

- Label Field:

  If any subsequent NAME statement references this TERMINAL statement in its OUTPUT= operand, a label must be specified for the TERMINAL statement.

- Operand Field:

ADDR=

specifies the physical terminal addressing character in terminal code, hexadecimal representation, for all terminals. For example, physical terminal address 'A' for a 2740 would be coded ADDR=E2.

If the terminal is 2260 or 3270 local mode, the terminal address must be specified as three hexadecimal digits.

If the terminal is a 2265 or a 3275, the address must be specified as four hexadecimal digits. For a 2265 terminal, the first two digits specify the 2845 address, while the last two specify the terminal address (example, ADDR=4050). For a 3275 terminal, the first two digits specify the control unit polling address, while the last two specify the terminal address (example, ADDR=4040). Note that for the 3275, the Customer Engineer will assign the selection address, which must be converted to a polling address for specification here.

If the terminal is a 3277 or 3284/3286 attached to a 3271, the address must be specified as two hexadecimal digits, specifying the terminal address related to the particular display station or printer. For instance, the first display station or printer would have an address of ADDR=40, the second ADDR=C1, etc.

If and only if the terminal being defined is a 1031 or 1035, an address list of up to 24 unique addresses is valid. This address list specification allows the user to define multiple 1031 or 1035 terminals on a single communication line with one TERMINAL macro statement. When the address list specification is used, the LTERM operand on this macro is required. Independent of terminal type, all input terminal addresses for a line must be unique. This operand is not required for 2741, READER, PRINTER, PUNCH, TAPE, DISK, SPOOL, or 2740 Model 1 Non-Station-Control terminals.

EDIT=

means that the user-supplied message output edit routine for this terminal type specified in the associated LINEGRP is to be used for this TERMINAL. Default value is NO.

UNIT=

is valid for 2260, 3270, and 1030 line groups only. Defines this terminal as an output-only device for 2260 and 1030 line groups (that is, 1053 or 1033, respectively). If the terminal is 2260 local mode, this operand should be omitted, since 2260 is the default and 1053 is invalid for 2260 local mode. This operand is also used to define the 2845 control unit for 2265 terminals which are part of a 2260 line group. For 3270 line groups, this operand is used to define this terminal as a 3284 or 3286 printer or 3277 display unit. 3270 display units may not be defined in the same local line group as 3284 or 3286 printers. This operand is also used to define a 3275 terminal which is part of a 3270 remote line group.

PAGDEL=

is valid for 2260, 2265, 3270, and 2770 with 2265 component line groups only. Specifies the desire for the Automatic Page Deletion feature of the Paging special feature. Default value is YES.

The PAGDEL=YES option should be used for terminals used in an interactive or conversational manner to provide consistent and understandable terminal action for the terminal operator. If input from the terminal is normally followed by a response to the terminal, the YES option should be used.

The PAGDEL=NO option is provided for terminals that will normally produce many input messages for which no response will be made. When NO is specified, input from a paged message will normally be responded to by IMS with the first page of that same message - any application program response will be queued. This type of operation can be useful for specific kinds of data entry applications, but will result in terminal operator confusion for most applications.

LTERM=

may specify a logical terminal name for this TERMINAL statement. This name is used to check input terminal security for this terminal. The referenced NAME macro must precede this TERMINAL statement. When this operand is present, the TERMINAL statement can not be followed by a NAME statement. This operand is required for READER terminals. This operand would be used in the following manner: Assume that a previously defined TERMINAL statement defined a 1033 output device and was followed by the NAME ABLE. In addition, assume that this TERMINAL statement refers to one or more 1031s on the same communication line which are input-only devices. All input security for the 1031s would be checked against the security characteristics assigned to ABLE.

Also, when a list of LTERMs is defined under the previous TERMINAL statement to which reference is made, only the first LTERM name may be selected from that list. If an attempt is made to select other than the first, IMS/360 will ignore the name chosen and assign the value of the first name listed.

Example:

```
TERMINAL    ADDR=E2
  NAME      LTERM1
  NAME      LTERM2
  NAME      LTERM3

TERMINAL    ADDR=E4,LTERM=LTERM2
```

In this example, IMS/360 treats the second TERMINAL statement as though it read LTERM=LTERM1.

FEAT=

- For 1050 Line Groups:

with parameters 1053, 1052, CARD, and TAPE, defines the physical characteristics of terminals:

The first subparameter (1053) or (1052) specifies that the component attached to address 01 is either a 1053 or a 1052. Default value is 1052.

The second subparameter specifies that the component attached to address 03 is either a card or a tape punch. Default value is TAPE.

The third subparameter specifies that the component attached to address 04 is either a card or a tape punch. Default value is CARD.

If FEAT=INPUT is specified, the terminal is treated as an input-only terminal and cannot receive output.

- For 2780 Line Groups:

with the indicated parameters specifies 2780 terminal features:

The first subparameter (MULTIPLE or M) or (STANDARD or S) specifies the MULTIPLE or STANDARD record transmission feature. Default value is STANDARD.

The second subparameter specifies the size of the print line. Default value is 80.

The third subparameter specifies the terminal model number. Default value is 1.

- For 2741 Line Groups:

with the parameter INTERUPT or NOINTRPT, specifies whether or not this 2741 terminal is equipped with the Receive Interrupt feature.

- For TWX (33/35 Teletypewriter) Line Groups:

with the AUTOSIGN parameter, specifies that this TWX terminal is equipped with the Automatic Sign-On feature.

- For 2265 Line Groups and Terminals:

specifies the 2845 control unit features.

The first subparameter specifies that this terminal is either the 2265 or the 1053 component.

The second subparameter specifies the screen configuration, as follows:

1 - 15 lines of 64 characters
2 - 12 lines of 80 characters

The third subparameter specifies the write line addressing (WLA) feature for the 2265.

- For 2770 Line Groups:

specifies the VIDEO component screen size and the terminal output buffer size:

The first subparameter specifies the screen configuration for the VIDEO component if a VIDEO component is defined. If this subparameter is specified and a VIDEO component is not defined, the subparameter is ignored. Specifications and applicable screen sizes follow (default value is 1):

1 - 15 lines of 64 characters
2 - 12 lines of 80 characters

The second subparameter specifies the terminal output buffer size. If omitted, an output buffer size of 128 bytes is implied. The specification of the terminal Buffer Expansion Feature (BUFEXP) or additional Buffer Expansion Feature (BUFEXP1) specifies a terminal output buffer size of 256 or 512 bytes, respectively. The output buffer size specified herein may not exceed the largest terminal input buffer size specified with the FEAT= keyword of the LINE macro instruction for the line within which this terminal is to reside.

- For 2980 Line Groups:

The first subparameter defines the 2980 model number (1, 2, or 4). If omitted, Model 1 is default. The second subparameter defines the number of tabs between the first carriage stop and the intermediate carriage stop on 2980 Models 1 and 4. If no value is specified, zero (0) is default.

- For SPOOL Line Groups:

with the parameter AUTOSCH or NOAUTOSCH, specifies whether or not the Spool Print Program is to be automatically scheduled to print a full or terminated SPOOL data set.

- For 3270 Line Groups:

The first subparameter (1 or 2) specifies the terminal model number. Default value is 1. The terminal model number specified for 3284, 3286, and 3277 terminals must be less than or equal to the 3271 control unit model number specified. The terminal model number specified for a 3277 terminal which specifies the

COPY feature must be less than or equal to the terminal model number specified for any candidate printers. For a detailed discussion of the COPY feature and candidate printers for 3270 remote line groups, refer to the IMS/360 System/Application Design Guide (SH20-0910).

The second subparameter specifies whether or not the COPY function is available for this 3275 or 3277 terminal. The specification of the COPY feature for a 3284, 3286, or 3270 local terminal is invalid. Default value is COPY.

The third subparameter specifies whether a Data Entry Keyboard (DEKYBD) or Program Function Keys (PFK) are available on this 3275 or 3277 terminal. PFK and DEKYBD are invalid for 3284 and 3286 terminals. Default value is PFK.

The fourth subparameter specifies whether the Operator Identification Card Reader (CARD) is available on this 3275 or 3277 terminal. CARD specification is invalid for 3284 and 3286 terminals. Default value is CARD.

The fifth subparameter specifies whether the Selector Pen is available on this 3275 or 3277 terminal. PEN specification is invalid for 3284 and 3286 terminals. Default value is PEN.

COMPT=

specifies output components. This operand is only valid for 1050, 2770, 2780, and 3275 terminals.

- For 1050 terminals:

Valid entries are PTR1, PTR2, PCH1, and PCH2 or 1, 2, 3, and 4, respectively. PTR1 or 1 is associated with physical terminal address 01. PTR2 or 2 is associated with physical terminal address 02. PCH1 or 3 is associated with physical terminal address 03. PCH2 or 4 is associated with physical terminal address 04. These entries may appear in any sequence. If this keyword specification is omitted, it is assumed that the user has a full complement of output components for the device.

- For 2780 terminals:

Valid entries are PTR1 and PCH1 or 1 and 2 respectively. These entries may appear in any sequence. If this keyword specification is omitted, it is assumed that the user has a full complement of output components for the device.

- For 3275 terminals:

The only valid entry is PTR1. PTR1 specifies that a 3284 Model 3 printer is attached to this 3275 terminal. The video display will be considered COMPT1 and the 3284 Model 3 printer will be considered COMPT2.

- For 2770 terminals:

specifies both input and output components to be associated with this 2770 Data Communication System. The COMPT= keyword and operand are required. The four positions within the operand sublist relate directly to the 2772 Multi-Purpose Control Unit's four spaces available for attaching units of different types of input and output media. Position four of the operand sublist should not be used unless the 2772 Multi-Purpose Control Unit is equipped with the Expanded I/O Capability special feature.

KYBD may be specified only in operand sublist position one. PTR may be specified in operand sublist position one and/or once in one, and only one, of the remaining operand sublist positions. The remaining component specifications must be specified in operand sublist positio two, three, or four and may only be specified in one of these positions.

The valid sublist entry specifications and their definitions are as follows:

KYBD - As the keyboard is basic in any 2770 Data Communication System, this entry is used merely to define the absence of a printer.

PTR - PTR defines the presence of a serial printer (1053 or 2213) or a line printer (2203).

CARD - CARD defines the presence of an output device for handling cards (545 Output Punch) or the presence of both an input and an output device for handling cards (2502 Card Reader and 545 Output Punch).

READER -READER defines the presence of an input-only device for handling cards (2502 Card Reader).

VIDEO - VIDEO defines the presence of a 2265 Display Station.

PT - PT defines the presence of an output device for handling tape (1018 Paper Tape Punch) or the presence of both an input and an output device for handling paper tape (1017 Paper Tape Reader and 1018 Paper Tape Punch).

PTRDR - PTRDR defines the presence of an input-only device for handling paper tape (1017 Paper Tape Reader).

MDI - MDI defines the presence of the IBM 50 Magnetic Data Inscriber.

MICR - MICR defines the presence of the IBM 1255 Magnetic Character Reader.

Note: If a printer (2203) is attached in other than the first component slot, no input device will be supported in the same slot.

BUFSIZE=

valid only for 2740 Model 2 and 3270 lines.

For 2740 Model 2 lines, specifies the size of the output buffer for this 2740 Model 2 terminal. Value cannot exceed the value specified for the preceding LINE statement.

For 3270 remote lines, specifies the number of print positions per line for a 3284 or 3286 Model 1 or 2 terminal.

For 3270 local lines, specifies the number of print positions per line for the 3284 or 3286 terminal being defined.

HEADCTL=

specifies header control feature for this 2740 Model 2 terminal; valid only for 2740 Model 2 lines and specifies the number of characters reserved for header control. If either BUFSIZE or HEADCTL operands are omitted, the default values are the values specified for the preceding LINE statement. Entered value for header control cannot exceed 28 and must be a multiple of 4.

PTSEG=

is valid only for 2770 terminals which have defined in the COMPT keyword the presence of paper tape media. Specifies the largest input paper tape segment to be presented to IMS. This value may not exceed the size of a multisegment message buffer minus 40 bytes. The size of the multisegment message buffer is specified through the use of the BUFFERS keyword on the MSGQUEUE macro statement. Default value is 80 bytes.

MDISEG=

is valid only for 2770 terminals which have defined in the COMPT keyword the presence of an IBM 50 Magnetic Data Inscriber (MDI). Specifies the largest input MDI segment to be presented to IMS. This value may not exceed the size of a multisegment message buffer minus 40 bytes. The size of the multisegment message buffers is specified through the use of the BUFFERS keyword on the MSGQUEUE macro-statement. Default value is 80 bytes.

ERROPT=

is valid only for 2770 terminals which have defined in the COMPT keyword the presence of an IBM 50 Magnetic Data Inscriber. Specifies the handling of erroneous records as follows:

IGNORE - error conditions are to be disregarded, and the record is not to be passed to the user program.

ACCEPT - error conditions are to be disregarded, but the record is to be passed to the user program.

REJECT - the entire message is to be rejected. Default value is REJECT.

VERCHK=

is valid only for 2770 terminals which have defined in the COMPT keyword the presence of an IBM 50 Magnetic Data Inscriber. Specifies whether (VOKCHK) or not (NOCHK) the records are to be checked for verify-OK codes. If VOKCHK is specified and a record does not contain the verify-OK code, the record is considered to be in error.

REPLACE=

is valid only for 2770 terminals which have defined the COMPT=MDI operand. This operand specifies the code to be used as a replacement character whenever a 2772 replacement character is detected in the input. It must be specified as two hexadecimal characters. Default value is 19.

<u>WARNING</u>: If the REPLACE character is a value less than X'40', the character will be edited out by the system input edit routine before the application program receives the message.

NAME Macro

The NAME specification defines a logical terminal name (LTERM) associated with a physical terminal. The presence of the keyword MASTER in the LTERM operand designates this logical terminal name as the master terminal. The COMPT operand associates the specified LTERM with a specific component in a 1050, 2770, 2780, or 3275 terminal complex. The COMPT operand is valid only for 1050, 2770, 2780, 2980, and 3270 line groups.

```
|            |           |                 {     lterm      }
|            |   NAME    |                 {(lterm,MASTER), }
|            |           | [
|            |           | [,COMPT=    {1}   component name options   ]
|            |           | [           {2}   for 1050 and 2770 line   ]
|            |           | [           {3}   groups                   ]
|            |           | [           {4}                            ]
|            |           | [                                          ]
|            |           | [           |1|   component name options   ]
|            |           | [           |2|   for 2780 line groups     ]
|            |           | [                                          ]
|            |           | [         PTR1    3275 terminals only      ]
|            |           | [                                          ]
|            |           | [         COMMON  2980 line groups only    ]
|            |           |
|            |           | [,OUTPUT= terminal label]
|            |           | [,EDIT=     ({NO }  ,  {ULC})]
|            |           | [           ({YES}     {UC })]
```

- Operand Field:

lterm

defines a name for a logical terminal associated with a previously defined physical terminal (TERMINAL). This name must begin with either a letter or a digit. WTOR is an invalid lterm name. The entered name cannot begin with the character sequence INQU. The operand is required. If the operand is entered as a sublist with the keyword MASTER as a second parameter, the entered name becomes the IMS/360 master terminal.

One lterm must be specified as the master terminal. The lterm chosen as the master terminal cannot be on a switched LINEGRP and must be either a 1050 or a 2740 with Station Control physical terminal.

Note: For terminals with multiple NAME lterms assigned, system error message responses will be presented to the lterm name with the lowest collating sequence, unless the terminal is an input-only terminal. For input only, the system responses will go to the first assigned lterm OUTPUT= terminal.

COMPT=

defines the output component associated with this lterm. If this lterm is the master terminal, the COMPT= operand value must be either not specified or 1.

For 1050 terminal components, the user can specify 1 or PTR1, 2 or PTR2, 3 or PCH1, or 4 or PCH2.

For 2770 terminal media, the user can specify 1, 2, 3, or 4. These values relate to the four spaces available on the 2772 Multi-Purpose Control Unit for attaching input and/or output media.

For 2780 terminal components, the user can specify 1 or PTR1, or 2 or PCH1.

For 3275 terminals, the user may only specify PTR1, which refers to a 3284 Model 3 printer attached to a 3275 terminal within a remote 3270 line group.

For 2980 terminals, the user can specify COMMON. The specification of COMMON assigns an output-only logical terminal to the common buffer of the 2972 control unit defined by the previous CTLUNIT statement.

OUTPUT=

defines the label field of a TERMINAL macro to be used as the output terminal for this logical terminal name. The referenced TERMINAL statement must occur before this NAME statement.

EDIT=

specifies whether or not the logical terminal user-supplied edit routine is to be used when routing a message to this logical terminal. The logical terminal edit routine is only included within the IMS/360 control program nucleus if one of the NAME statements specifies EDIT=YES. If ULC is specified, output will be transmitted as received. If UC is specified, output will be translated to uppercase before transmission.

The following examples describe LINEGRP, LINE, TERMINAL, and LTERM statements for various communication configurations.

This example describes a 1030 terminal configuration. Two communication lines are assumed. One line contains three 1031s and one 1033. The other line contains three 1031s. All input terminal security for the 1031s on the first line is to be checked against the logical terminal name OUT1033. All input terminal security for the 1031s on the second line is to be checked against the logical terminal name OUT1033A. Any output enqueued on the logical terminal name OUT1033A is to be sent to the physical 1033 terminal. Any output enqueued on the logical terminal name OUT1033 is to be sent to the physical 1033 terminal.

<u>Note:</u> All underlined addresses on all these examples are to be replaced with the user's addresses.

```
  LINEGRP       DDNAME=DD1030,UNITYPE=1030
   LINE         ADDR=0A6,BUFSIZE=100
TERMA TERMINAL  ADDR=62,UNIT=1033
    NAME        OUT1033
```

```
 TERMINAL     ADDR=(62,64,67),LTERM=OUT1033,UNIT=1031
LINE          ADDR=0A7,BUFSIZE=100
 TERMINAL     ADDR=62,UNIT=1031
 NAME         OUT1033A,OUTPUT=TERMA
 TERMINAL     ADDR=(64,67),LTERM=OUT1033A
```

This example describes a 2260 terminal configuration. One communication line is assumed with two 2848 control units attached. One 2848 is a Model 2, has no write-at-line-address feature, and has two 2260s attached.

```
LINEGRP       DDNAME=DD2260,UNITYPE=2260
 LINE         ADDR=0A0
 CTLUNIT      ADDR=40,MODEL=2
  TERMINAL    ADDR=A0
   NAME       2260A
  TERMINAL    ADDR=A1
   NAME       2260B
 CTLUNIT      ADDR=41,MODEL=3,WLA=(YES,1053)
  TERMINAL    ADDR=A2
   NAME       2260C
  TERMINAL    ADDR=A3,UNIT=1053
   NAME       1053A
```

This example describes a 2740 Model 1 and Model 2 nonswitched communication line environment. One line attaches two 2740 Model 1 terminals, one of which is the IMS/360 master terminal. The second line attaches three 2740 Model 2 terminals. The maximum buffer size on the second line is 248 bytes and the header control is 16 bytes. The first 2740 Model 2 has buffer and header control functions as specified on the LINE macro. The second 2740 Model 2 has a buffer size of 120 bytes and header control as specified on the LINE macro. The third 2740 Model 2 has no header control but buffer size as specified on LINE macro.

```
LINEGRP       DDNAME=DD2740,UNITYPE=2740
 LINE         ADDR=027
  TERMINAL    ADDR=E2
   NAME       (CTRL,MASTER)
  TERMINAL    ADDR=E4
   NAME       B2740
 LINE         ADDR=028,MODEL=(2,248,16)
  TERMINAL    ADDR=E2
   NAME       A27402
  TERMINAL    ADDR=E4,BUFSIZE=120
   NAME       B27402
  TERMINAL    ADDR=E6,HEADCTL=0
   NAME       C27402
```

This example describes a 1050 configuration. One nonswitched communication line is assumed with three 1050s attached. The first 1050 has output components of 1052, 1055, and 1057. It may also have 1054 and 1056 input components. Four logical terminal names are defined for queuing output for various components. The first logical terminal name, 1050A, queues output for the 1052 printer. The second 1050 has the full complex of components. The third 1050 has only a 1052 printer.

```
LINEGRP      DDNAME=DD1050,UNITYPE=1050
 LINE        ADDR=030
  TERMINAL   ADDR=E2,COMPT=(PTR1,PCH1,PCH2)
   NAME      1050A
   NAME      PRINTER,COMPT=PTR1
   NAME      PTPNCH,COMPT=PCH1
   NAME      CDPNCH,COMPT=PCH2
  TERMINAL   ADDR=E4
   NAME      PRINTA,COMPT=PTR1
   NAME      PRINTB,COMPT=PTR2
   NAME      PTPNCHA,COMPT=PCH1
   NAME      CDPNCHA,COMPT=PCH2
  TERMINAL   ADDR=E6,COMPT=PTR1
   NAME      1050B
```

POOL Macro

The POOL macro-instruction describes a pool of logical terminals which are to be associated with a set of switched communication lines. The IMS/360 user need have only one logical terminal pool for all autoanswer communication lines. All POOL macro-instructions must follow after all LINE macro-instructions within a switched LINEGRP. See Figure 6A, Switched Communications Terminal Set.

	POOL	

It is suggested that reference be made to Chapter 3 of the *IMS/360 System/Application Design Guide* for a description of dial telecommunication support. A discussion is provided on the use of POOL and SUBPOOL macros.

SUBPOOL Macro

The SUBPOOL macro-instruction defines a set of logical terminals within a POOL which may be associated with a given physical terminal on a switched communication line when the /IAM command is executed. One or more SUBPOOLs may be defined within a POOL macro-instruction. At least one SUBPOOL must be defined for each POOL macro-instruction.

	SUBPOOL	

The subpool statement must be followed by one or more NAME macro statements. These NAME statements may optionally contain the COMP= operand. If the COMP= operand is present on any NAME statement within a SUBPOOL, it must be defined within a 1050 line group. If the COMP= operand is not present on any NAME statement within a SUBPOOL, it may be defined within a 2740 or 1050 line group.

MAXIMUM SYSTEM DEFINITION MACRO-INSTRUCTION OCCURRENCES

The IMS/360 system produced by IMS/360 system definition has defined limitations on the number of each system resource type. The maximum number of any resource type is controlled by the number of occurrences of macro-instructions in each system definition. The initial resource limits set within IMS/360 system definition are shown by the following table:

Macro Instruction/Resource	Maximum Occurrences/Specification
IMSCTRL	1
IMSCTF	1
SPAREA	1
MSGQUEUE	1
BUFPOOLS	1
DATABASE	300
APPLCTN	350
TRANSACT	584
*Transaction Edit Routines	20
LINEGRP	50
**User Routines	20
LINE	200-A
CTLUNIT	40
TERMINAL	250-B
NAME	300
POOL	200-D
SUBPOOL	250-C
IMSGEN	1

* Transaction Edit Routines are specified through the use of the EDIT operand of the TRANSACT macro. While the initial limit of TRANSACT statements which may be specified is 584, only 20 Transaction Edit Routines are accepted.

** User Routines refers to the user-supplied physical terminal output edit routines specified through the use of the EDIT operand of the LINEGRP macro plus the required edit routines for 1030, 2980, and 7770 line groups.

where:

A

is the number of message or batch-message processing regions defined in the IMSCTRL macro-instruction plus the number of occurrences of the POOL macro-instruction.

B

is the number of occurrences of the SUBPOOL macro-instruction.

C

is the number of occurrences of the TERMINAL macro-instructions.

D

is the number of message or batch-message processing regions defined in the IMSCTRL macro-instruction plus the number of occurrences of the LINE macro-instruction.

MODIFYING SYSTEM DEFINITION RESOURCE LIMITATIONS

The maximum number of any resource type, controlled by the number of occurrences of macro-instructions in an IMS/360 system defintion, may be modified to reach one or more of the limits shown by the following table:

Macro Instruction/Resource	Maximum Occurrences/Specification
IMSCTRL	1
IMSCTF	1
SPAREA	1
MSGQUEUE	1
BUFPOOLS	1
DATABASE	1000
APPLCTN	819
TRANSACT	584
Transaction Edit Routines	255
LINEGRP	255
User Routines	100
LINE	255-A
CTLUNIT	255
TERMINAL	1000-B
NAME	744
POOL	255-D
SUBPOOL	1000-C
IMSGEN	1

where:

A

is the number of message or batch-message processing regions defined in the IMSCTRL macro-instruction plus the number of occurrences of the POOL macro-instruction.

B

is the number of occurrences of the SUBPOOL macro-instruction.

C

is the number of occurrences of the TERMINAL macro-instruction.

D

is the number of message or batch-message processing regions defined in the IMSCTRL macro-instruction plus the number of occurrences of the LINE macro-instruction.

The initial resource limits set within IMS/360 system defintion are designed to allow the user to perform an IMS/360 system definition using OS component Assembler F. Any increase in the limit of a resource without a corresponding decrease in the limit of another resource, may result in OS component Assembler F error message 'IEU054 LOCAL DICTIONARY FULL'. The following formula may be used to establish a balance of resource limits which will allow an IMS/360 system definition to be performed using OS component Assembler F:

$$(3*(A+B+E+J))+(8*(C+I))+(2*(D+F))+G+H \leq 11000$$

where:

A = number of DATABASE statements
B = number of APPLCTN statements
C = number of TRANSACT statements
D = number of Transaction Edit Routines
E = number of LINEGRP statements
F = number of User Routines
G = number of LINE and POOL statements
H = number of CTLUNIT statements
I = number of TERMINAL and SUBPOOL statements
J = number of NAME statements

The preceding formula may be disregarded and one or more or all of the resource limits may be set to their respective maximums if OS program product Assembler H is used to perform the IMS/360 Stage 1 system definition.

IMSMAX Macro

The IMSMAX macro is used to produce an IEBUPDTE job which will modify the required members of IMS2.GENLIB in order to increase and/or decrease the initial IMS/360 system definition resource limits. The format of the macro is as follows:

```
IMSMAX          DBASES=number
               ,PGMS=number
               ,TRANS=number
               ,TEDITS=number
               ,LGRPS=number
               ,LINES=number
               ,CUNITS=number
               ,TERMS=number
               ,NAMES=number
               ,RTNES=number
               ,JACCT=job accounting
               ,SACCT=step accounting
               ,JCLASS=job class
               ,MCLASS=output class
               ,PRTY=job priority
```

- Operand Field

DBASES=

specifies the maximum number of DATABASE statements that may be specified for an IMS/360 Stage 1 system definition. This operand must be specified as a decimal number from 1 to 1000. Default value is 300.

PGMS=

specifies the maximum number of APPLCTN statements that may be specified for an IMS/360 Stage 1 system definition. This operand must be specified as a decimal number from 1 to 819. Default value is 350.

TRANS=

specifies the maximum number of TRANSACT statements that may be specified for an IMS/360 Stage 1 system definition. This operand must be specified as a decimal number from 1 to 584. Default value is 584.

TEDITS=

specifies the maximum number of Transaction Edit Routines that will be specified for an IMS/360 Stage 1 system definition. This operand must be specified as a decimal number from 1 to 255 and should not exceed the specified value of the TRANS operand. Default value is 20.

LGRPS=

specifies the maximum number of LINEGRP statements that may be specified for an IMS/360 Stage 1 system definition. This operand must be specified as a decimal number from 1 to 255. Default value is 50.

LINES=

specifies the maximum number of LINE and POOL statements that may be specified for an IMS/360 Stage 1 system definition. This operand must be specified as a decimal number from 1 to 255. Default value is 200.

CUNITS=

specifies the maximum number of CTLUNIT statements that may be specified for an IMS/360 Stage 1 system definition. This operand must be specified as a decimal number from 1 to 255. Default value is 40.

TERMS=

specifies the maximum number of TERMINAL and SUBPOOL statements that may be specified for an IMS/360 Stage 1 system definition. This operand must be specified as a decimal number from 1 to 1000. Default value is 250.

NAMES=

specifies the maximum number of NAME statements and logical terminal names that may be specified for an IMS/360 Stage 1 system definition. This operand must be specified as a decimal number from 1 to 744. Default value is 300.

RTNES=

specifies the maximum number of user routines that may be specified for an IMS/360 Stage 1 system definition. This operand must be specified as a decimal number from 1 to 100. Default value is 20.

JACCT=

specifies any job accounting information that may be desired for placement on the produced JOB card. The length of this operand is limited to 40 bytes. If the operand is specified, a program name of IMS is provided.

SACCT=

specifies any step accounting information that may be desired for placement on the produced EXEC card. The length of this operand is limited to 40 bytes.

JCLASS=

specifies the job class to be assigned to the produced job. Default value is A.

MCLASS=

specifies the output class to be assigned within the produced job. Default value is A.

PRTY=

specifies the job priority to be assigned to the produced job. No PRTY parameter is generated if the operand is omitted.

The following example illustrates the Job Control Language (JCL) for Stage 1 of IMS/360 system definition default resource limits modification. Use the operating system assembler procedure (ASMFC) with the following SYSLIB DD card overide.

```
//              JOB
//              EXEC    ASMFC
//ASM.SYSLIB    DD DSN=IMS2.GENLIB,DISP=SHR
//ASM.SYSIN     DD *

       IMSMAX control statement
              END

/*
```

The resulting output deck completes Stage 1 and is ready for input to Stage 2 execution. Place this deck of cards in the job stream.

The following example illustrates the output that would be produced if the following limitations were specified:

number of data bases to be defined	255
number of application programs to be defined	255
number of transactions to be defined	450
number of transaction edit routines required	20
number of line groups to be defined	255
number of user routines required	20
number of lines to be defined	255
number of control units to be defined	45
number of terminals to be defined	400
number of logical terminal names required	510

```
1          IMSMAX DBASES=255,PGMS=255,TRANS=450,                          C
                 LGRPS=250,LINES=254,CUNITS=48,                           C
                 TERMS=400,NAMES=510


3+         PUNCH '//IMSUPDTE JOB MSGLEVEL=1,MSGCLASS=A,CLASS=A'
4+         PUNCH '//STEP1 EXEC PGM=IEBUPDTE '
5+         PUNCH '//SYSPRINT DD SYSOUT=A,DCB=BLKSIZE=1210'
6+         PUNCH '//SYSUT1 DD DSN=IMS2.GENLIB,DISP=SHR'
7+         PUNCH '//SYSUT2 DD DSN=IMS2.GENLIB,DISP=OLD'
8+         PUNCH '//SYSIN DD *  '
9+         PUNCH './ REPL NAME=DFSGLOBS'
10+        PUNCH './ NUMBER NEW1=1000,INCR=1000'
11+        PUNCH '          GBLA  &&CTL(48) CTLUNIT CONTROL VALUES'
12+        PUNCH '          GBLA  &&CTLTC(250) DEVICE INDEX BYTE'
13+        PUNCH '          GBLA  &&DBKEY(255) DATABASE SORT KEY'
14+        PUNCH '          GBLA  &&DXB1(400) CIB/CRB/CXB VALUES'
15+        PUNCH '          GBLA  &&DXB2(400) CIB/CRB/CXB VALUES'
16+        PUNCH '          GBLA  &&KEY(510)   SORT KEY'
17+        PUNCH '          GBLA  &&LADDR(254) LINE ADDR AND CONTROL'
18+        PUNCH '          GBLA  &&LTS(510) LTERM CONTROL VALUES'
19+        PUNCH '          GBLA  &&PSKEY(255) APPLCTN SORT KEY'
20+        PUNCH '          GBLA  &&TAC(400) TERMINAL CONTROL VALUES'
21+        PUNCH '          GBLA  &&TAD(400) TERMINAL ADDRESS'
22+        PUNCH '          GBLA  &&TF(400) TERMINAL FEATURES/FLAGS'
23+        PUNCH '          GBLA  &&TRKEY(450) TRANSACT SORT KEY'
24+        PUNCH '          GBLA  &&TRP(450) TRANSACT PRIORITIES'
25+        PUNCH '          GBLA  &&TRPC(450) TRANSACT CONTROL VALUES'
26+        PUNCH '          GBLA  &&TRPS(450) TRANSACT PROCLIM'
27+        PUNCH '          GBLA  &&TRSPS(450) TRANSACT SPA SIZE'
28+        PUNCH '          GBLB  &&CIB(400) CIB SWITCH'
29+        PUNCH '          GBLB  &&CRB(400) CRB SWITCH'
30+        PUNCH '          GBLB  &&CWL(48) CTLUNIT WLA SWITCH'
31+        PUNCH '          GBLB  &&DBI(255) INDEX DATABASE SWITCH'
32+        PUNCH '          GBLB  &&DXB(400) CXB SWITCH'
33+        PUNCH '          GBLB  &&GBM1(250) 2780 MODE SWITCH 1'
34+        PUNCH '          GBLB  &&GBM2(250) 2780 MODE SWITCH 2'
35+        PUNCH '          GBLB  &&LGF(250) SWITCHED LINEGRP SWITCH'
36+        PUNCH '          GBLB  &&LGP(250) AUTOPOLL LINEGRP SWITCH'
37+        PUNCH '          GBLB  &&LGPC1(250) BSC POLLING OPTION 1'
38+        PUNCH '          GBLB  &&LGPC2(250) BSC POLLING OPTION 2'
39+        PUNCH '          GBLB  &&LOV(254) COB SWITCH'
40+        PUNCH '          GBLB  &&LTE(510) LTERM EDIT SWITCH'
41+        PUNCH '          GBLB  &&LTU(510) UPPER CASE EDIT SWITCH'
42+        PUNCH '          GBLB  &&PL(254) POOL STATEMENT SWITCH'
43+        PUNCH '          GBLB  &&POPT(255) APPLCTN OPTION SWITCH'
44+        PUNCH '          GBLB  &&POV(255) APPLCTN OVERLAY SWITCH'
45+        PUNCH '          GBLB  &&PQF(255) IQF APPLCTN SWITCH'
46+        PUNCH '          GBLB  &&PT(255) APPLCTN PGMTYPE SWITCH'
47+        PUNCH '          GBLB  &&SPOOL(250) SPOOL LINEGRP SWITCH'
48+        PUNCH '          GBLB  &&SUB(400) SUBPOOL STATEMENT SWITCH'
49+        PUNCH '          GBLB  &&TE(400) TERMINAL EDIT SWITCH'
50+        PUNCH '          GBLB  &&TRE(450) TRANSACT EDIT SWITCH'
51+        PUNCH '          GBLB  &&TRI(450) TRANSACT INQUIRY SWITCH'
```

```
         PUNCH '         GBLB  &&TRM(450) TRANSACT MODE SWITCH'
         PUNCH '         GBLB  &&TRR(450) TRANSACT MSGTYPE 2'
         PUNCH '         GBLB  &&TRS(450) TRANSACT MSGTYPE 1'
         PUNCH '         GBLB  &&TRSP(450) TRANSACT SPA 2'
         PUNCH '         GBLB  &&TU(400) TERMINAL UNIT SWITCH'
         PUNCH '         GBLC  &&DBN(255) DATABASE NAME'
         PUNCH '         GBLC  &&LGD(250) LINEGRP DDNAME'
         PUNCH '         GBLC  &&LTN(510) LOGICAL TERMINAL NAME'
         PUNCH '         GBLC  &&PSBN(255) PSB NAME'
         PUNCH '         GBLC  &&RTNE(20) USER ROUTINE NAME'
         PUNCH '         GBLC  &&SMBE(20) TRANSACT EDIT ROUTINE'
         PUNCH '         GBLC  &&TNAME(400) TERMINAL STATEMENT LABL'
         PUNCH '         GBLC  &&TRC(450) TRANSACTION CODE'
         PUNCH './ REPL NAME=DFSGLOB1'
         PUNCH './ NUMBER NEW1=1000,INCR=1000'
         PUNCH '&&MAX(1)  SETA  255   DATABASE MAXIMUM'
         PUNCH '&&MAX(2)  SETA  255   APPLCTN MAXIMUM'
         PUNCH '&&MAX(3)  SETA  450   TRANSACT MAXIMUM'
         PUNCH '&&MAX(4)  SETA  20   TRANSACT EDIT MAXIMUM'
         PUNCH '&&MAX(5)  SETA  250   LINEGRP MAXIMUM'
         PUNCH '&&MAX(6)  SETA  254   LINE MAXIMUM'
         PUNCH '&&MAX(7)  SETA  48   CTLUNIT MAXIMUM'
         PUNCH '&&MAX(8)  SETA  400   TERMINAL MAXIMUM'
         PUNCH '&&MAX(9)  SETA  510   LTERM MAXIMUM'
         PUNCH '&&MAX(10) SETA  20   USER ROUTINES MAXIMUM'
         PUNCH './ ENDUP'
         PUNCH '/*'
         END
```

The following example illustrates the output that would be produced if the following limitations were specified:

number of data bases to be defined	450
number of application programs to be defined	450
number of transactions to be defined	584
number of transaction edit routines required	20
number of line groups to be defined	10
number of user routines required	4
number of lines to be defined	20
number of control units to be defined	15
number of terminals to be defined	30
number of logical terminal names required	50

```
1            IMSMAX DBASES=450,PGMS=450,LGRPS=10,RTNES=4,                        C
                   LINES=20,CUNITS=15,TERMS=30,NAMES=50,                         C
                   JACCT=(JOB-ACCOUNTING-INFORMATION),                           C
                   SACCT=(STEP-ACCOUNTING-INFORMATION),                          C
                   JCLASS=C,MCLASS=I,PRTY=10


  3+         PUNCH '//IMSUPDTE JOB (JOB-ACCOUNTING-INFORMATION),IMS,'
  4+         PUNCH '//                MSGLEVEL=1,MSGCLASS=I,CLASS=C,PRTY=10'
  5+         PUNCH '//STEP1 EXEC PGM=IEBUPDTE,'
  6+         PUNCH '//                ACCT=(STEP-ACCOUNTING-INFORMATION) '
  7+         PUNCH '//SYSPRINT DD SYSOUT=I,DCB=BLKSIZE=1210'
  8+         PUNCH '//SYSUT1 DD DSN=IMS2.GENLIB,DISP=SHR'
  9+         PUNCH '//SYSUT2 DD DSN=IMS2.GENLIB,DISP=OLD'
 10+         PUNCH '//SYSIN DD *  '
 11+         PUNCH './ REPL NAME=DFSGLOBS'
 12+         PUNCH './ NUMBER NEW1=1000,INCR=1000'
 13+         PUNCH '         GBLA  &&CTL(15) CTLUNIT CONTROL VALUES'
 14+         PUNCH '         GBLA  &&CTLTC(10) DEVICE INDEX BYTE'
 15+         PUNCH '         GBLA  &&DBKEY(450) DATABASE SORT KEY'
 16+         PUNCH '         GBLA  &&DXB1(30) CIB/CRB/CXB VALUES'
 17+         PUNCH '         GBLA  &&DXB2(30) CIB/CRB/CXB VALUES'
 18+         PUNCH '         GBLA  &&KEY(584)   SORT KEY'
 19+         PUNCH '         GBLA  &&LADDR(20) LINE ADDR AND CONTROL'
 20+         PUNCH '         GBLA  &&LTS(50) LTERM CONTROL VALUES'
 21+         PUNCH '         GBLA  &&PSKEY(450) APPLCTN SORT KEY'
 22+         PUNCH '         GBLA  &&TAC(30) TERMINAL CONTROL VALUES'
 23+         PUNCH '         GBLA  &&TAD(30) TERMINAL ADDRESS'
 24+         PUNCH '         GBLA  &&TF(30) TERMINAL FEATURES/FLAGS'
 25+         PUNCH '         GBLA  &&TRKEY(584) TRANSACT SORT KEY'
 26+         PUNCH '         GBLA  &&TRP(584) TRANSACT PRIORITIES'
 27+         PUNCH '         GBLA  &&TRPC(584) TRANSACT CONTROL VALUES'
 28+         PUNCH '         GBLA  &&TRPS(584) TRANSACT PROCLIM'
 29+         PUNCH '         GBLA  &&TRSPS(584) TRANSACT SPA SIZE'
 30+         PUNCH '         GBLB  &&CIB(30) CIB SWITCH'
 31+         PUNCH '         GBLB  &&CRB(30) CRB SWITCH'
 32+         PUNCH '         GBLB  &&CWL(15) CTLUNIT WLA SWITCH'
 33+         PUNCH '         GBLB  &&DBI(450) INDEX DATABASE SWITCH'
 34+         PUNCH '         GBLB  &&DXB(30) CXB SWITCH'
 35+         PUNCH '         GBLB  &&GBM1(10) 2780 MODE SWITCH 1'
 36+         PUNCH '         GBLB  &&GBM2(10) 2780 MODE SWITCH 2'
 37+         PUNCH '         GBLB  &&LGF(10) SWITCHED LINEGRP SWITCH'
 38+         PUNCH '         GBLB  &&LGP(10) AUTOPOLL LINEGRP SWITCH'
 39+         PUNCH '         GBLB  &&LGPC1(10) BSC POLLING OPTION 1'
 40+         PUNCH '         GBLB  &&LGPC2(10) BSC POLLING OPTION 2'
 41+         PUNCH '         GBLB  &&LOV(20) COB SWITCH'
 42+         PUNCH '         GBLB  &&LTE(50) LTERM EDIT SWITCH'
 43+         PUNCH '         GBLB  &&LTU(50) UPPER CASE EDIT SWITCH'
 44+         PUNCH '         GBLB  &&PL(20) POOL STATEMENT SWITCH'
 45+         PUNCH '         GBLB  &&POPT(450) APPLCTN OPTION SWITCH'
 46+         PUNCH '         GBLB  &&POV(450) APPLCTN OVERLAY SWITCH'
 47+         PUNCH '         GBLB  &&PQF(450) IQF APPLCTN SWITCH'
 48+         PUNCH '         GBLB  &&PT(450) APPLCTN PGMTYPE SWITCH'
 49+         PUNCH '         GBLB  &&SPOOL(10) SPOOL LINEGRP SWITCH'
```

```
         PUNCH '         GBLB  &&SUB(30) SUBPOOL STATEMENT SWITCH'
         PUNCH '         GBLB  &&TE(30) TERMINAL EDIT SWITCH'
         PUNCH '         GBLB  &&TRE(584) TRANSACT EDIT SWITCH'
         PUNCH '         GBLB  &&TRI(584) TRANSACT INQUIRY SWITCH'
         PUNCH '         GBLB  &&TRM(584) TRANSACT MODE SWITCH'
         PUNCH '         GBLB  &&TRR(584) TRANSACT MSGTYPE 2'
         PUNCH '         GBLB  &&TRS(584) TRANSACT MSGTYPE 1'
         PUNCH '         GBLB  &&TRSP(584) TRANSACT SPA 2'
         PUNCH '         GBLB  &&TU(30)-TERMINAL UNIT SWITCH'
         PUNCH '         GBLC  &&DBN(450) DATABASE NAME'
         PUNCH '         GBLC  &&LGD(10) LINEGRP DDNAME'
         PUNCH '         GBLC  &&LTN(50) LOGICAL TERMINAL NAME'
         PUNCH '         GBLC  &&PSBN(450) PSB NAME'
         PUNCH '         GBLC  &&RTNE(4) USER ROUTINE NAME'
         PUNCH '         GBLC  &&SMBE(20) TRANSACT EDIT ROUTINE'
         PUNCH '         GBLC  &&TNAME(30) TERMINAL STATEMENT LABL'
         PUNCH '         GBLC  &&TRC(584) TRANSACTION CODE'
         PUNCH './ REPL NAME=DFSGLOB1'
         PUNCH './ NUMBER NEW1=1000,INCR=1000'
         PUNCH '&&MAX(1)  SETA  450   DATABASE MAXIMUM'
         PUNCH '&&MAX(2)  SETA  450   APPLCTN MAXIMUM'
         PUNCH '&&MAX(3)  SETA  584   TRANSACT MAXIMUM'
         PUNCH '&&MAX(4)  SETA  20   TRANSACT EDIT MAXIMUM'
         PUNCH '&&MAX(5)  SETA  10   LINEGRP MAXIMUM'
         PUNCH '&&MAX(6)  SETA  20   LINE MAXIMUM'
         PUNCH '&&MAX(7)  SETA  15   CTLUNIT MAXIMUM'
         PUNCH '&&MAX(8)  SETA  30   TERMINAL MAXIMUM'
         PUNCH '&&MAX(9)  SETA  50   LTERM MAXIMUM'
         PUNCH '&&MAX(10) SETA  4  USER ROUTINES MAXIMUM'
         PUNCH './ ENDUP'
         PUNCH '/*'
         END
```

SYSTEM DEFINITION JOB CONTROL LANGUAGE STATEMENTS

The Job Control Language (JCL) for Stage 1 of system definition is for an assembly execution. Use the standard operating system assembler procedure (ASMFC) with the following SYSLIB DD card override. The user generates a card deck of the following format and places these cards in an OS job stream.

```
//          JOB
//          EXEC  ASMFC,REGION=130K
//ASM.SYSLIB DD DSNAME=IMS2.GENLIB,DISP=OLD
//ASM.SYSIN  DD *


              IMS/360 Stage 1 -
           INPUT CONTROL CARDS -
           SYSTEM DEFINITION PROGRAM


/*
```

The resulting output deck completes Stage 1 and is ready for input to Stage 2 execution. It is suggested that this output deck be listed and examined to understand the content. Prior to Stage 2 execution, the IMS/360 system data sets used in Stage 2 must be allocated and cataloged. Place this deck of cards in the job stream.

Examples of system definition are shown at the end of this chapter.

IMS/360 SYSTEM DATA SETS

As received by the user, the IMS/360 system consists of either two or three unloaded partitioned data sets on an unlabeled 800- or 1600-bpi tape. The data set names are IMS2.GENLIB, IMS2.LOAD, and, optionally, IMS2.SOURCE. The IMS2.GENLIB and IMS2.LOAD data sets are required for execution of the IMS/360 system definition utility and must exist as cataloged partitioned data sets (PDSs) on the generating system. The IMS2.SOURCE PDS contains the source code for all IMS/360 action modules. The IMS2.GENLIB PDS contains the IMS/360 system definition utility macro definitions, the PSB and DBD generation utility macro definitions, and those IMS/360 macro definitions necessary to assemble action modules from the IMS2.SOURCE PDS.

The user must specify the name of a cataloged partitioned data set to be used as output of object modules created during Stage 2 execution. The name of this data set is specified in the IMSGEN macro.

Stage 2 of the IMS/360 system definition utility assumes the existence of the following cataloged partitioned data sets, in addition to those specified in the IMSGEN macro-instruction of Stage 1, on the generated system:

IMS2.RESLIB — Depending on the type of system being generated, Stage 2 places the batch and/or teleprocessing system nucleus, composite online control blocks module, and all action modules loaded by the control program in this library.

IMS2.MACLIB — The macro definition set requested in Stage 1 is placed into this library as required by the type of generation being performed.

IMS2.PROCLIB — All IMS/360-generated cataloged procedures are placed into this data set as required by the type of generation being performed.

The generated IMS/360 system requires that the following data sets and libraries be allocated and cataloged on the operating system used for IMS/360 execution:

IMS2.RESLIB	contains the IMS/360 nucleus and required action modules
IMS2.PGMLIB	contains user-written application programs
IMS2.PSBLIB	contains the user's program specification blocks (PSBs). This data set is used by the block builder utility program and may optionally be used if a DL/I batch region is executing.
IMS2.DBDLIB	contains the user's data base description blocks (DBDs). This data set is used by the block builder utility program and by the utility type (ULU & UDR) regions. It may optionally be used if a DL/I batch region is executing.
IMS2.ACBLIB	contains the application description and data base control blocks produced by the block builder utility program. The data set is required by the IMS/360 Data Base/Data Communications System. Its use in DL/I batch regions is optional.
IMS2.MACLIB	required to execute the DBD and PSB generation utilities
IMS2.SPA	contains the specified direct access device scratchpad areas, as required by the SPAREA statement in Stage 1 of system definition.
IMS2.DBLLOG	is a work data set used by emergency restart to save the data base log records until they are needed by the data base backout routines of the restart facility.
IMS2.SYSOnnn	direct access data set(s) used to store spool SYSOUT data that will be printed by the IMS/360 print utility when the spool SYSOUT option is selected during system definition. The 'nnn' is a numeric suffix assigned sequentially by system definition.
IMS2.QBLKS IMS2.SHMSG IMS2.LGMSG	required by the IMS/360 Data Base/Data Communication system for message queuing.

The various partitioned data sets used by IMS/360 for libraries must be defined and allocated by the user. The DCB characteristics for these data sets should be specified at time of allocation. In all cases, these DCB characteristics should be equated to existing operati system partitioned data sets. This can be done with a DCB= operand of the DD statement used for allocation. The following lists the IMS/360-operating system data sets which would have equivalent DCB characteristics:

IMS/360	Operating System
IMS2.RESLIB	SYS1.LINKLIB
IMS2.PGMLIB	SYS1.LINKLIB
IMS2.PROCLIB	SYS1.PROCLIB
IMS2.MACLIB	SYS1.MACLIB

IMS2.PSBLIB	SYS1.LINKLIB
IMS2.DBDLIB	SYS1.LINKLIB
IMS2.ACBLIB	SYS1.LINKLIB

It is suggested that the Operating System utility program, IEHPROGM, be used to allocate and catalog these IMS/360 system data sets.

To summarize, the different libraries made available or modified by the user or by the system definition program are as follows:

Data Base - Data Communication <u>System Definition</u>		Data Base <u>System Definition</u>
IMS2.RESLIB		IMS2.RESLIB
IMS2.MACLIB		IMS2.MACLIB
IMS2.PSBLIB		IMS2.PSBLIB
IMS2.PGMLIB		IMS2.PGMLIB
IMS2.PROCLIB		IMS2.PROCLIB
IMS2.DBDLIB		IMS2.DBDLIB
IMS2.ACBLIB		IMS2.ACBLIB
IMS2.SPA		--------
SYS1.SVCLIB	(OSAM SVC and	SYS1.SVCLIB
SYS1.LPALIB	channel-end appendage)	SYS1.LPALIB
SYS1.LINKLIB	(Link pack modules)	SYS1.LINKLIB
SYS1.NUCLEUS	(Type 1 and 2 SVCs)	------------
IMS2.DBLLOG		------------
IMS2.SYSOnnn		------------
IMS2.QBLKS	(Message	------------
IMS2.SHMSG	Queue	------------
IMS2.LGMSG	Data sets)	------------

All of the IMS/360 data sets above are to be allocated as the operating system partitioned data sets except IMS2.SPA, IMS2.DBLLOG, IMS2.QBLKS, IMS2.SHMSG, and IMS2.LGMSG. These data sets are to be allocated with DSORG=PS.

IMS/360 SYSTEM DATA SET ALLOCATION

Space allocation for IMS/360 MACRO, PSB, SPA, DBD, PROGRAM, PROCEDURE, and RESLIB libraries depends upon user requirements. Space requirements for user libraries of programs, program specification blocks, and data base definition blocks depend entirely upon the user's operating environment. Some examples may be useful:

- DBD Library - Each DBD (one per data base) requires approximately 1500 to 2500 bytes of direct access storage. Exact requirements depend upon number of data set groups, segments, fields, and hierarchical levels.

- PSB Library - Each PSB (one per program) requires approximately 250 to 500 bytes of direct access storage. Exact requirements depend upon number of data bases (PCB) used in PSB and number of sensitive segments.

- ACBLIB Library - requires space for each PSB and all unique physical DBDs. The space can be calculated from the formulae supplied in the <u>IMS/360 System/Application Design Guide</u> under PSB and DBD storage estimates.

- PROCLIB Library - About 10 tracks (2314) of space are required.

- RESLIB Library - About 5 cylinders of 2314 space are required.

- MACLIB Library - About 10 cylinders of 2314 space are required for ALL macro-instructions. About one cylinder is required for PSBGEN and DBDGEN macro-instructions only.

- PGMLIB Library - This library contains the application programs.

- IMS2.OBSDSET - Allocate one cylinder on 2314.

- IMS2.SPA allocation should be a sequential data set with the SPACE parameter similar to SPACE= (number of SPAs, size of an SPA + 10). Refer to the SPAREA macro.

- IMS2.DBLLOG allocation should be a sequential data set with the space parameter similar to SPACE= (512, X times Y), where: X= the number of processing regions specified in the MAXREGN parameter of the IMSCTRL statement of system definition, and Y= the largest number of data base segments that may be modified during any single scheduling of a user message processing program. If any data base segments are larger than 460 bytes, Y must be adjusted upward by 1 for each 460-byte portion of the segment. If pointers and indexes are used in the data base, Y must also be incremented by 1 for each pointer or index that may be modified.

- The space requirements for all message queue data sets vary depending upon the user's environment. Allocation guidelines are provided in the next section.

- IMS2.SYSnn data sets should be allocated as sequential data sets, with sufficient space in the primary allocation to contain the amount of output desired to be batched by the user.

When the Interactive Query Facility (IQF) feature is incorporated into IMS/360 (with the combined Data Base/Data Communication system) approximately 10 cylinders of 2314 space are required for IQF.

Message Queue Space Allocation

The amount of direct access storage space allocated to the message queue data sets depends upon how many transaction codes and logical terminal names have been specified during system definition, and how many messages, both short and long, are to be held by the system during any period of time. The best way to provide guidance for space allocation is to consider a specific example. Assume:

1. 200 transaction codes were defined. In addition, 10 of the defined transaction codes are used to accumulate error statistics for twice a day batch processing.

   One half of the defined transaction codes are single mode, and the remainder are multiple mode, with processing limit counts of 10.

   200 logical terminal names were defined.

2. The specification in the MSGQUEUE macro was DSETS=(2314),SHUTDWN=140 which will cause a common block size of 576, with:

long message length	=	576	blocking factor 1
short message length	=	192	blocking factor 3
queue block record length	=	48	blocking factor 12

3. Estimated message volume for a 12-hour operating period:

Input -	short	(average ≤	132 characters)	40,000
	long	(average ≤	500 characters)	10,000
Output-	short	(average ≤	132 characters)	25,000
	long	(average ≤	1,000 characters)	25,000

4. Ten blocks will fit on a 2314 track

From the above assumptions,

A. the space required for the IMS2.QBLKS data set can be calculated:

$$\text{tracks} = \frac{(2 * (\text{Transaction codes} + \text{Logical terminals}) + \text{Shutdown})}{(\text{blocking factor} * \text{number of blocks per track})}$$

$$= \frac{(2 * (200 + 200) + 140)}{12 * 10}$$

$$= \frac{940}{120}$$

$$= 8$$

B. The amount of direct access space required for the IMS2.SHMSG and IMS2.LGMSG data sets is dependent on message throughput, since the disk space is reusable as soon as the message to which it was allocated has been processed and is no longer required for recovery in case of failure.

For single-mode transactions, a message space is available as soon as it has been processed by an application program and the program terminates normally or requests the next message.

For multiple-mode transactions, the message spaces are made available only after the application program which processed them terminates normally.

For logical terminal messages, a given message space is made available after the succeeding message has been successfully transmitted to a terminal device.

40,000 short messages are processed against the 200 transaction codes in 12 hours.

1. With an even distribution, that would be 200 per transaction code

   If 10 transaction codes are queued for batch processing every six hours, there would be

2. $\frac{200 \times 10}{2}$ = 1,000 short messages stored at the end of each 6-hour period

3. 100 transaction codes are single mode, so the message space required if the messages were all processed immediately would be 100 short records.

   100 transaction codes are multiple mode, with processing limit count of 10. Therefore, each transaction code could hold 10 message spaces for processed messages before releasing them.

4. 100 x 10 = 1,000 short messages

25,000 short messages are processed against the 200 logical terminals in 12 hours. With an even distribution, that would be 125 short messages per logical terminal. Because messages may be produced for an output device which is not in operation (terminal turned OFF in a remote location, etc.), allocation must include space to save these messages.

5. Estimate 20 of the logical terminals transmit no output.

6. 20 x 125 = 2500 short messages

The remaining logical terminals would have one message in process and one prior message held.

7. 180 x 2 = 360 short messages

Space must be allocated in the IMS2.SHMSG data set to hold

8.

1,000	held input messages
100	single mode input messages
1,000	multiple mode input messages
2,500	held output messages
360	processing output messages
140	messages reserved for shutdown
5,100	

9. Space required $\frac{5,100}{3 * 10}$ = 170 tracks

C. The amount of direct access space required for the IMS2.LGMSG data set utilizes the same assumptions as in B., with the following difference:

Since the long message output is approximately 1000 bytes, two records are required for each message.

25,000 x 2 = 50,000 long output message records

1. $\frac{10,000}{200}$ = 50 long messages per transaction code

2. $\frac{50 \times 10}{2}$ = 250 input messages held

3. 100 input single mode

4. 100 x 10 = 1,000 multiple mode

5. $\frac{50,000}{200}$ = 250 message records per logical terminal

6. 20 x 250 = 5,000 output messages held

7. 180 x 2 = 360 processing output

8.

250	held input messages
200	single mode input messages
1,000	multiple mode input messages
5,000	
360	
140	
6,950	

9. Space required = $\frac{6,950}{1 \times 10}$ = 695 tracks

D. From the above, space allocation in 2314 cylinders may be determined

	Tracks	+19/20	Cylinders
IMS2.QBLKS	8	$\frac{24}{20}$	1
IMS2.SHMSG	170	$\frac{189}{10}$	9
IMS2.LGMSG	695	$\frac{714}{20}$	35

Message Queue Space Allocation - Secondary

For most efficient operation, message queue data set space should be allocated in terms of contiguous cylinders. Secondary allocation is ignored unless the secondary space has been preallocated (that is, multivolume data set with preallocated space on both volumes).

SYSOUT Data Set Allocation

Space for spool SYSOUT data sets may be allocated by the user as required, but the user must not specify secondary allocation. DCB parameters DSORG=PS and RECFM=UM are required. If not supplied, these parameters will be set automatically. Blocksize can be specified on the DD card, but it may be adjusted downward if it is larger than the system definition specification.

Records written to this data set are standard OS variable-length blocked records. The designation of the undefined-record format specification reduces the buffer space requirement in the IMS/360 control region. Minimum block size is 20 bytes, but this would normally be sufficient to accommodate one print line. Maximum block size would be the track size of the device on which the data set is allocated.

The buffer size specified in the LINE macro instruction for system definition must be at least 16 bytes larger than the desired block size.

Direct Output Data Sets

For direct SYSOUT lines defined to IMS, any valid output device, supported by operating system BSAM access method, may be used. The following record formats can be specified: F, FM, FB, FBM, FBS, FBSM, V, VM, VB, and VBM. Blocksizes may be specified, but will be adjusted downward at execution time, if they are in excess of system generation maximums.

For fixed record formats, the system-defined buffer size must be at least 20 bytes longer than the DCB block size for the data set. For variable-length records, buffer size must be 16 bytes longer than the desired block size. Logical record specifications may be selected by the user to accommodate the data to be written and are restricted as follows:

a. For fixed-format records, the blocksixe must be an even multiple of logical record length.

b. For variable-format records, maximum logical record length is equal to block size minus 4.

If no DCB parameters are supplied, the following defaults apply:

Device Type	RECFM	LRECL	BLKSIZE
1403, 1443	VM	125	129
3211	VM	137	141
2540P	V	85	89
2400 series tape	VBM	125	Note 1
DASD	VBM	125	1/4 Track

Note: Blocksize is only dependent on system definition buffer size. Each segment is treated as a logical record. When blocking is specified, all segments of a message will be contained within a block, unless the block size is not large enough to hold them.

Fixed length segments are always padded with trailing blanks. If blocking is used, the balance of the block will also be padded if a message does not contain the same number of segments as logical records in the block.

Tape blocks are never shorter than 18 bytes regardless of the record format.

Since volume switching is provided by operator command when tape is used, it is necessary that the volume count subparameter of the VOLUME keyword, on the associated DD card, be specified with a large value; for example, 99. If this procedure is used in an IMS system where binary synchronous devices are also operating and only one tape drive is allocated, timeout problems can occur.

Use of SYSIN for Local Card Reader Lines

When a SYSIN data stream is assigned to a local card reader line, the following conditions apply:

a - IMS running as a problem program; the DD card referring to the line must contain DCB=BLKSIZE=80.

b - IMS running as a system task; the SYSIN test stream must be placed in a sequential data set with the DCB attributes: RECFM=F, BLKSIZE=80, and pointed to by a DD card for a reader line within the IMS procedure.

3270 Format Services Library Allocation Guidelines

The DD cards below show the initial minimum allocation that should be made for the 3270 format services libraries based upon 2314 devices.

The IMS2.REFERAL and IMS2.FORMAT data sets should have additional one (1) track allocated for each user defined FMT/MSG descriptor. The first six (6) data sets below constitute two (2) HIDAM data bases used to remember inter-descriptor relationships. More precise storage allocation for the data bases can be performed by consulting the DBD definition source members, DFSUTS30, DFSUTS40, DFSUTS50, and DFSUTS60. These definitions are contained within the distribution DCSOURCE library. With the DBD source members, normal DL/I allocation computation applies.

```
//FMTINDX  DD       DSN=IMS2.FMTINDX,
//                  DCB=DSORG=IS,
//                  DISP=(NEW,KEEP),
//                  UNIT=sysda,
//                  VOL=SER=serial,
//                  SPACE=(CYL,(5,,1))
//FMTINDXO DD       DSN=IMS2.FMTINDXO,
//                  DCB=DSORG=PS,
//                  DISP=(NEW,KEEP),
//                  UNIT=sysda,
//                  VOL=SER=serial
//                  SPACE=(CYL,(1,1))
//FMTDB    DD       DSN=IMS2.FORMATS,
//                  DCB=DSORG=PS,
//                  DISP=(NEW,KEEP),
//                  UNIT=sysda,
//                  VOL=SER=serial,
//                  SPACE=(CYL,(1,1))
//MSGINDX  DD       DSN=IMS2.MSGINDX,
//                  DCB=DSORG=IS,
//                  DISP=(NEW,KEEP),
//                  UNIT=sysda,
//                  VOL=SER=serial,
//                  SPACE=(CYL,(5,,1))
//MSGINDXO DD       DSN=IMS2.MSGINDXO,
//                  DCB=DSORG=PS,
//                  DISP=(NEW,KEEP),
//                  UNIT=sysda,
//                  VOL=SER=serial,
//                  SPACE=(CYL,(1,1))
//MSGDB    DD       DSN=IMS2.MESSAGES,
//                  DCB=DSORG=PS,
//                  DISP=(NEW,KEEP),
//                  UNIT=sysda,
//                  VOL=SER=serial,
//                  SPACE-(CYL,(1,1))
//HISTORY  DD       DSN=IMS2.REFERAL,
//                  DCB=(RECFM=U,DSORG=PO,
//                  BLKSIZE=.ge.IMS2.LOAD),
//                  DISP=(NEW,KEEP),
//                  UNIT=sysda,
//                  VOL=SER=serial,
//                  SPACE=(CYL,(5,2,10))
//ONLINE   DD       DSN=IMS2.FORMAT,
//                  DCB=DSORG=PO,
//                  DISP=(NEW,KEEP),
//                  UNIT=sysda,
//                  VOL=SER=serial,
//                  SPACE=(CYL,(5,2,10))
```

The 'FMTINIT' procedure is placed in the user specified procedure library by IMS System Definition, Stage II. 3270 users must execute this procedure prior to on-line execution. To initiate the 'FMTINIT' procedure enter from the O.S. system console:

S IMSRDR,MBR = FMTINIT

The user should scratch and reallocate the format utility data sets prior to the building of the default blocks.

If at any subsequent time the format utility data bases are scratched and reallocated, the user should scratch and reallocate the format utility data sets and rerun the FMTINIT procedure.

SYSTEM DEFINITION GUIDE

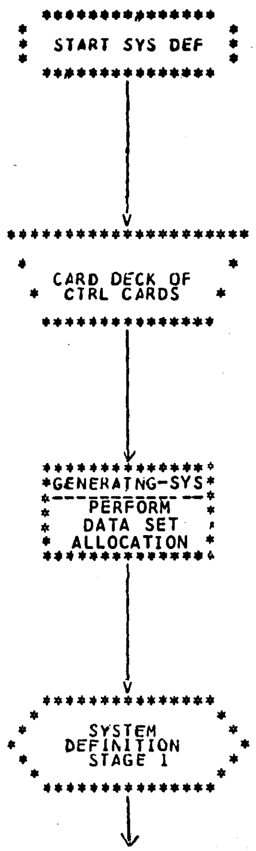
Execution of the system definition utility is shown in general flowchart form following. It provides for both Stage 1 and Stage 2 of system definition and all the other requirements to make IMS/360 operative.

System Definition Job Steps

The steps to be accomplished in performing an IMS/360 system definition are listed below. Depending on the individual definition involved, certain of the assemblies may be skipped. However, all steps must be accomplished in the order listed.

- Copy macros to IMS2.MACLIB.
- Add cataloged procedures to IMS2.PROCLIB.
- Assemble the batch nucleus.
- Assemble the PSB directory, SMB, and the DMB directory.
- Assemble CLBs, LERBs, line DCBs, communication open lists and polling lists, CTBs, COBs, CNTs, and CTM.
- Assemble CCBs, CTTs, and translate tables.
- Assemble CVBs.
- Assemble the SDB.
- Assemble system control blocks (XC, IOBs, MSG DCBs, PSTs, save areas, and queues)
- Assemble secondary SCD.
- Assemble SCD.
- Link-edit reentrant modules.
- Link-edit reusable modules.
- Link-edit overlay modules.

Start system definition.

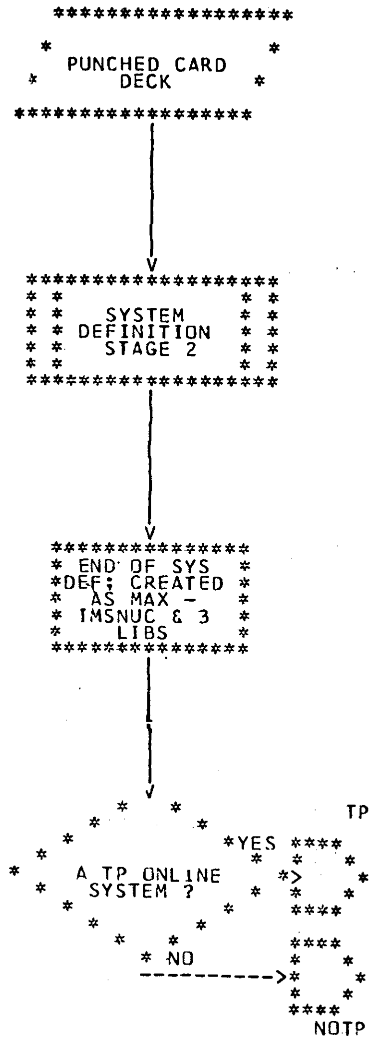


Stage 1 system definition is a deck of control cards prepared from the macro-instructions of system definition shown earlier in this chapter.

The computer system used to execute Stage 1 and Stage 2 of system definition need not be the actual IMS/360 computer system. However, the version of the operating system used for Stage 2 must be the same version under which the defined system will be executed. If it is not the same system, perform data set allocation for IMS2.GENLIB, IMS2.LOAD, and specify the data set in the OBJDSET of the operand of the IMSGEN macro.

System definition Stage 1 requires an assembly run for compilation of the control statements. The assembly requires its SYSLIB DD statement to point to the IMS/360 system definition macro data set IMS2.GENLIB.

Go to next page



System definition Stage 2 takes as input a punched deck of cards created as output from Stage 1.

Perform Stage 2 of system definition.

When Stage 2 is complete, system definition creates an IMS/360 nucleus and establishes content in three libraries (as maximum output), IMS2.RESLIB, IMS2.MACLIB, and IMS2.PROCLIB, on the preallocated data sets specified on the generating system.

Is the IMS/360 system that was first generated a teleprocessing (online) system? Yes, go to label TP. No, go to label NOTP.

```
              TP
            ***************
***         *             *
*   *       * PERFORM OS  *
*    -->*     DATA SET    *
*   *       * ALLOC (MSG  *
***         * QUE & LIBS) *
            ***************
                  |
```

Label TP: If the system being generated is to process message processing and/or batch message processing programs, the message queue data sets must be allocated on the computer system under which IMS/360 is to operate.

```
          NOTP    V
            ***************
***         *             *
*   *       *PERFORM PSB &*
*    -->*DBD DATA SET     *
*   *       * ALLOCATION  *
***         *             *
            ***************
                  |
```

Label NOTP: Perform PSB and DBD data set allocation, whether generated system is a message processing and batch message processing, or a batch processing region system.

```
                  V
            ***************
            *   CONNECT   *
            *             *
            *  LNKLST00   *
            *             *
            ***************
                  |
```

Connect IMS2.RESLIB to system link library using the LNKLST00 member of SYS1.PARMLIB.

```
                  V
            ***************
            *  MOVE OSAM  *
            *CHAN END APP *
            *& SVC MODULE *
            *     TO      *
            * SYS1.SVCLIB *
            ***************
                  |
```

System definition creates OSAM channel end appendage and Type 4 SVC modules and places these in IMS2.RESLIB. OSAM channel end appendage and SVC modules must be moved to SYS1.SVCLIB (MFT, MVT, and VS1) or SYS1.LPALIB (VS2). (Described later in this manual.) If this is a generation for an OS/VS system, the privileged name table and program properties table must be modified to include the name DFSRRC00.

```
                  V
               *     *      USER-APPL
            *           *
         *               *NO    ***
      *                    *  *     *
    *     TP ONLINE         *>      *
      *   SYSTEM ?         *  *     *
         *               *      ***
            *           *
               *     *
                  *  YES
                  |
                  V
```

Again, is this an online (teleprocessing) system? No, go to label USER-APPL.

Go to next page

```
**************
*            *
*  LINK EDIT *
*  SVC'S TO  *
*     OS     *
*   NUCLEUS  *
**************
      |
      V
```

Two of the user SVC load modules created in Stage 2 of system definition must be link-edited into the OS nucleus prior to attempting to execute IMS/360.

```
**************
*            *
*   MODIFY   *
* PRIVILEGED *
*NAME TABLE - *
*SYS1.LINKLIB *
**************
      |
      V
```

In order to initiate the IMS/360 online control program through a START IMS command, the name DFSRRC00 must be added to the OS privileged name table (described later in this chapter).

```
**************
*            *
*    MOVE    *
* IMS,IMSRDR *
*  PROCS TO  *
*SYS1.PROCLIB *
**************
      |
      V
```

The IMS and IMSRDR procedures must be copied from IMS2.PROCLIB to SYS1.PROCLIB for proper execution of the START IMS command and the /START REGION or /START MSG REGION master terminal command.

```
         *   *          USER-APPL
       *       *
     *           *NO    ***
   *   SECURITY    *  *     *
 *   MAINTENANCE    *>      *
   *   REQ'D ?     *  *     *
     *           *      ***
       *       *
         *   *
           * YES
           |
           V
```

Message region procedures which are going to be specified in the /START REG Command must be added to IMS2.PROCLIB.

Go to next page

```
   *******************
  *                 *
 *  CARD DECK OF   *
*   CTRL CARDS    *
******************
         |
```

Set up control cards for input card deck to the security maintenance program. (See the description of control cards in Chapter 4 of this manual.)

```
         V
***************
*             *
*RUN SECURITY *
* MAINTENANCE *
*   PROGRAM   *
*             *
***************
         |
```

Execute the security maintenance program (SMP).

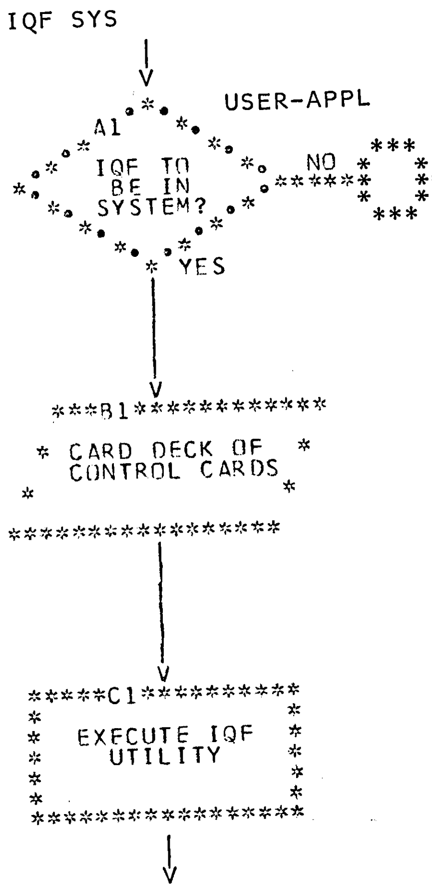
```
         V
***************
*             *
*OUTPUT OF SEC*
* MAINT PGM - *
*   CHANGES   *
*   IMSNUC    *
***************
         |
```

The output of SMP is added to the IMS2.RESLIB library

```
         V
***************
*NRESTART     *
*-------------*
*INITIATE NEW *
* SECURITY PG *
*             *
***************
         |
         V
```

The result of the SMP does not become effective until the next normal restart (See Security Maintenance Program (SMP), Chapter 4 of this manual, for more details.)

Go to next page

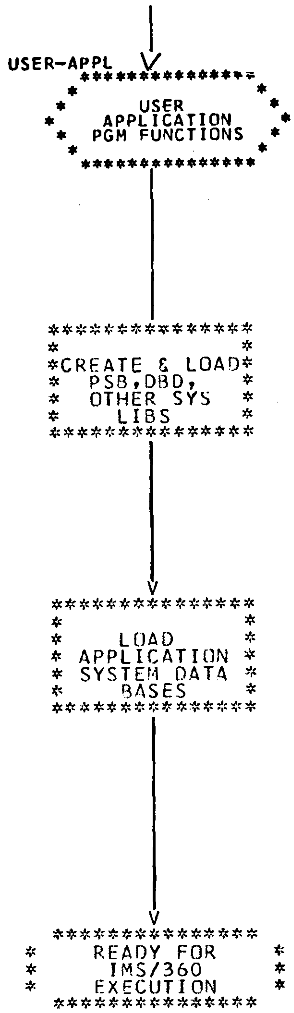


If the Interactive Query Facility (IQF) feature is to be included in the IMS system, an additional process is required.

Set up IQF Utility control card deck as described in the <u>IMS/360 Version 2 Utilities Reference Manual</u> (SH20-0915).

Execute the IQF Utility, Creating the IQF Processor data bases.

Go to next page



Label USER-APPL: The user must perform all the application programming functions; that is, load his application programs in the libraries and the names in the directories, etc.

The PSB and DBD data sets must be loaded prior to executing IMS/360. Any other IMS/360 oriented procedures or data sets desired could also be loaded at this time.

The application system data bases must be loaded before execution of IMS/360 can proceed.

Execute IMS/360 ("IPL") per instructions in Chapter 6 of the <u>IMS/360 Operator's Manual</u>.

PREPARATION OF OPERATING SYSTEM FOR IMS/360

The successful completion of system definition Stage 2 has done the following:

1. Placed the following modules in IMS2.RESLIB:

   a. The OSAM channel end appendage routine.

   b. The OSAM and other Type 4 SVC routines.

   c. The Type 1 and 2 SVC routines.

   d. The Type 1 and 2 SVC interface routines.

   e. The region control module DFSRRC00.

2. Placed the cataloged procedures IMS and IMSRDR in IMS2.PROCLIB.

The IMS/360 system user must perform the following functions:

1. Link edit the Type 1 and 2 SVC interface modules with his operating system nucleus.

2. Link edit the OSAM channel end appendage and SVC modules into SYS1.SVCLIB (MFT, MVT, and VS1) or SYS1.LPALIB (VS2).

3. Move the cataloged procedures IMS and IMSRDR to SYS1.PROCLIB.

4. Concatenate IMS2.RESLIB with SYS1.LINKLIB by a change to SYS1.PARMLIB. The member name to be changed in SYS1.PARMLIB is LNKLST00. See the System/360 Operating System, System Programmer's Guide (GC28-6550).

5. Change the operating system privileged name list in SYS1.LINKLIB to include DFSRRC00. The member name is IEEVLNKT. See the System/360 Operating System, Job Management Program Logic Manual (GY28-6660). This step must be performed for both the Data Base/Data Communications System and the Data Base-only System in an OS/VS environment.

6. If the online system contains 7770-3 lines, link edit the 7770-3 channel/abnormal end appendage and additional load of the Type 4 SVC into SYS1.SVCLIB (MFT, MVT, and VS1) or SYS1.LPALIB (VS2).

IMS/360 SUPERVISOR CALL ROUTINES

The IMS/360 system utilizes three supervisor call (SVC) routines. One of these is used for interregion communication. The second routine is used for IMS/360 online control program functions. The third routine is used by OSAM to create its multivolume data extent block (DEB). All three routines must be specified during system definition for the IMS/360 system even if only a batch-only Data Base System is being generated. IMS/360 system definition creates these routines with user-defined SVC numbers. The next section of this chapter explains how to perform the link edit.

INCLUSION OF IMS/360 SVC ROUTINES IN THE OPERATING SYSTEM NUCLEUS

Two user SVC routines must be added to the operating system nucleus and one to SYS1.SVCLIB (MFT, MVT, and VS1) or SYS1.LPALIB (VS2) for execution of the IMS/360 system. These routines are not required if the batch Data Base System execution is used *exclusively*. These SVC routines are created by IMS/360 system definition from macro-instructions. The SVC numbers to be used may be specified by the IMS/360 system user. The load modules which represent the SVC routines are placed in IMS2.RESLIB by system definition. Two SVC routines are used by the IMS/360 online system. A Type 1 SVC is used for interregion communications, and a Type 2 SVC is used for IMS/360 control functions. The SVC routines used for OSAM are part of a Type 4 SVC. The batch system (data base) uses only the OSAM facility of the Type 4 SVC.

When the IMS/360 user performs his operating system generation, provision must be made for the later incorporation of the SVC routines. The IMS/360 SVC routines need not and normally would not be incorporated at operating system generation. They may, however, be incorporated at that time, if desired. The following SVCTABLE macro-instruction operands should be included in the Stage I input to the OS system generation no matter when the SVC routines are incorporated.

```
(For OS)    SVCTABLE  SVC-nnn-T1-S0,SVC-nnn-T2-S6,SVC-nnn-T4-S6

(For VS1)   SVCTABLE  SVC-nnn-D1-S0,SVC-nnn-E2-S6,SVC-nnn-E4-S6

(For VS2)   SVCTABLE  SVC-nnn-D1,SVC-nnn-E2,SVC-nnn-E4
```

If the actual SVC routines are not incorporated during the operating system system generation, three "dummy" load modules should be placed in the RESMODS partitioned data set. This should be done prior to Stage II of operating system system generation. These modules are of the format:

```
For SVC Types 1 and 2        For Type 4 SVC
    IGCnnn  CSECT            IGC00nnn     CSECT
            BR  14                        BR  14
            END                           END
```

where nnn is the unique SVC number. This effectively "no-ops" the SVC number.

The alternate approach, which would cause inclusion of the actual SVC routines during operating system system generation, requires placement of the actual SVC modules into the partitioned data set referred to by the RESMODS macro-instruction. This would require IMS/360 system definition execution prior to Stage II of operating system system generation. The RESMODS control card could then refer to the IMS2.RESLIB data set for the incorporation of the SVC routines.

If the SVC routines are added after operating system generation, the technique for the resident SVCs is to relink-edit the operating system nucleus. Basically, this involves replacing the "dummy" SVC routines through the link-edit with the actual SVC routines. The best explanation for performing this link-edit is to:

1. Start with JCL and control cards of link-edit step from Stage II of OS system generation.

2. Provide an additional DD statement allocating the IMS2.RESLIB data set to access the SVC modules.

3. Provide an additional DD statement to allocate the SYS1.NUCLEUS data set other than //SYSLMOD.

4. Provide additional INCLUDE control cards for the two SVC routines from IMS2.RESLIB immediately after the INSERT control cards of the original link-edit.

5. Replace the INCLUDE cards from the original operating system

nucleus link-edit with one INCLUDE card for the old operating system nucleus (that is, the one without the SVC routines).

6. Provide a NAME card for the new operating system nucleus (for example, IEANUC02).

It may be good practice to consider the output from the link-edit of the nucleus as another member in SYS1.NUCLEUS (for example, IEANUC02). The _OS Operators Manual_ explains how to IPL an alternate operating system nucleus. If everything executes properly, then IEANUC02 can be renamed IEANUC01.

OSAM CHANNEL END APPENDAGE AND TYPE 4 SVC

OSAM requires a channel end appendage module and Type 4 SVC modules created as load modules during execution of IMS/360 system definition. The modules are distributed on IMS2.LOAD and are renamed during system definition. The renamed modules are placed in IMS2.RESLIB.

It is the user's responsibility to link edit the created OSAM channel end appendage module and Type 4 SVC modules from IMS2.RESLIB to SYS1.SVCLIB (MFT, MVT, and VS1) or SYS1.LPALIB (VS2).

Type 4 SVC load modules and channel end appendage module to be placed into SYS1.SVCLIB (MFT, MVT, and VS1) or SYS1.LPALIB (VS2).

IMS2.LOAD	IMS2.RESLIB	
DFSVC400	IGC00nnn	Basic interface module
DFSVC410	IGC01nnn	System control
DFSAOSA0	IGC04nnn	OSAM open/close EOV entry and initialization
DFSAOSB0	IGC05nnn	OSAM open
DFSAOSC0	IGC06nnn	OSAM open/EOV
DFSAOSD0	IGC07nnn	OSAM open/close EOV final processing
DFSICSC0	IGC10nnn	Issue START and STOP commands
DFSVCI00	IGC11nnn	Type 1 and 2 vector initialization
DFSVC440	IGC13nnn	VS system initialization

where:

nnn

is the specified Type 4 SVC number converted to a signed decimal number.

WARNING: If the low-order digit of the SVC number is a zero, the signed integer value of zero is an unprintable character. For example, assuming a specified Type 4 SVC value of 240, the Type 4 SVC module names would appear on listings as IGC0024b (where lowercase b indicates a blank).

DFSIOCE0 IGG019XX channel-end appendage

where:

xx
: is the specified channel end appendage number

These modules must be link edited with the RENT, REFR attributes.

7770-3 CHANNEL/ABNORMAL END APPENDAGE AND TYPE 4 SVC

The 7770-3 device support requires a channel/abnormal end appendage module and a load of the Type 4 OSAM SVC modules created as load modules during execution of the IMS/360 system definition. The modules are distributed on IMS2.LOAD and are renamed during system definition. The renamed modules are placed in IMS2.RESLIB.

It is the user's responsibility to link edit the created 7770-3 modules from IMS2.RESLIB to SYS1.SVCLIB (MFT, MVT, and VS1) or SYS1.LPALIB (VS2).

The appendage and Type 4 SVC load to be placed into SYS1.SVCLIB (MFT, MVT, and VS1) or SYS1.LPALIB (VS2) are:

IMS2.LOAD	IMS2.RESLIB	
DFSAAP10	IGG019xx	appendage module
DFSASV10	IGC12nnn	Type 4 load

where:

xx
: is the appendage suffix specified on the IMSCTF macro

nnn
: is the Type 4 SVC number converted to a signed decimal number from the value specified in the IMSCTF macro.

These modules must be link edited with the 'RENT,REFR' attributes.

IMS AND IMSRDR PROCEDURES

The IMS and IMSRDR procedures are used to initiate the IMS/360 control program region and message regions. These are placed into IMS2.PROCLIB by system definition. The user must copy these procedures to SYS1.PROCLIB prior to IMS/360 execution. Copies of these procedures are supplied later in this chapter.

CONCATENATING IMS2.RESLIB TO SYS1.LINKLIB

Since the IMS/360 control program region operates as a system task, the modules which comprise it must exist in SYS1.LINKLIB. It is the user's responsibility to concatenate IMS2.RESLIB with SYS1.LINKLIB. This is performed by modification of the member in SYS1.PARMLIB named LNKLST00. (This is also required for the DL/I batch region.) An example of this member is provided below in its modified format.

LNKLST00

SYS1.LINKLIB,IMS2.RESLIB

PRIVILEGED PROGRAM NAME LIST MODIFICATION

The IMS control program operates as an OS system task and therefore must be started from the OS operator's console with an OS START command. The IMS/360 procedure named IMS is invoked by the 'START IMS' command. This procedure, which must exist in SYS1.PROCLIB, initiates the IMS control program as a system task. The first IMS/360 module to gain control is the region controller, DFSRRC00. The name of this module must be included in the OS privileged program name list. To accomplish this, OS module IEEVLNKT must be reassembled to include the name DFSRRC00. After reassembly the new CSECT must be link edited and renamed depending upon the type of OS system, MFT, MVT, VS1, or VS2 being used. It may be necessary to include the IEEVLNKT CSECT as part of several modules. The following examples show the JCL required for a reassembly and link edit of the privileged program name list for Release 20 of OS. These examples are intended as guidelines, and although they are accurate for Release 20.1 of OS, future changes of OS may require modifications to the JCL examples. Information about the privileged program name list can be found in the OS Job Management PLMs, Forms GY28-6660 for MVT and GY27-7128 for MFT. To verify that all modules have been correctly changed, the OS system generation stage I output should be checked to see where the name list is used.

Example I. Assembly and Linkedit of the privilleged name table (CSECT IEEVLNKT) into SYS1.LINKLIB. This applies to MVT, MFT, VS1, and VS2.

```
//IEEVLNKT JOB      1,IMS,MSGLEVEL=1
//STEP1    EXEC     ASMFCL
//ASM.SYSIN DD     *
IEEVLNKT   CSECT
REFTBL     EQU  *
           DC       CL8'IEFIRC'       ***
           DC       CL8'IEFSD080'       *
           DC       CL8'IEEVMNT2'       *   ASSUMED PREVIOUS
           DC       CL8'IEFVRRC'        *   CONTENT OF
           DC       CL8'IEFVMA'         *   PRIVILEGED
           DC       CL8'IEFIIC'         *   PROGRAM
           DC       CL8'IKJEFF40'       *   NAME LIST
           DC       CL8'IKJFATRC'       *
           DC       CL8'IHLGTF'         *
           DC       CL8'HHLGTF'         *
           DC       CL8'AHLGTF'         *
           DC       CL8'HASP'         ***
           DC       CL8'DFSRRC00'       IMS REGION CONTROLLER ADDED
REFEND     DC       X'00'
*           TABLE 2     PROGRAMS FOR STARTABLE SYSTEM TASKS WHICH
REFTBL2    EQU   *      ARE NOT TO HAVE DATA SET INTEGRITY
           DC       CL8'IEFSD080'
REFEND2    DC       X'00'
            END
/*
//LKED.SYSLMOD DD DSN=SYS1.LINKLIB(IEEVLNKT),DISP=SHR
/*
```

Example II. Link Edit JCL for a MVT System

```
//MVTLINK   JOB  1,IMS,MSGLEVEL=1
//LNK      EXEC LKED,PARM.LKED='NCAL,LIST,XREF,LET'
//LKED.SYSLMOD  DD  DSN=SYS1.LINKLIB,DISP=SHR
//LKED.SYSIN  DD  *
   INCLUDE SYSLMOD(IEEVLNKT)
   INCLUDE SYSLMOD(IEEVRCTL)
   ENRTY   IEEVRCTL
   NAME    IEEVRCTL(R)
/*
```

Example III. Link Edit JCL For an MFT System

```
//E              EXEC     LKED,PARM.LKED='NCAL,LIST,XREF,LET'
//LKED.CI505     DD       DSN=SYS1.CI505,VOL=SER=DL1B02,UNIT=2314,DISP=SHR
//LKED.SYSLMOD DD         DSN=SYS1.LINKLIB,DISP=OLD
//LKED.SYSIN     DD       *
    INCLUDE          SYSLMOD(IEEVLNKT)        NEW NAME LIST
    INCLUDE          CI505(IEEVRJCL,IEEPSN)
    ENTRY            IEEVRJCL
    ALIAS            IEEPSN
    NAME             IEEVRJCL(R)
    INCLUDE          CI505(IEEVSMSG)
    INCLUDE          CI505(IEEVACTL,IEEVSMBA)
    INCLUDE          SYSLMOD(IEEVLNKT)        NEW NAME LIST CSECT
    ENTRY            IEEVACTL
    NAME             IEEVACTL(R)
/*
```

Example IV. Link Edit JCL For An OS/VS1 System

```
//E              EXEC     LKED,PARM.LKED='NCAL,LIST,XREF,LET'
//LKED.SYSLMOD DD         DSN=SYS1.LINKLIB,DISP=SHR
//LKED.SYSIN     DD       *
    INCLUDE          SYSLMOD (IEEVLNKT)       NEW NAME LIST CSECT
    INCLUDE          SYSLMOD (IEEVRCTL)
    ALIAS            IEEVIC
    ENTRY            IEEVRCTL
    NAME             IEEVRCTL (R)
    INCLUDE          SYSLMOD (IEEVLNKT)       NEW NAME LIST CSECT
    INCLUDE          SYSLMOD (IEFIRC)
    ENTRY            IEFIRC
    NAME             IEFIRC(R)
```

Example V. Link Edit JCL for a VS2 System

```
//VS2LINK  JOB  1,IMS,MSGLEVEL=1
//LK      EXEC LKED,PARM.LKED='NCAL,LIST,XREF,LET'
//LKED.LINK DD DSN=SYS1.LINKLIB,DISP=SHR
//LKED.SYSLMOD DD DSN=SYS1.LPALIB,DISP=SHR
//LKED.SYSIN  DD  *
   INCLUDE LINK(IEEVLNKT)
   NAME IEEVLNKT(R)
/*
```

PROGRAM PROPERTIES TABLE MODIFICATION IN OS/VS1

When executing in an OS/VS1 system, IMS/360 requires the authorized subsystem status. To accomplish this, OS/VS1 CSECT IEFSDPPT must be reassembled to include name DFSRRC00. This CSECT is comprised of four sections, each with its own entry point. The fourth section, with an entry point named AUTHSEC, should be changed to include the name DFSRRC00. This CSECT should then be re-linked into module IEFSD161. The OS/VS1 SYSGEN should be consulted to determine the correct method for replacing this module.

PROGRAM PROPERTIES TABLE IN OS/VS2

No changes are required to the program properties table in OS/VS2.

DFSILNK0 TO SYS1.LINKLIB

Prior to using any of the generated IMS/360 assembler or compiler procedures, the user must either link edit module DFSILNK0 from IMS2.RESLIB to SYS1.LINKLIB or must have concatenated IMS2.RESLIB to SYS1.LINKLIB through the LINKLST00 member of SYS1.PARMLIB. This module permits the use of SHR disposition on SYSLMOD data sets in the link steps of procedures. This module invokes the linkage editor under the alias name LINKEDIT.

OPERATING SYSTEM LINK PACK MODULES

In an operating environment where there are several batch regions or a combination of online and batch regions, it may be advantageous to place some of the frequently used IMS and BISAM modules in the operating system RAM area (MFT or VS1) or link pack area (MVT or VS2). The following is a list of loaded modules which are reentrant and are recommended candidates for the RAM or link pack area. The modules to be included must previously exist in either the SYS1.SVCLIB or the SYS1.LINKLIB or its concatenations. (IMS2.RESLIB should be concatenated with SYS1.LINKLIB.) To determine additional candidates for the link pack area, refer to the list of IMS modules with their linkage editor attributes provided in Chapter 1 of the IMS/360 System Manual.

From SYS1.LINKLIB

Module name	Module definition
DFSDLA00	Data Language/I Analyzer
DFSDLR00	Data Language/I Retrieve
DFSDBH00	Data Base Buffer Handler
DFSDLE00	Data Language/I Insert
DFSDLD00	Data Language/I Delete/Replace
DFSAOS50	OSAM Common Subroutines
DFSAOS20	OSAM Read/Write
DFSAOS30	OSAM CHECK
DFSDISM0	Data Language/I ISAM Simulator
DFSDHDS0	Data Language/I Space Management
DFSARW00	7770-3 Read/Write Module
DFSFCPY0	Interregion Copy Routine

From SYS1.SVCLIB

Module name	Module definition
IGG019Z9	OSAM Channel End Appendage
IGG019GX	BISAM Asynchronous Read/Write
IGG019G9	BISAM Appendage with Write Check
IGG019JV	BISAM Non-Privileged Macro-Time Read/Write
IGG019J7	BISAM Privileged Macro-Time Read/Write
IGG019Z8	7770-3 Channel/Abnormal End Appendage
IGC00NNN	OSAM Basic Interface Module (DFSVC400)
IGC04NNN	OSAM Open/Close/EOV (DFSAOSA0)
IGC05NNN	OSAM Open (DFSAOSB0)
IGC06NNN	OSAM Open/EOV (DFSAOSC0)
IGC07NNN	OSAM Open/Close/EOV (DFSAOSD0)

The last two characters of module IGG019Z9 (OSAM channel end appendage) and module IGG019Z8 (7770-3 channel/abnormal end appendage) are determined by the IMS/360 user during system definition.

OPERATING SYSTEM RESIDENT ROUTINES

The procedure for placing loaded modules into the RAM or link pack area, and building the required parameter library lists, varies with the type of system, MFT, MVT, VS1, or VS2. This information can be found in the Resident Routine Options section in the following SRLs:

MFT Guide (GC27-6939)
MVT Guide (GC28-6720)
VS1 Planning and Use Guide (GC24-5090)
VS2 Planning and Use Guide (GC28-0600)

SYSTEM DEFINITION STAGE 1 OUTPUT WARNINGS

The following machine listing is an output example from Stage 1 of IMS/360 system definition. This listing informs the IMS/360 system user of actions which must be performed prior to IMS/360 system execution.

SYSTEM DEFINITION - ALL FUNCTIONS

- Batch and Teleprocessing
- 1030 Line Groups
- LTERM Edit
- Terminal Conversation Feature (Disk SPAs)
- Cataloged Procedures

```
*,*******************************************************
*,*******************************************************
*,**                                                   **
*,**       SUCCESSFUL IMS/360 SYSTEM DEFINITION        **
*,**               FOR ALL FUNCTIONS.                  **
*,
*,     --- BEFORE STAGE II EXECUTION ---
*,
*,       THE FOLLOWING IMS/360 DATA SETS MUST BE ALLO-
*,  CATED AND CATALOGED BEFORE ATTEMPTING TO EXECUTE
*,  THE GENERATED STAGE II JOB STREAM:
*,       IMS2.LOAD   IMS2.GENLIB  IMS2.RESLIB
*,       IMS2.MACLIB
*,       IMS2.OBJDSET  USER.PDS.LIBRARY
*,
*,       OTHER REQUIRED DATA SETS MUST BE ALLOCATED
*,  AND CATALOGED PRIOR TO EXECUTION OF THE IMS/360
*,  CONTROL PROGRAM.
*,
*,       SPECIFIED USER SUPPLIED MODULES MUST RESIDE
*,  IN USER.PDS.LIBRARY BEFORE EXECUTION OF THE
*,  CONTROL BLOCK LINK EDIT STEP OF STAGE II.
*,
*,       1030 LINE GROUPS REQUIRE LOAD MEMBER NAMED
*,  (DFS10300) AND MUST RESIDE IN USER.PDS.LIBRARY
*,  BEFORE EXECUTION OF THE CONTROL BLOCK LINK EDIT
*,  STEP OF STAGE II.
*,
*,       REQUESTED LTERM EDIT FUNCTION REQUIRES LOAD
*,  MEMBER NAMED (DFSCNTEO) RESIDE IN USER.PDS.LIBRARY
*,  BEFORE EXECUTION OF THE CONTROL BLOCK LINK EDIT
*,  STEP OF STAGE II.
*,
*,
*,      ---- BEFORE SYSTEM EXECUTION ----
*,
*,       THE FOLLOWING FUNCTIONS MUST BE COMPLETED
*,  BEFORE SUCCESSFUL EXECUTION OF THE IMS/360 CONTROL
*,  PROGRAM CAN BE ACHIEVED:
*,
*,     THE REQUIRED TYPE 4 SVC AND APPENDAGE MODULES
*,     MUST BE PLACED IN SYS1.SVCLIB (MFT,MVT AND VS1) OR SYS1.LPALIB(VS2).
*,
*,     THE RESIDENT SVC INTERFACE MODULES MUST BE LINK
*,     EDITED WITH THE OS/360 NUCLEUS. THESE MODULES ARE
*,     PLACED IN IMS2.RESLIB BY THE STAGE II JOB STREAM,
*,     AND ARE NAMED IGC253 AND IGC254.
*,
*,     THE LOAD MEMBER DFSRRC00 MUST BE ADDED TO THE
*,     'PRIVILIGED PROGRAM NAME LIST' MEMBER (IEFVLNKT)
*,     IN SYS1.LINKLIB.
*,
*,     THE IMS2.RESLIB DATASET MUST BE CONCATENATED WITH
*,     SYS1.LINKLIB BY ADDING THE NAME 'IMS2.RESLIB' TO
*,     THE LNKLST00 MEMBER OF SYS1.PARMLIB.

*,
*,     THE TERMINAL CONVERSATION FEATURE REQUIRES THE
*,     IMS2.SPA DATA SET BE ALLOCATED AND CATALOGED,
*,     WITH SPACE SPECIFICATION - SPACE=(150,22).
*,
*,          REFER TO THE IMS/360 SYSTEM PROGRAMMING
*,     REFERENCE MANUAL FOR INSTRUCTIONS AND ASSIST-
*,     ANCE IN PERFORMING THESE AND OTHER REQUIRED
*,     FUNCTIONS BEFORE SYSTEM EXECUTION.
*,**                                                   **
*,**                                                   **
*,*******************************************************
*,*******************************************************
```

• Terminal Conversation Feature (Disk SPAs)

```
*,*****************************************************
*,*****************************************************
*,**                                                 **
*,**       SUCCESSFUL IMS/360 SYSTEM DEFINITION      **
*,**               FOR NUCLEUS FUNCTION.             **
*,
*,      --- BEFORE STAGE II EXECUTION ---
*,
*,         THE FOLLOWING IMS/360 DATA SETS MUST BE ALLO-
*,    CATED AND CATALOGED BEFORE ATTEMPTING TO EXECUTE
*,    THE GENERATED STAGE II JOB STREAM:
*,         IMS2.LOAD    IMS2.GENLIB   IMS2.RESLIB
*,         IMS2.MACLIB IMS2.PROCLIB
*,         IMS2.OBJDSET  USER.PDS.LIBRARY
*,
*,         OTHER REQUIRED DATA SETS MUST BE ALLOCATED
*,    AND CATALOGED PRIOR TO EXECUTION OF THE IMS/360
*,    CONTROL PROGRAM.
*,
*,         SPECIFIED USER SUPPLIED MODULES MUST RESIDE
*,    IN USER.PDS.LIBRARY BEFORE EXECUTION OF THE
*,    CONTROL BLOCK LINK EDIT STEP OF STAGE II.
*,
*,
*,         ----  BEFORE SYSTEM EXECUTION  ----
*,
*,         THE FOLLOWING FUNCTIONS MUST BE COMPLETED
*,    BEFORE SUCCESSFUL EXECUTION OF THE IMS/360 CONTROL
*,    PROGRAM CAN BE ACHIEVED:
*,
*,       NUCLEUS GENERATION ASSUMES SPECIFIED OS/360 SVC,
*,       SYS1.LINKLIB, AND SYS1.PARMLIB INTERFACES ARE
*,       CONSISTENT WITH SPECIFIED VALUES.
*,
*,       THE TERMINAL CONVERSATION FEATURE REQUIRES THE
*,       IMS2.SPA DATA SET BE ALLOCATED AND CATALOGED,
*,       WITH SPACE SPECIFICATION - SPACE=(200,39).
*,
*,       THE CATALOGED PROCEDURE 'IMS' MUST BE UPDATED
*,       TO INCLUDE DD STATEMENTS FOR ON-LINE DATABASES
*,       AND PLACED IN SYS1.PROCLIB BEFORE THIS PROCEDURE
*,       CAN BE SUCCESSFULLY EXECUTED.
*,
*,       THE CATALOGED PROCEDURE 'IMSRDR' MUST BE PLACED
*,       IN SYS1.PROCLIB BEFORE ANY OTHER IMS GENERATED
*,       PROCEDURE CAN BE SUCCESSFULLY EXECUTED.
*,
*,             REFER TO THE IMS/360 SYSTEM PROGRAMMING
*,       REFERENCE MANUAL FOR INSTRUCTIONS AND ASSIST-
*,       ANCE IN PERFORMING THESE AND OTHER REQUIRED
*,       FUNCTIONS BEFORE SYSTEM EXECUTION.
*,**                                                 **
*,**                                                 **
*,*****************************************************
*,*****************************************************
```

```
*,*****************************************************
*,*****************************************************
*,**                                                 **
*,**        SUCCESSFUL IMS/360 SYSTEM DEFINITION     **
*,**               FOR ALL FUNCTIONS.                **
*,
*,     ---  BEFORE STAGE II EXECUTION  ---
*,
*,       THE FOLLOWING IMS/360 DATA SETS MUST BE ALLO-
*,  CATED AND CATALOGED BEFORE ATTEMPTING TO EXECUTE
*,  THE GENERATED STAGE II JOB STREAM:
*,       IMS2.LOAD   IMS2.GENLIB  IMS2.RESLIB
*,       IMS2.MACLIB
*,       IMS2.OBJDSET  USER.PDS.LIBRARY
*,
*,       OTHER REQUIRED DATA SETS MUST BE ALLOCATED
*,  AND CATALOGED PRIOR TO EXECUTION OF THE IMS/360
*,  CONTROL PROGRAM.
*,
*,       SPECIFIED USER SUPPLIED MODULES MUST RESIDE
*,  IN USER.PDS.LIBRARY BEFORE EXECUTION OF THE
*,  CONTROL BLOCK LINK EDIT STEP OF STAGE II.
*,
*,       1030 LINE GROUPS REQUIRE LOAD MEMBER NAMED
*,  (DFS10300) AND MUST RESIDE IN USER.PDS.LIBRARY
*,  BEFORE EXECUTION OF THE CONTROL BLOCK LINK EDIT
*,  STEP OF STAGE II.
*,
*,       REQUESTED LTERM EDIT FUNCTION REQUIRES LOAD
*,  MEMBER NAMED (DFSCNTE0) RESIDE IN USER.PDS.LIBRARY
*,  BEFORE EXECUTION OF THE CONTROL BLOCK LINK EDIT
*,  STEP OF STAGE II.
*,
*,
*,       ----  BEFORE SYSTEM EXECUTION  ----
*,
*,       THE FOLLOWING FUNCTIONS MUST BE COMPLETED
*,  BEFORE SUCCESSFUL EXECUTION OF THE IMS/360 CONTROL
*,  PROGRAM CAN BE ACHIEVED:
*,
*,     THE REQUIRED TYPE 4 SVC AND APPENDAGE MODULES
*,     MUST BE PLACED IN SYS1.SVCLIB (MFT,MVT AND VS1) OR SYS1.LPALIB(VS2).
*,
*,     THE RESIDENT SVC INTERFACE MODULES MUST BE LINK
*,     EDITED WITH THE OS/360 NUCLEUS. THESE MODULES ARE
*,     PLACED IN IMS2.RESLIB BY THE STAGE II JOB STREAM,
*,     AND ARE NAMED IGC253 AND IGC254.
*,
*,     THE LOAD MEMBER DFSRRC00 MUST BE ADDED TO THE
*,     'PRIVILEGED PROGRAM NAME LIST' MEMBER (IEEVLNKT)
*,     IN SYS1.LINKLIB.
*,
*,     THE IMS2.RESLIB DATASET MUST BE CONCATENATED WITH
*,     SYS1.LINKLIB BY ADDING THE NAME 'IMS2.RESLIB' TO
*,     THE LNKLST00 MEMBER OF SYS1.PARMLIB.
*,
*,          REFER TO THE IMS/360 SYSTEM PROGRAMMING
*,     REFERENCE MANUAL FOR INSTRUCTIONS AND ASSIST-
*,     ANCE IN PERFORMING THESE AND OTHER REQUIRED
*,     FUNCTIONS BEFORE SYSTEM EXECUTION.
*,**                                                 **
*,**                                                 **
*,*****************************************************
*,*****************************************************
```

```
*,*******************************************************
*,*******************************************************
*,**                                                  **
*,**       SUCCESSFUL IMS/360 SYSTEM DEFINITION       **
*,**               FOR CTLBLKS FUNCTION.              **
*,
*,     ---  BEFORF STAGE II EXECUTION  ---
*,
*,        THE FOLLOWING IMS/360 DATA SETS MUST BE ALLO-
*,   CATED AND CATALOGED BEFORE ATTEMPTING TO EXECUTE
*,   THE GENERATED STAGE II JOB STREAM:
*,        IMS2.LOAD    IMS2.GENLIB   IMS2.RESLIB
*,        IMS2.OBJDSET  USER.PDS.LIBRARY
*,
*,        OTHER REQUIRED DATA SETS MUST BE ALLOCATED
*,   AND CATALOGED PRIOR TO EXECUTION OF THE IMS/360
*,   CONTROL PROGRAM.
*,
*,        SPECIFIED USER SUPPLIED MODULES MUST RESIDE
*,   IN USER.PDS.LIBRARY BEFORE EXECUTION OF THE
*,   CONTROL BLOCK LINK EDIT STEP OF STAGE II.
*,
*,
*,        ----  BEFORE SYSTEM EXECUTION  ----
*,
*,        THE FOLLOWING FUNCTIONS MUST BE COMPLETED
*,   BEFORE SUCCESSFUL EXECUTION OF THE IMS/360 CONTROL
*,   PROGRAM CAN BE ACHIEVED:
*,
*,      CTLBLKS GENERATION ASSUMES SPECIFIED OS/360 SVC,
*,      SYS1.LINKLIB, AND SYS1.PARMLIB INTERFACES ARE
*,      CONSISTENT WITH SPECIFIED VALUES.
*,
*,          REFER TO THE IMS/360 SYSTEM PROGRAMMING
*,      REFERENCE MANUAL FOR INSTRUCTIONS AND ASSIST-
*,      ANCE IN PERFORMING THESE AND OTHER REQUIRED
*,      FUNCTIONS BEFORE SYSTEM EXECUTION.
*,**                                                  **
*,**                                                  **
*,*******************************************************
*,*******************************************************
```

```
*,*****************************************************
*,*****************************************************
*,**                                                 **
*,**      SUCCESSFUL IMS/360 SYSTEM DEFINITION       **
*,**               FOR BATCH FUNCTION.               **
*,
*,    ---  BEFORE STAGE II EXECUTION  ---
*,
*,       THE FOLLOWING IMS/360 DATA SETS MUST BE ALLO-
*,  CATED AND CATALOGED BEFORE ATTEMPTING TO EXECUTE
*,  THE GENERATED STAGE II JOB STREAM:
*,       IMS2.LOAD   IMS2.GENLIB  IMS2.RESLIB
*,       IMS2.MACLIB
*,       IMS2.OBJDSET  USER.PDS.LIBRARY
*,
*,       OTHER REQUIRED DATA SETS MUST BE ALLOCATED
*,  AND CATALOGED PRIOR TO EXECUTION OF THE IMS/360
*,  CONTROL PROGRAM.
*,
*,
*,       ----  BEFORE SYSTEM EXECUTION  ----
*,
*,       THE FOLLOWING FUNCTIONS MUST BE COMPLETED
*,  BEFORE SUCCESSFUL EXECUTION OF THE IMS/360 CONTROL
*,  PROGRAM CAN BE ACHIEVED:
*,
*,     THE REQUIRED TYPE 4 SVC AND APPENDAGE MODULES
*,     MUST BE PLACED IN SYS1.SVCLIB (MFT,MVT AND VS1) OR SYS1.LPALIB(VS2).
*,
*,     THE LOAD MEMBER DFSRRC00 MUST BE ADDED TO THE
*,     'PRIVILEGED PROGRAM NAME LIST' MEMBER (IEEVLNKT)
*,     IN SYS1.LINKLIB.
*,
*,     THE IMS2.RESLIB DATASET MUST BE CONCATENATED WITH
*,     SYS1.LINKLIB BY ADDING THE NAME 'IMS2.RESLIB' TO
*,     THE LNKLST00 MEMBER OF SYS1.PARMLIB.
*,
*,           REFER TO THE IMS/360 SYSTEM PROGRAMMING
*,     REFERENCE MANUAL FOR INSTRUCTIONS AND ASSIST-
*,     ANCE IN PERFORMING THESE AND OTHER REQUIRED
*,     FUNCTIONS BEFORE SYSTEM EXECUTION.
*,**                                                 **
*,**                                                 **
*,*****************************************************
*,*****************************************************
```

SYSTEM PROCEDURES

If PROCLIB=YES is specified on the IMSGEN macro-instruction as is suggested in the Stage I input of IMS/360 system definition, certain procedures are created and placed in IMS2.PROCLIB. The created procedures should be examined carefully to determine if the desired JCL has been correctly generated. These procedures may not apply to all applications, but can be used as guidelines for user generated account oriented procedures. If an online IMS/360 system has been defined, particular attention should be devoted to the terminal device allocation generated within the IMS procedure. A list of terminal addresses and logical and physical terminals is printed by Stage I of IMS/360 system definition which should be helpful in checking for correct terminal allocation. Dependent upon the type of system being defined, the procedures which may be created are as follows:

PROCEDURE LIBRARY

Member Name	Description
ACBGEN	A one-step execution procedure for ACBLIB maintenance
DBBBATCH	A one-step execution procedure for a stand-alone Data Language/I batch processing region using IMS2.ACBLIB
DBDGEN	A two-step assemble and link edit procedure to produce data base definition blocks (DBDs)
DLIBATCH	A one-step execution procedure for a stand-alone Data Language/I batch processing region using PSB and DBD libraries
FMTINIT	A six-step job used to initialize the Format Utility data bases, setup the required PSBs and DBDs, and load the default blocks
FMTUTL	A four-step execution procedure for defining message and format descriptions to the Format Utility program
FMTSRVC	A one-step execution procedure for maintaining the message format data bases
IMS	Execution of IMS/360 online control region
IMSBATCH	Execution of IMS/360 online batch message processing region
IMSCOBGO	A three-step compile, link edit, and go procedure combining the procedure IMSCOBOL with an execution step for a stand-alone Data Language/I batch processing region
IMSCOBOL	A two-step compile and link edit procedure for IMS/360 applications written in COBOL

IMSMSG Execution of IMS/360 message processing region

Member Name	Description
IMSPLI	A two-step compile and link edit procedure for IMS/360 applications written in PL/I
IMSPLIGO	A three-step compile, link edit, and go procedure combining the procedure IMSPLI with an execution step for a stand-alone Data Language/I batch processing region
IMSRDR	DASD reader procedure to read IMSMSG procedure into the operating system job stream from direct access devices
IMSWTnnn	These are procedures to print datasets created by the SPOOL SYSOUT options.
MFDBDUMP	This is a procedure to dump the sample problem data base onto a SYSOUT data set.
MFDBLOAD	A Data Language/I batch execution procedure used to load the sample problem data base. Input data for the data base procedure is contained in the MFDFSYSN member of IMS2.MACLIB if the user specified MACLIB=ALL on the IMSGEN statement of his system definition. If MACLIB=ALL was _not_ specified, Stage 1 of IMS/360 system definition did _not_ produce the necessary IEBCOPY control cards to place member MFDFSYSN into MACLIB, and this procedure is not usable as generated.
PSBGEN	A two-step assemble and link edit procedure to produce program specification blocks (PSBs)
SECURITY	A three-step execution, assembly, and link edit procedure for terminal and password security which invokes the security maintenance program
IQFUT	This is a procedure for executing the Interactive Query Facility (IQF) Utility system. An EXEC statement to invoke the procedure is included in the Stage II OS job stream by the IQF module DMGSI1 (Part 1 of IQF Stage I). After system definition, this procedure is contained in IMS2.PROCLIB.
IQFFC	This procedure causes execution of the IQF System Data Base (Field File) Creation Utility program during the Stage II OS job stream created by IQF Stage I. An EXEC statement to invoke the procedure is included in the job stream by the DMGSI1 module. After system definition, this procedure is contained in the IMS2.PROCLIB.

IQFIU — This procedure causes execution of the IQF Index Creation/Update Utility program during the Stage II OS job stream created by Stage I. An EXEC statement to invoke the procedure is included in the job stream by the IQF DMGSI2 module (Part 2 of IQF Stage I). After system definition, this procedure is contained in IMS2.PROCLIB.

In addition to the procedures placed in IMS2.PROCLIB, the following Data Language/I interfaces are also generated:

CBLTDLI — Control cards necessary to establish a COBOL to Dl/I interface

PLITDLI — Control cards necessary to establish a PL/I to DL/I interface

Note that the generated procedures accommodate either MVT, MFT, VS1, or VS2 configurations of the operating system.

All procedures should be placed into IMS2.PROCLIB except for the IMS and IMSRDR procedures. These two procedures should be placed into SYS1.PROCLIB.

EXECUTING JOBS USING PROCEDURES FROM IMS2.PROCLIB

The operating system reader/interpreter requires that the reader procedure used to enter jobs into the operating system job stream contain allocation to the procedure library containing the procedures used by those jobs. The procedure library is defined on the reader procedure's IEFPDSI DD statement. IMS/360 system definition provides a reader procedure called IMSRDR which satisfies these requirements. This procedure is used as generated to start message regions for the online System. If entered from the operating system operator's console using the start command (that is, S IMSRDR), it will cause a message processing region to be started; however, if S IMSRDR, DDD (where DDD is the device address of the card reader) is entered, it will read jobs into the operating system job stream from that card reader, allowing those jobs to use procedures from the IMS2.PROCLIB data set. S IMSRDR, DDD will not work unless a DCB BLKSIZE is included with the start command

S IMSRDR,DDD,DCB=BLKSIZE=80

<u>MEMBER</u> <u>NAME</u> <u>ACBGEN</u>

```
//          PROC    SOUT=A,COMP=,RGN=100K
//G         EXEC    PGM=DFSRRC00,PARM='UPB,&COMP',REGION=&RGN
//SYSPRINT DD       SYSOUT=&SOUT
//IMS       DD      DSN=IMS2.PSBLIB,DISP=SHR
//          DD      DSN=IMS2.DBDLIB,DISP=SHR
//IMSACB    DD      DSN=IMS2.ACBLIB,DISP=OLD
//SYSUT3    DD      UNIT=SYSDA,SPACE=(80,(100,100))
//SYSUT4    DD      UNIT=SYSDA,SPACE=(256,(100,100)),DCB=KEYLEN=8
//COMPCTL   DD      DSN=IMS2.PROCLIB(DFSACBCP),DISP=SHR
```

EXEC CARD PARAMETERS FOR THE ACBGEN PROCEDURE

SOUT=
is the SYSOUT class - Default 'A'.

COMP=
'PRECOMP,POSTCOMP' in any combination to cause the required inplace compression. Default is none.

RGN=
the region size for this execution. Default is 100K.

MEMBER NAME DBBBATCH

```
//           PROC     MBR=TEMPNAME,SOUT=A,PSB=,BUF=8,SPIE=0,TEST=0
//G          EXEC     PGM=DFSRRC00,REGION=130K,
//                    PARM='DBB,&MBR,&PSB,&BUF,&SPIE&TEST'
//STEPLIB    DD       DSN=IMS2.RESLIB,DISP=SHR
//           DD       DSN=IMS2.PGMLIB,DISP=SHR
//IMSACB     DD       DSN=IMS2.ACBLIB,DISP=SHR
//IEFRDER    DD       DSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),UNIT=(2400,,DEFER),
//                    DCB=(RECFM=VBS,BLKSIZE=1408,LRECL=1400,BUFNO=1)
//SYSUDUMP   DD       SYSOUT=&SOUT,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),
//                    SPACE=(605,(500,500),RLSE,,ROUND)
```

Assumes

1. User adds DD statements for data sets representing data bases.

Note: Symbolic parameters are the same as for DLIBATCH.

MEMBER NAME DBDGEN

```
//           PROC     MBR=TEMPNAME,SOUT=A
//C          EXEC     PGM=IEUASM,REGION=120K,PARM='LOAD,NODECK'
//SYSLIB     DD       DSN=IMS2.MACLIB,DISP=SHR
//SYSGO      DD       UNIT=SYSDA,DISP=(,PASS),SPACE=(80,(100,100),RLSE),
//                    DCB=(BLKSIZE=400,RECFM=FB,LRECL=80)
//SYSPRINT   DD       SYSOUT=&SOUT,DCB=(LRECL=121,RECFM=FBM,BLKSIZE=605),
//                    SPACE=(121,(500,500),RLSE,,ROUND)
//SYSUT1     DD       UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(100,50))
//SYSUT2     DD       UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(100,50))
//SYSUT3     DD       UNIT=(SYSDA,SEP=(SYSLIB,SYSUT1,SYSUT2)),
//                    SPACE=(1700,(100,50))
//L          EXEC     PGM=DFSILNK0,PARM='XREF,LIST',COND=(0,LT,C),REGION=120K
//STEPLIB    DD       DSN=IMS2.RESLIB,DISP=SHR
//SYSLIN     DD       DSN=*.C.SYSGO,DISP=(OLD,DELETE)
//SYSPRINT   DD       SYSOUT=&SOUT,DCB=(LRECL=121,RECFM=FBA,BLKSIZE=605),
//                    SPACE=(121,(100,100),RLSE)
//SYSLMOD    DD       DSN=IMS2.DBDLIB(&MBR),DISP=SHR
//SYSUT1     DD       UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),DISP=(,DELETE),
//                    SPACE=(1024,(100,10),RLSE)
```

MEMBER NAME DLIBATCH

```
//          PROC     MBR=TEMPNAME,SOUT=A,PSB=,BUF=8,SPIE=0,TEST=0
//G         EXEC     PGM=DFSRRC00,REGION=130K,
//                   PARM='DLI,&MBR,&PSB,&BUF,&SPIE&TEST'
//STEPLIB   DD       DSN=IMS2.RESLIB,DISP=SHR
//          DD       DSN=IMS2.PGMLIB,DISP=SHR
//IMS       DD       DSN=IMS2.PSBLIB,DISP=SHR
//          DD       DSN=IMS2.DBDLIB,DISP=SHR
//IEFRDER   DD       DSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),UNIT=(2400,DEFER),
//   DCB=(RECFM=VBS,BLKSIZE=1408,RECL=1400,BUFNO=1)
//SYSUDUMP  DD       SYSOUT=&SOUT,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),
//   SPACE=(605,(500,500),RLSE,,ROUND)
```

Assumes

1. User adds DD statements for data sets representing DL/I data bases.

EXEC CARD PARAMETERS FOR THE DLIBATCH PROCEDURE

MBR=

is an application program name.

PSB=

is an optional parameter specifying a PSB name when the PSB name and application program name are different.

BUF=

specifies the data base buffer size. If not present, the default size specified at system definition will be used. Buffer size is specified in 1K multiples. Values may range from 1 through 999.

SPIE=

SPIE option:

0= allow user SPIE, if any, to remain in effect while processing the application program call.

1= negate the user's SPIE while processing the application program call. Negated SPIEs are reinstated before returning to the application program.

TEST=

validity check option

0= no validity checking
1= validity check the addresses in the user's call list.

MEMBER NAME FMTINIT

```
//IMSFMT   JOB      1,IMS,MSGLEVEL=1
//STEP1    EXEC     PGM=IEWL,REGION=130K,
//                  PARM='XREF,LIST,LET,NCAL,DCBS'
//SYSPRINT DD       SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),
//                  SPACE=(605,(10,10),RLSE,,ROUND)
//LOAD     DD       DSN=IMS2.LOAD,DISP=SHR
//SYSLMOD  DD       DSN=IMS2.DBDLIB,DISP=OLD
//SYSUT1   DD       UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSLIN   DD       *
   INCLUDE  LOAD(DFSUTS10)
   NAME DFSUTS10(R)
   INCLUDE  LOAD(DFSUTS20)
   NAME DFSUTS20(R)
//STEP2    EXEC     PGM=IEWL,REGION=130K,
//                  PARM='XREF,LIST,LET,NCAL,DCBS'
//SYSPRINT DD       SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),
//                  SPACE=(605,(10,10),RLSE,,ROUND)
//LOAD     DD       DSN=IMS2.LOAD,DISP=SHR
//SYSLMOD  DD       DSN=IMS2.DBDLIB,DISP=OLD
//SYSUT1   DD       UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSLIN   DD       *
   INCLUDE  LOAD(DFSUTS30)
   NAME DFSUTS30(R)
   INCLUDE  LOAD(DFSUTS40)
   NAME DFSUTS40(R)
   INCLUDE  LOAD(DFSUTS50)
   NAME DFSUTS50(R)
   INCLUDE  LOAD(DFSUTS60)
   NAME DFSUTS60(R)
//STEP3    EXEC     PGM=IEBCOPY
//SYSPRINT DD       SYSOUT=A
//SYSUT1   DD       DSN=IMS2.LOAD,DISP=SHR
//SYSUT2   DD       DSN=IMS2.REFERAL,DISP=SHR
//SYSUT3   DD       UNIT=SYSDA,SPACE=(80,(10,20))
//SYSUT4   DD       UNIT=SYSDA,SPACE=(256,(2,4))
//SYSIN    DD       *
   COPY OUTDD=SYSUT2,INDD=((SYSUT1,R))
   SELECT MEMBER=DFSMI1
   SELECT MEMBER=DFSMI2
   SELECT MEMBER=DFSMI4
   SELECT MEMBER=DFSMO1
   SELECT MEMBER=DFSMO2
   SELECT MEMBER=DFSMO3
   SELECT MEMBER=DFSMO4
   SELECT MEMBER=DFSDF1
   SELECT MEMBER=DFSDF2
   SELECT MEMBER=DFSDF4
//STEP4    EXEC     PGM=DFSRRC00,REGION=200K,
//                  PARM='DLI,DFSUTS00,DFSUTS10,4,00'
//IMS      DD       DSN=IMS2.PSBLIB,DISP=SHR
//         DD       DSN=IMS2.DBDLIB,DISP=SHR
//MSGINDX  DD       DSN=IMS2.MSGINDX,DISP=OLD
//MSGINDXO DD       DSN=IMS2.MSGINDXO,DISP=OLD
//MSGDB    DD       DSN=IMS2.MESSAGES,DISP=OLD
//FMTINDX  DD       DSN=IMS2.FMTINDX,DISP=OLD
//FMTINDXO DD       DSN=IMS2.FMTINDXO,DISP=OLD
//FMTDB    DD       DSN=IMS2.FORMATS,DISP=OLD
//SYSPRINT DD       SYSOUT=A,DCB=BLKSIZE=80
//SYSIN    DD       DSN=IMS2.GENLIB(DFSUTS70),DISP=SHR
//STEP5    EXEC     PGM=DFSRRC00,REGION=250K,
//                  PARM='DLI,DFSUNU20,DFSUTS20,4,00'
//IMS      DD       DSN=IMS2.PSBLIB,DISP=SHR
```

```
//           DD       DSN=IMS2.DBDLIB,DISP=SHR
//IEFRDER    DD       DSN=&&IMSLOG,DISP=(,PASS),
//                    UNIT=SYSDA,SPACE=(CYL,(1,1)),DCB=DSORG=PS
//DFSLOGTT DD         DSN=IMS2.LOGT,DISP=SHR
//SYSUDUMP DD         SYSOUT=A
//SYSPRINT DD         SYSOUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
//UTPRINT    DD       SYSOUT=A
//SYSTEXT    DD       DSN=IMS2.GENLIB(DFSUTS90),DISP=SHR
//REFERAL    DD       DSN=IMS2.REFERAL,DISP=SHR
//MSGINDX    DD       DSN=IMS2.MSGINDX,DISP=OLD
//MSGINDXO DD         DSN=IMS2.MSGINDXO,DISP=OLD
//MSGDB      DD       DSN=IMS2.MESSAGES,DISP=OLD
//FMTINDX    DD       DSN=IMS2.FMTINDX,DISP=OLD
//FMTINDXO DD         DSN=IMS2.FMTINDXO,DISP=OLD
//FMTDB      DD       DSN=IMS2.FORMATS,DISP=OLD
//SEQBLKS    DD       DSN=&&BLKS,DISP=(NEW,PASS),
//                    UNIT=SYSDA,SPACE=(CYL,(1,1))
//STEP6      EXEC     PGM=DFSUNU30,REGION=200K
//SEQBLKS    DD       DSN=&&BLKS,DISP=(OLD,DELETE)
//SYSPRINT DD         SYSOUT=A
//UTPRINT    DD       SYSOUT=A
//SYSUDUMP DD         SYSOUT=A
//FORMAT     DD       DSN=IMS2.FORMAT,DISP=OLD
//DUMMY      DD       DUMMY,DCB=BLKSIZE=80
```

Note: STEP4 PROGRAM,DFSUTS00, may create a S03B abend as part of its normal operation. To ensure transparency to the user, no SYSUDUMP or SYSABEND DD card has been supplied for this step of the procedure. If the user elects to supply a dump DD card for this step, the S03B abend should be ignored. For further information, refer to the description of MODULE DFSUTS00 in the _IMS/360 SYSTEM MANUAL VOLUME I_.

MEMBER NAME FMTUTL

```
//            PROC     SNODE=IMS2,SOR=NOLIB,MBR=NOMBR,MNODE=IMS2,MAC=GENLIB,
//                     RGN=250K,BUF=4,SPIE=0,TEST=0,SOUT=A
//PREPROC   EXC        PGM=IEUASM,PARM='NOLOAD,DECK',REGION=100K
//SYSPRINT DD          SYSOUT=&SOUT
//SYSPUNCH DD          DSN=&&ITBPASS,DISP=(NEW,PASS),SPACE=(CYL,(1,1)),
//                     UNIT=SYSDA
//SYSLIB    DD         DSN=&MNODE..&MAC,DISP=SHR
//          DD         DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1    DD         UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT2    DD         UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT3    DD         UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN     DD         DSN=&SNODE..&SOR(&MBR),DISP=SHR
//PHASE1    EXEC       PGM=DFSUNU10,COND=(8,LE,PREPROC),
//                     PARM='COMPRESS',REGION=&RGN
//SYSPRINT DD          SYSOUT=&SOUT
//SYSUDUMP DD          SYSOUT=&SOUT
//UTPRINT   DD         SYSOUT=&SOUT
//SYSLIN    DD         UNIT=SYSDA,SPACE=(CYL,(1,1)),DCB=BLKSIZE=80
//SYSLMOD   DD         DSN=IMS2.REFERAL,DISP=OLD
//DUMMY     DD         DUMMY,DCB=BLKSIZE=80
//SYSUT1    DD         UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSTEXT   DD         DSN=&&TXTPASS,DISP=(NEW,PASS),UNIT=SYSDA,
//                     SPACE=(CYL,(1,1)),DCB=BLKSIZE=80
//SYSIN     DD         DSN=&&ITBPASS,DISP=(OLD,DELETE)
//PHASE3    EXEC       PGM=DFSRRC00,COND=(4,LE,PHASE1),REGION=&RGN,
//                     PARM='DLI,DFSUNU20,DFSUTS20,&BUF,&SPIE&TEST'
//IMS       DD         DSN=IMS2.PSBLIB,DISP=SHR
//          DD         DSN=IMS2.DBDLIB,DISP=SHR
//IEFRDER   DD         DSN=&&IMSLOG,DISP=(,PASS),UNIT=SYSDA,
//                     SPACE=(CYL,(1,1)),DCB=DSORG=PS
//DFSLOGTT DD          DSN=IMS2.LOGT,DISP=SHR
//SYSUDUMP DD          SYSOUT=&SOUT
//SYSPRINT DD          SYSOUT=&SOUT,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
//UTPRINT   DD         SYSOUT=&SOUT
//SYSTEXT   DD         DSN=&&TEXTPASS,DISP=(OLD,DELETE)
//REFERAL   DD         DSN=IMS2.REFERAL,DISP=SHR
//SEQBLKS   DD         DSN=&&BLKS,DISP=(NEW,PASS),UNIT=SYSDA,
//                     SPACE=(CYL,(2,2))
//FMTINDX   DD         DSN=IMS2.FMTINDX,DISP=OLD
//FMTINDXO DD          DSN=IMS2.FMTINDXO,DISP=OLD
//FMTDB     DD         DSN=IMS2.FORMATS,DISP=OLD
//MSGINDX   DD         DSN=IMS2.MSGINDX,DISP=OLD
//MSGINDXO DD          DSN=IMS2.MSGINDXO,SIDP=OLD
//MSGDB     DD         DSN=IMS2.MESSAGES,DISP=OLD
//PHASE4    EXEC       PGM=DFSUNU30,REGION=&RGN,PARM='COMPRESS'
//SEQBLKS   DD         DSN=&&BLKS,DISP=(OLD,DELETE)
//SYSPRINT DD          SYSOUT=&SOUT
//UTPRINT   DD         SYSOUT-&SOUT
//FORMAT    DD         DSN=IMS2.FORMAT,DISP=OLD
//SYSUDUMP DD          SYSOUT=&SOUT
//DUMMY     DD         DUMMY,DCB=BLKSIZE=80
```

EXEC CARD PARAMETERS FOR THE FMTUTL PROCEDURE

SNODE=

is the NODE of the PREPROC step SYSIN data set name

SOR=

is the remainder of the qualified name of the PREPROC step SYSIN data set name

MBR= is the member name of the PREPROC step SYSIN partitioned data set to be used as input

MNODE=

is the NODE of the PREPROC step SYSLIB data set

MAC=

is the remainder of the qualified name of the PREPROC step SYSLIB data set

RGN=

is the value to be specified in the REGION= parameter of the EXEC cards for the PHASE1 and PHASE3 steps

BUF=

specifies the data base buffer size. Buffer size is specified in 1K multiples. Values may range form 1 through 999.

SPIE=

0= allow user SPIE, if any, to remain in effect while processing the program call.

1= negate user SPIE

TEST=

validity check option

0= no validity checking

1= validity check the addresses in the user's call list

SOUT=

is class assigned for SYSOUT DD cards

MEMBER NAME FMTSRVC

```
//           PROC     SOUT=A
//FMTSRVC  EXEC     PGM=DFSRRC00,REGION=250K,
//                  PARM='DLI,DFSUTSA0,DFSUTS20,4,00'
//IMS        DD     DSN=IMS2.PSBLIB,DISP=OLD
//           DD     DSN=IMS2.DBDLIB,DISP=OLD
//IEFRDER    DD     DSN=&&IMSLOG,UNIT=SYSDA,
//                  SPACE=(CYL,(1,1)),DCB=DSORG=PS
//DFSLOGTT DD       UNIT=SYSDA,SPACE=(TRK,(1))
//*
//*    FORMAT DATA BASE (HIDAM)
//*
//FMTINDX  DD       DSN=IMS2.FMTINDX,DISP=OLD
//FMTINDXO DD       DSN=IMS2.FMTINDXO,DISP=OLD
//FMTDB    DD       DSN=IMS2.FORMATS,DISP=OLD
//*
//*    MESSAGE DATA BASE (HIDAM)
//*
//MSGINDX  DD       DSN=IMS2.MSGINDX,DISP=OLD
//MSGINDXO DD       DSN=IMS2.MSGINDXO,DISP=OLD
//MSGDB    DD       DSN=IMS2.MESSAGES,DISP=OLD
//*
//*    PRINT FILES
//*
//SYSPRINT DD       SYSOUT=&SOUT
//*                 DCB=(RECFM=VBA,LRECL=137)
//SYSSNAP  DD       SYSOUT=&SOUT
//*                 DCB=(RECFM=VBA,LRECL=125,BLKSIZE=1632)
//SYSUDUMP DD       SYSOUT=&SOUT
//*
//*    REFERAL LIBRARY
//*
//REFERAL  DD       DSN=IMS2.REFERAL,DISP=OLD
//*
//*    ON-LINE FORMAT LIBRARY
//FORMAT   DD       DSN=IMS2.FORMAT,DISP=OLD
//*
//*
//*    //SYSIN DD * MUST BE SUPPLIED BY
//*    USER WITH INPUT CONTROL CARD STREAM
//*
//*    ALL DISP=OLD SPECIFICATIONS OF THIS
//*    PROCEDURE ARE REQUIRED ...
//*
//*
```

EXEC CARD PARAMETERS FOR FMTSRVC PROCEDURE

SOUT=

is class assigned to sysout data sets

MEMBER NAME IMS

```
//           PROC     RGN=300K,DMBP=000,PSBP=000,DBBP=000,TPDP=000,WKAP=000,
//                    S=0,PKEY=1,PTY=254,TEST=1,QCR=000,FBP=000,SOUT=A
//IEFPROC    EXEC     PGM=DFSRRC00,REGION=&RGN,
//                    PARM=(CTL,
//                    &PTY&PKEY.0&TEST&S&QCR.&FBP&PSBP&DMBP&DBBP&TPDP.
//                    000000&WKAP)
//*
//*
//*                   PARM=(AAA,BBBCDEFGGGHHHIIIJJJKKKLLL000000OOO)
//*
//*                   AAA         REGION TYPE
//*                   BBB         DISPATCHING PRIORITY
//*                   C           REGION PROTECT KEY(0=0; 1=NON ZERO)
//*                   E           TEST OPTION
//*                   F           CONTROL PROGRAM SUFFIX
//*                   GGG         NUMBER OF QUEUE BUFFERS
//*                   HHH         FORMAT BUFFER POOL SIZE (IN 1K BLOCKS)
//*                   III         PSB POOL SIZE (IN 1K BLOCKS)
//*                   JJJ         DMB POOL SIZE (IN 1K BLOCKS)
//*                   KKK         DATABASE BUFFER POOL SIZE (IN 1K BLOCKS)
//*                   LLL         LINE BUFFER POOL (IN 1K BLOCKS)
//*                   000000      RESERVED PARM AREA
//*                   OOO         WORKING STORAGE (IN 1K BLOCKS)
//*
//*                   IF SYMBOLIC PARMS ARE NOT SPECIFIED; THE VALUES
//*                   PROVIDED BY SYSTEM DEFINITION ARE USED.
//*
//IEFRDER    DD       DSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),UNIT=(2400,,DEFER),
//                    DCB=(RECFM=VBS,BLKSIZE=1408,LRECL=1400,BUFNO=1)
//IMSLOGR    DD       DSN=IMSLOG,DISP=(OLD,KEEP),VOL=SER=000000,
//                    UNIT=AFF=IEFRDER
//QBLKS      DD       DSN=IMS2.QBLKS,DISP=OLD
//SHMSG      DD       DSN=IMS2.SHMSG,DISP=OLD
//LGMSG      DD       DSN=IMS2.LGMSG,DISP=OLD
//IMSACB     DD       DSN=IMS2.ACBLIB,DISP=SHR
//IMSDILIB   DD       DSN=IMS2.FORMAT,DISP=OLD *
//IMSSPA     DD       DSN=IMS2.SPA,DISP=OLD
//SYSUDUMP   DD       SYSOUT=&SOUT,DCB=(LRECL=125,RECFM=FBA,BLKSIZE=3129),
//                    SPACE=(6050,300,,,ROUND)
//IMSDBL     DD       DSN=IMS2.DBLLOG,DISP=SHR
```

__Note 1__: This procedure can not be entered in the normal OS job stream (that is, via a card reader) unless modified as described in Chapter 6 of the __IMS/360 Operator's Reference Manual__, under "Execution of IMS/360 as a Problem Program".

It is assumed that the IMS/360 modules are in the IMS2.RESLIB data set. The user must add DD statements for the data sets representing data bases.

__Note 2__: In a VS System, the //IEFPROC EXEC card must indicate that real storage is to be used:

*__Note 3__: This DD statement __must__ specify DISP=OLD.

```
//IEFPROC    EXEC     PGM=DFSRRC00,REGION=&RGN,ADDRSPC=REAL,
```

EXEC CARD PARAMETERS FOR THE IMS PROCEDURE

RGN=

size of the OS region to be allocated to the IMS control program. This parameter has no effect in an MFT system.

DMBP=

number of 1K blocks of subpool 0 to be reserved for the DMB pool. (Identified in a core dump as DLDP.)

PSBP=

number of 1K blocks of subpool 0 to be reserved for the PSB pool. (Identified in a core dump as DLMP.)

DBBP=

number of 1K blocks of subpool 0 to be reserved for the data base buffer pool. (Identified in a core dump as DBAS.)

TPDP=

number of 1K blocks of subpool 0 to be reserved for the teleprocessing line buffer pool. (Identified in a core dump as I/OP.)

WKAP=

number of 1K blocks of subpool 0 to be reserved for the control program working area.

S=

control program name suffix. Character to be used as the last character of the eight-character control program name. This allows multiple copies of the IMS nucleus to reside on IMS2.RESLIB.

PKEY=

protect key option. If PKEY=1 is selected, IMS will operate under a non-zero protect key. If PKEY=0 is selected, IMS will run with protect key of 0. PKEY=1 is the recommended option. This parameter is ignored in MFT or VS1 because all system tasks run under protect key 0.

PTY=

dispatching priority. OS/360 dispatching priority at which the IMS control region should operate. This parameter is ignored in MFT.

TEST=

test option. If TEST=1 is selected, all subpool 0 core obtained for the control region will be cleared before use.

QCR=

number of queue buffers in subpool 0 to be allocated to the queue pool. (Identified in a core dump as QBUF.)

FBP=

number of 1K blocks of subpool 0 to be reserved for the message format block pool (MFBP). (This is identified in a main storage dump as MFBP)

SOUT=

class assigned for SYSOUT DD cards.

MEMBER NAME IMSBATCH

```
//          PROC    MBR=TEMPNAME,SOUT=3,OPT=N,SPIE=0,TEST=0,PSB=,IN=,OUT=,
                    DIRCA=000
//G         EXED    PGM=DFSRRC00,REGION=30K,
//                  PARM='BMP,&MBR,&PSB,&IN,&OUT,&OPT&SPIE&TEST&DIRCA
//STEPLIB  DD       DSN=IMS2.RESLIB,DISP=SHR
//         DD       DSN=IMS2.PGMLIB,DISP=SHR
//SYSUDUMP DD       SYSOUT=&SOUT,DCB=(LRECL=121,RECFM=VBA,BLKSIZE=3129),
//                  SPACE=(125,(2500,100),RLSE,,ROUND)
```

EXEC CARD PARAMETERS FOR THE IMSBATCH PROCEDURE

MBR=

is an application program name.

SOUT=

specifies the SYSOUT class for the SYSUDUMP DD card.

PSB=

is an optional parameter specifying a PSB name when the PSB name and application program name are different.

IN=

is an input transaction code. This parameter is necessary only when the application program intends to access the message queues. If this parameter is specified, the OUT= parameter is ignored.

OUT=

is an output transaction code or logical terminal name. It is necessary when the application program desires to send output without accessing the input queues. This parameter is ignored if IN= is also specified.

OPT=

action to be taken if the batch message region starts and no control program is active.
N= ask operator for decision. This is the default.
W= wait for a control program.
C= cancel the batch message region automatically.

SPIE=

SPIE option

0= allow user SPIE, if any, to remain in effect while processing the application program call.

1= negate the user's SPIE while processing the application program call. Negated SPIEs are reinstated before returning to the application program.

SPIE macros issued by the application program are only effective for program checks which occur within the batch message region.

TEST=

validity check option.

0= no validity checking.

1= validity check the addresses in the user's call list.

DIRCA=

number of 1K blocks of subpool 253 to be reserved to hold a copy of the user's PCBs. If this value is not specified, the system will reserve an area which can hold the PCBs for any program in the online system. This parameter applies only to OS/VS systems and must be a three-digit number (for example, 001). To determine the size of the PCBs, see the output of the ACB utility program.

MEMBER NAME IMSCOBGO

```
//           PROC    MBR=,PAGES=60,SOUT=A,PSB=,SPIE=0,TEST=0,BUF=8
//C          EXEC    PGM=IKFCBL00,PARM='SIZE=130K,BUF=10K,LINECNT=50',REGION=150K
//SYSLIN     DD      DSN=&&LIN,DISP=(MOD,PASS),UNIT=SYSDA,
//                   DCB=(LRECL=80,RECFM=FB,BLKSIZE=400),
//                   SPACE=(CYL,(4,1),RLSE)
//SYSPRINT DD        SYSOUT=&SOUT,DCB=(LRECL=121,BLKSIZE=605,RECFM=FBA),
//                   SPACE=(605,(&PAGES.0,&PAGES),RLSE,,ROUND)
//SYSUT1     DD      UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(10,1),RLSE)
//SYSUT2     DD      UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(10,1),RLSE)
//SYSUT3     DD      UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(10,1),RLSE)
//SYSUT4     DD      UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(10,1),RLSE)
//L          EXEC    PGM=DFSILNK0,REGION=120K,PARM='XREF,LET,LIST',
//                   COND=(4,LT,C)
//STEPLIB    DD      DSN=IMS2.RESLIB,DISP=SHR
//SYSLIB     DD      DSN=SYS1.COBLIB,DISP=SHR
//           DD      DSN=SYS1.PL1LIB,DISP=SHR
//RESLIB     DD      DSN=IMS2.RESLIB,DISP=SHR
//SYSLIN     DD      DSN=&&LIN,DISP=(OLD,DELETE),VOL=REF=*.C.SYSLIN
//           DD      DSN=IMS2.PROCLIB(CBLTDLI),DISP=SHR
//           DD      DDNAME=SYSIN
//SYSLMOD    DD      DSN=IMS2.PGMLIB(&MBR),DISP=SHR
//SYSPRINT DD        SYSOUT=&SOUT,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),
//                   SPACE=(605,(&PAGES.0,&PAGES),RLSE,,ROUND)
//SYSUT1     DD      UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),DISP=(,DELETE),
//                   SPACE=(CYL,(10,1),RLSE)
//G          EXEC    PGM=DFSRRC00,REGION=150K,TIME=2,COND=(4,LT),
//                   PARM='DLI,&MBR,&PSB,&BUF,&SPIE&TEST'
//STEPLIB    DD      DSN=IMS2.RESLIB,DISP=SHR
//           DD      DSN=IMS2.PGMLIB,DISP=SHR
//IMS        DD      DSN=IMS2.PSBLIB,DISP=SHR
//           DD      DSN=IMS2.DBDLIB,DISP=SHR
//IEFRDER    DD      DSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),UNIT=(2400,,DEFER),
//                   DCB=(RECFM=VBS,BLKSIZE=1408,LRECL=1400,BUFNO=1)
//SYSOUT     DD      SYSOUT=&SOUT,SPACE=(CYL,(1,1)),DCB=(LRECL=133,RECFM=FA)
//SYSUDUMP DD        SYSOUT=&SOUT,DCB=(LRECL=121,RECFM=FBA,BLKSIZE=3025),
//                   SPACE=(3025,(200,100),RLSE,,ROUND)
```

Assumes

1. User supplies source data from SYSIN
2. Output Class A
3. MBR=NAME, when name is load module name for program
4. SYSDA is generic device name
5. User add DD statements in execute set for data sets representing Data Language/I data bases.
6. Execution time limit of 2 minutes specified.

MEMBER NAME IMSCOBOL

```
//          PROC     MBR=,PAGES=60,SOUT=A
//C         EXEC     PGM=IKFCBL00,PARM='SIZE=130K,BUF=10K,LINECNT=50'REGION=150K
//SYSLIN    DD       DSN=&&LIN,DISP=(MOD,PASS),UNIT=SYSDA,
//                   DCB=(LRECL=80,RECFM=FB,BLKSIZE=400),
//                   SPACE=(CYL,(4,1),RLSE)
//SYSPRINT DD        SYSOUT=&SOUT,DCB=(LRECL=121,BLKSIZE=605,RECFM=FBA),
//                   SPACE=(605,(&PAGES.0,&PAGES),RLSE,,ROUND)
//SYSUT1    DD       UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(10,1),RLSE)
//SYSUT2    DD       UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(10,1),RLSE)
//SYSUT3    DD       UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(10,1),RLSE)
//SYSUT4    DD       UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(10,1),RLSE)
//L         EXEC     PGM=DFSILNK0,REGION=120K,PARM='XREF,LET,LIST',
//                   COND=(4,LT,C)
//STEPLIB   DD       DSN=IMS2.RESLIB,DISP=SHR
//SYSLIB    DD       DSN=SYS1.COBLIB,DISP=SHR
//          DD       DSN=SYS1.PL1LIB,DISP=SHR
//RESLIB    DD       DSN=IMS2.RESLIB,DISP=SHR
//SYSLIN    DD       DSN=&&LIN,DISP=(OLD,DELETE),VOL=REF=*.C.SYSLIN
//          DD       DSN=IMS2.PROCLIB(CBLTDLI),DISP=SHR
//          DD       DDNAME=SYSIN
//SYSLMOD   DD       DSN=IMS2.PGMLIB(&MBR),DISP=SHR
//SYSPRINT DD        SYSOUT=&SOUT,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),
//                   SPACE=(605,(&PAGES.0,&PAGES),RLSE,,ROUND)
//SYSUT1    DD       UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),DISP=(,DELETE),
//                   SPACE=(CYL,(10,1),RLSE)
```

Assumes

1. User supplies source data from SYSIN
2. Output Class A
3. MBR=NAME, when name is load module name for program
4. SYSDA is generic device name
5. RESLIB cataloged.

MEMBER NAME IMSMSG

```
//MESSAGE JOB 1,IMS,MSGLEVEL=1,PRTY=11,CLASS=A,MSGCLASS=3,REGION=30K
//REGION    EXEC     PGM=DFSRRC00,REGION=30K,TIME=1440,
//                   PARM='MSG,001000000000'
//STEPLIB   DD       DSN=IMS2.RESLIB,DISP=SHR
//          DD       DSN=IMS2.PGMLIB,DISP=SHR
//SYSUDUMP DD        SYSOUT=3,DCB=(LRECL=125,BLKSIZE=3219,RECFM=VBA),
//                   SPACE=(125,(2500,100),RLSE,,ROUND)
```

EXEC CARD PARAMETERS FOR THE IMSMSG PROCEDURE

PARM=

'MSG,AAAAAAAAAAAA,BCDEFFGGG'

MSG=

a required positional parameter indicating a message region is to be started.

AAAAAAAAAAAA=

a required positional parameter specifying 4 three-digit decimal numbers indicating which classes of messages will be handled by this message region. That is, if classes 1, 2, and 3 are to be processed by this region, the PARM field would be specified as PARM='MSG,001002003000'

The sequence of specifying the classes determines relative class priority within the message region. In the above example, all Class 1 messages are selected for scheduling before any Class 2 messages would be considered. Class numbers can not be greater than the maximum number of classes specified at system definition time.

B=

action to be taken if the message region starts and no control region is active.
W - wait for control program to start
N - ask operator for decision - this is the default
C - cancel message region automatically

C=

overlay supervisor option.
0 - allow OS to load and delete the overlay supervisor for every overlay application program - this is the default
1 - load and retain a copy of the overlay supervisor at message region initialization time

D=

SPIE option.
0 - allow user SPIE, if any, to remain in effect while processing the application program call
1 - negate the user's SPIE while processing the application program call. Negated SPIEs are reinstated before returning to the application program.

SPIE macros issued by the application program are only effective for program checks which occur within the message region.

E=

validity check option
0 - no address validity checking will be made.
1 - validity check the addresses in the user's call list.

FF=

termination limit option. A decimal number between 1 and 99. When the number of application program ABENDs reaches this limit the message region will be automatically terminated. This will allow OS to print the accumulated SYSOUT data sets. Default value is 1.

GGG=

number of 1K blocks of subpool 253 to be reserved to hold a copy of the user's PCBs. If this value is not specified, the system will reserve an area which can hold the PCBs for any application program in the system. This parameter applies only

to OS/VS systems and must be a three-digit number (for example, 001).
Note: The output from the ACB utility program DFSUACB0 will specify application program PCB sizes.

MEMBER NAME IMSPLI

```
//          PROC    MBR=,PAGES=50,SOUT=A
//C         EXEC    PGM=IEMAA,REGION=114K,
//                  PARM='XREF,ATR,LOAD,NODECK,NOMACRO,,OPT=1'
//SYSUT1    DD      UNIT=SYSDA,SPACE=(1024,(60,60),RLSE,,ROUND),
//                  DCB=BLKSIZE=1024,DISP=(,DELETE)
//SYSUT3    DD      UNIT=SYSDA,SPACE=(1024,(60,60),RLSE,,ROUND),
//                  DCB=BLKSIZE=1024,DISP=(,DELETE)
//SYSPRINT DD       SYSOUT=&SOUT,DCB=(LRECL=125,BLKSIZE=629,RECFM=VBA),
//                  SPACE=(605,(&PAGES.0,&PAGES),RLSE)
//SYSLIN    DD      UNIT=SYSDA,SPACE=(80,(250,80),RLSE),DCB=BLKSIZE=80,
//                  DISP=(,PASS)
//L         EXEC    PGM=DFSILNK0,PARM='XREF,LIST,LET',COND=(4,LT,C),
//                  REGION=120K
//STEPLIB   DD      DSN=IMS2.RESLIB,DISP=SHR
//SYSLIB    DD      DSN=SYS1.PL1LIB,DISP=SHR
//          DD      DSN=SYS1.COBLIB,DISP=SHR
//RESLIB    DD      DSN=IMS2.RESLIB,DISP=SHR
//SYSLIN    DD      DSN=*.C.SYSLIN,DISP=(OLD,DELETE)
//          DD      DSN=IMS2.PROCLIB(PLITDLI),DISP=SHR
//          DD      DDNAME=SYSIN
//SYSLMOD   DD      DSN=IMS2.PGMLIB(&MBR),DISP=SHR
//SYSPRINT DD       SYSOUT=&SOUT,DCB=(LRECL=121,RECFM=FBA,BLKSIZE=605),
//                  SPACE=(605,(&PAGES.0,&PAGES),RLSE)
//SYSUT1    DD      UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(5,1),RLSE)
```

Same assumptions as IMSCOBOL

MEMBER NAME IMSPLIGO

```
//           PROC      MBR=,PAGES=50,SOUT=A,PSB=,SPIE=0,TEST=0,BUF=0
//C          EXEC      PGM=IEMAA,REGION=114K,
//                     PARM='XREF,ATR,LOAD,NODECK,NOMACRO,,OPT=1'
//SYSUT1     DD        UNIT=SYSDA,SPACE=(1024,(60,60),RLSE,,ROUND),
//                     DCB=BLKSIZE=1024,DISP=(,DELETE)
//SYSUT3     DD        UNIT=SYSDA,SPACE=(1024,(60,60),RLSE,,ROUND),
//                     DCB=BLKSIZE=1024,DISP=(,DELETE)
//SYSPRINT DD          SYSOUT=&SOUT,DCB-(LRECL=125,BLKSIZE=629,RECFM=VBA),
//                     space=(605,(&PAGES.0,&PAGES),RLSE)
//SYSLIN     DD        UNIT=SYSDA,SPACE=(80,(250,80),RLSE),DCB=BLKSIZE=80,
//                     DISP=(,PASS)
//L          EXEC      PGM=DFSILNK0,PARM='XREF,LIST,LET',COND=(4,LT,C),
//                     REGION=120K
//STEPLIB    DD        DSN=IMS2.RESLIB,DISP=SHR
//SYSLIB     DD        DSN=SYS1.PL1LIB,DISP=SHR
//           DD        DSN=SYS1.COBLIB,DISP=SHR
//RESLIB     DD        DSN=IMS2.RESLIB,DISP=SHR
//SYSLIN     DD        DSN=*.C.SYSLIN,DISP=(OLD,DELETE)
//           DD        DSN=IMS2.PROCLIB(PLITDLI),DISP=SHR
//           DD        DDNAME=SYSIN
//SYSLMOD    DD        DSN=IMS2.PGMLIB(&MBR),DISP=SHR
//SYSPRINT DD          SYSOUT=&SOUT,DCB=(LRECL=121,RECFM=FBA,BLKSIZE=605),
//                     SPACE=(605,(&PAGES.0,&PAGES),RLSE)
//SYSUT1     DD        UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(5,1),RLSE)
//G          EXEC      PGM=DFSRRC00,REGION=150K,TIME=5,COND=(4,LT),
//                     PARM='DLI,&MBR,&PSB,&BUF,&SPIE&TEST'
//STEPLIB    DD        DSN=IMS2.RESLIB,DISP=SHR
//           DD        DSN=IMS2.PGMLIB,DISP=SHR
//IMS        DD        DSN=IMS2.PSBLIB,DISP=SHR
//           DD        DSN=IMS2.DBDLIB,DISP=SHR
//IEFRDER    DD        DSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),UNIT=(2400,,DEFER),
//                     DCB=(RECFM=VBS,BLKSIZE=1408,LRECL=1400,BUFNO=1)
//SYSPRINT DD          SYSOUT=&SOUT,DCB=(LRECL=121,BLKSIZE=605,RECFM=FBA),
//                     SPACE=(605,(500,500),RLSE,,ROUND)
//SYSUDUMP DD          SYSOUT=&SOUT,DCB=(LRECL=121,BLKSIZE=605,RECFM=FBA),
//                     SPACE=(605,(500,500),RLSE,,ROUND)
```

Same assumptions as IMSCOBGO,
except execution time of 5 minutes specified

MEMBER NAME IMSRDR

```
//           PROC    MBR=IMSMSG
//IEFPROC    EXEC    PGM=IEFIRC,           READER FIRST LOAD
//                   REGION=48K                READER BASIC REGION
//                   PARM='00103005001024905210SYSDA    '  DEFAULT PARM FLD
//*                       BPPTTTOOOMMMIIICCCRLSSSSSSSS
//*
//*                  B            DEFINED PROGRAMMER NAME & ACCT NBR NOT NEEDED
//*                  PP           PRIORITY=01
//*                  TTT          JOB STEP INTERVAL=30 MINUTES
//*                  OOO          PRIMARY SYSOUT SPACE=50 TRACKS
//*                  MMM          SECONDARY SYSOUT SPACE=10 TRACKS
//*                  III          READER/INTERPRETER DISPATCHING PRIORITY=249
//*                  CCC          JOB STEP DEFAULT REGION=52K
//*                  R            DISPLAY & EXECUTE COMMANDS=1
//*                  L            BYPASS LABEL=0
//*                  SSSSSSSS     SYSOUT UNIT NAME=SYSDA
//*
//IEFRDER    DD      DSN=IMS2.PROCLIB(&MBR),DISP=SHR,DCB=BUFNO=1
//IEFPDSI    DD      DSN=IMS2.PROCLIB,DISP=SHR    PROCEDURE LIBRARY
//           DD      DSN=SYS1.PROCLIB,DISP=SHR
//IEFDATA    DD      UNIT=SYSDA,                  SPOOL DEVICE
//                   SPACE=(80,(500,500),RLSE,CONTIG),   AMOUNT
//                   DCB=(BUFNO=2,LRECL=80,BLKSIZE=80,RECFM=FB,BUFL=80
```

MEMBER NAME IMSWTnnn

```
//SPRTn      JOB     1,IMS,CLASS=A,MSGCLASS=3,MSGLEVEL=1
//PRINT      EXEC    PGM=DFSUPRT0,REGION=30K
//STEPLIB    DD      DSN=IMS2.RESLIB,DISP=SHR
//SYSPRINT   DD      SYSOUT=3,DCB=BLKSIZE=1410
//SYSUDUMP   DD      SYSOUT=3
//SPOOLn     DD      DSN=IMS2.SYSOn,DISP=SHR
```

Note: IMSWTnnn member(s) job class and message class are determined by the MAXREGN keyword specification on the IMSCTRL macro statement at System Definition time.

MEMBER NAME MFDBDUMP

```
//           PROC    SOUT=A
//DUMP       EXEC    PGM=DFSRRC00,PARM='DLI,DFSSAM08',REGION=130K
//STEPLIB    DD      DSN=IMS2.RESLIB,DISP=SHR
//           DD      DSN=IMS2.PGMLIB,DISP=SHR
//IMS        DD      DSN=IMS2.PSBLIB,DISP=SHR
//           DD      DSN=IMS2.DBDLIB,DISP=SHR
//SYSUDUMP   DD      SYSOUT=&SOUT
//DI21PART   DD      DSN=IMS2.DI21PART,DISP=SHR
//DI21PARO   DD      DSN=IMS2.DI21PARO,DISP=SHR
//OUTPUT     DD      SYSOUT=&SOUT
```

MEMBER NAME MFDBLOAD

```
//           PROC    SOUT=A
//LOAD       EXEC    PGM=DFSRRC00,PARM='DLI,DFSSAM01',REGION=130K
//STEPLIB    DD      DSN=IMS2.RESLIB,DISP=SHR
//           DD      DSN=IMS2.PGMLIB,DISP=SHR
//IMS        DD      DSN=IMS2.PSBLIB,DISP=SHR
//           DD      DSN=IMS2.DBDLIB,DISP=SHR
//SYSUDUMP   DD      SYSOUT=&SOUT
//DI21PART   DD      DSN=IMS2.DI21PART(PRIME),DISP=(,KEEP),DCB=DSORG=IS,
//                   SPACE=(CYL,3,,CONTIG),VOL=SER=&PSER,UNIT=&PUNIT
//DI21PARO   DD      DSN=IMS2.DI21PARO,DISP=(,KEEP),SPACE=(CYL,3,,CONTIG),
//                   VOL=SFR=&OSER,UNIT=&OUNIT
//SYSOUT     DD      SYSOUT=&SOUT
//INPUT      DD      DSN=IMS2.GENLIB(MFDFSYSN),DISP=SHR
```

MEMBER NAME PSBGEN

```
//           PROC    MBR=TEMPNAME,SOUT=A
//C          EXEC    PGM=IEUASM,REGION=120K,PARM='LOAD,NODECK'
//SYSLIB     DD      DSN=IMS2.MACLIB,DISP=SHR
//SYSGO      DD      UNIT=SYSDA,DISP=(,PASS),SPACE=(80,(100,100),RLSE),
//                   DCB=(BLKSIZE=400,RECFM=FB,LRECL=80)
//SYSPRINT   DD      SYSOUT=&SOUT,DCB=(LRECL=121,RECFM=FBM,BLKSIZE=605),
//                   SPACE=(121,(500,500),RLSE,,ROUND)
//SYSUT1     DD      UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(100,50))
//SYSUT2     DD      UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(100,50))
//SYSUT3     DD      UNIT=(SYSDA,SEP=(SYSLIB,SYSUT1,SYSUT2)),
//                   SPACE=(1700,(100,50))
//L          EXEC    PGM=DFSILNK0,PARM='XREF,LIST',COND=(0,LT,C),REGION=120K
//STEPLIB    DD      DSN=IMS2.RESLIB,DISP=SHR
//SYSLIN     DD      DSN=*.C.SYSGO,DISP=(OLD,DELETE)
//SYSPRINT   DD      SYSOUT=&SOUT,DCB=(LRECL=121,RECFM=FBA,BLKSIZE=605),
//                   SPACE=(121,(100,100),RLSE)
//SYSLMOD    DD      DSN=IMS2.PSBLIB(&MBR),DISP=SHR
//SYSUT1     DD      UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),DISP=(,DELETE),
//                   SPACE=(1024,(100,10),RLSE)
```

MEMBER NAME SECURITY

```
//           PROC    OPTN=UPDATE,IMS=',0',SOUT=A
//S          EXEC    PGM=DFSISMP0,PARM='&OPTN.&IMS.'
//STEPLIB    DD      DSN=IMS2.RESLIB,DISP=SHR
//SYSPRINT   DD      SYSOUT=&SOUT,DCB=(RECFM=VBA,BLKSIZE=400,BUFL=404)
//SYSPUNCH   DD      UNIT=SYSDA,SPACE=(80,(800,400),,,ROUND),
//                   DCB=(RECFM=FB,LRECL=80,BLKSIZE=400),DISP=(,PASS)
//SYSLIN     DD      UNIT=SYSDA,SPACE=(TRK,(1,1)),DCB=(RECFM=F,BLKSIZE=80),
//                   DISP=(,PASS)
//SYSUT1     DD      UNIT=SYSDA,SPACE=(100,(400,400),,,ROUND),
//                   DCB=(BLKSIZE=500,RECFM=FB)
//SYSUT2     DD      UNIT=(SYSDA,SEP=SYSUT1),SPACE=(100,(400,400),,,ROUND),
//                   DCB=*.S.SYSUT1
//SYSIN      DD      DSN=NO.SYSIN.DD.ASTERISK
//C          EXEC    PGM=IEUASM,PARM='LOAD,NODECK',COND=(12,LT,S),
                     REGION=96K
//SYSPRINT   DD      SYSOUT=&SOUT,DCB=(RECFM=FBM,LRECL=121,BLKSIZE=605)
//SYSGO      DD      UNIT=(SYSDA,SEP=SYSPRINT),DISP=(,PASS),
//                   DCB=*.S.SYSPUNCH,SPACE=(80,(400,400),,,ROUND)
//SYSUT1     DD      UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT2     DD      UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT3     DD      UNIT=(SYSDA,SEP=(SYSUT1,SYSUT2)),SPACE=(CYL,(5,1))
//SYSIN      DD      DSN=*.S.SYSPUNCH,DISP=(OLD,DELETE)
//L          EXEC    PGM=DFSILNK0,PARM='XREF,NE,OL',REGION=110K,COND=(4,LT,S)
//STEPLIB    DD      DSN=IMS2.RESLIB,DISP=SHR
//SYSPRINT   DD      SYSOUT=&SOUT,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605)
//SYSLMOD    DD      DSN=IMS2.RESLIB,DISP=SHR
//INPUT      DD      DSN=*.C.SYSGO,DISP=(OLD,DELETE)
//SYSUT1     DD      UNIT=(SYSDA,SEP=INPUT),SPACE=(CYL,(5,1))
//SYSLIN     DD      DSN=*.S.SYSLIN,DISP=(OLD,DELETE)
```

MEMBER NAME IQFUT

```
//           PROC     SOUT=A,SPCH=B,IMSREG=DLI
//SIA        EXEC     PGM=DMGSI1,REGION=120K
//STEPLIB    DD       DSN=IMS2.RESLIB,DISP=SHR
//SYSUT1     DD       UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(24,11))
//SYSPRINT DD         SYSOUT=&SOUT
//SYSPUNCH DD         SYSOUT=&SPCH
//SIB        EXEC     PGM=DFSRRC00,PARM='&IMSREG,DMGSI2,DMGSIB',REGION=150K,     *
//                    COND=(0,LT)
//STEPLIB    DD       DSN=IMS2.RESLIB,DISP=SHR
//IMS        DD       DSN=IMS2.PSBLIB,DISP=SHR
//           DD       DSN=IMS2.DBDLIB,DISP=SHR
//SYSPRINT DD         SYSOUT=&SOUT
//SYSPUNCH DD         SYSOUT=&SPCH
//QFF        DD       DSN=IQFIFFDB,DISP=SHR
//QFFOVF     DD       DSN=IQFOFFDB,DISP=SHR
//SYSUT1     DD       DSN=*.SIA.SYSUT1,DISP=(OLD,DELETE)
```

Assumes

1. User supplies source data for SYSIN.
2. SYSUT1 is a BSAM work data set.
3. Output Class A is used for listing.
4. Output Class B is used by DMGSI1 and DMGSI2 (Stage I) to produce job steps in the Stage II OS job stream
5. User defines IMS region type (Batch or Batch-Message) in PARM field of EXEC statement for executing the procedure. (Not required at initial creation time.)

Note: The SIB step is bypassed when the IQFUT procedure is executed to create the System Data Base.

MEMBER NAME IQFFC

```
//IQFFC      PROC
//FC1        EXEC     PGM=DFSRRC00,PARM='DLI,DMGFC1,DMGFC1',REGION=200K
//STEPLIB    DD       DSN=IMS2.RESLIB,DISP=SHR
//IMS        DD       DSNAME=IMS2.PSBLIB,DISP=SHR
//           DD       DSNAME=IMS2.DBDLIB,DISP=SHR
//SYSPRINT   DD       SYSOUT=A
//SYSOUT     DD       SYSOUT=A
//UTPRINT    DD       SYSOUT=A
//UTDBD      DD       UNIT=SYSDA,DSN=UTDBD,DISP=(NEW,DELETE),SPACE=(CYL,(1,1))
//UTSPL      DD       UNIT=SYSDA,DSN=UTSPL,DISP=(NEW,DELETE),SPACE=(CYL,(1,1))
//SORTLIB    DD       DSN=SYS1.SORTLIB,DISP=SHR
//SSYNIN     DD       DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),UNIT=SYSDA,           *
//                    DCB=(BLKSIZE=1040,LRECL=52,DSORG=PS,RECFM=FB),            *
//                    DSN=SSYNIN
//SSYNOUT    DD       DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),UNIT=SYSDA,           *
//                    DCB=(BLKSIZE=1040,LRECL=52,DSORG=PS,RECFM=FB),            *
//                    DSN=SSYNOUT
//SPCBIN     DD       DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),UNIT=SYSDA,           *
//                    DCB=(BLKSIZE=880,LRECL=44,DSORG=PS,RECFM=FB),             *
//                    DSN=SPCBIN
//SPCBOUT    DD       DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),UNIT=SYSDA,           *
//                    DCB=(BLKSIZE=880,LRECL=44,DSORG=PS,RECFM=FB),             *
//                    DSN=SPCBOUT
//SWRKIN     DD       DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),UNIT=SYSDA,           *
//                    DCB=(BLKSIZE=1920,LRECL=96,DSORG=PS,RECFM=FB),            *
//                    DSN=SWRKIN
//SWRKOUT    DD       DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),UNIT=SYSDA,           *
//                    DCB=(BLKSIZE=1920,LRECL=96,DSORG=PS,RECFM=FB),            *
//                    DSN=SWRKOUT
//SPCBWK01 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SPCBWK02 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SPCBWK03 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SPCBWK04 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SPCBWK05 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SPCBWK06 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SSYNWK01 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SSYNWK02 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SSYNWK03 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SSYNWK04 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SSYNWK05 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SSYNWK06 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SWRKWK01 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SWRKWK02 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SWRKWK03 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SWRKWK04 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SWRKWK05 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
//SWRKWK06 DD         UNIT=SYSDA,SPACE=(TRK,(5),,CONTIG)
```

Assumes

1. The DMGSI1 program (Stage I, Part 1) provides JCL to allocate data set groups at initial creation time.

Member Name IQFIU

```
//IQFIU     PROC    SOUT=A,IMSREG=DLI,DISPS=OLD,SSTLB=516,LSTLB=516
//IU1       EXEC    PGM=DFSRRC00,PARM='&IMSREG,DMGIU1,DMGIU1',REGION=200K
//STEPLIB   DD      DSN=IMS2.RESLIB,DISP=SHR
//IMS       DD      DSN=*.QUS2X1.L.SYSLMOD,DISP=(OLD,PASS)
//          DD      DSN=IMS2.PSBLIB,DISP=SHR
//          DD      DSN=IMS2.DBDLIB,DISP=SHR
//QFF       DD      DSN=IQFIFFDB,DISP=SHR
//QFFOVF    DD      DSN=IQFOFFDB,DISP=SHR
//QXS1      DD      DSN=IQFXS1DB,DISP=&DISPS
//QXS1OV    DD      DSN=IQFXOVS1,DISP=&DISPS
//QXL1      DD      DSN=IQFXL1DB,DISP=&DISPS
//QXL1OV    DD      DSN=IQFXOVL1,DISP=&DISPS
//HOLDS     DD      UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,PASS)
//HOLDL     DD      UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,PASS)
//IEFRDER   DD      DUMMY
//SYSPRINT  DD      SYSOUT=&SOUT
//SYSOUT    DD      SYSOUT=&SOUT
//IU2       EXEC    PGM=DFSRRC00,PARM='&IMSREG,DMGIU3,DMGIU1',REGION=200K,
//                  COND=(4,LT,IU1)
//STEPLIB   DD      DSN=IMS2.RESLIB,DISP=SHR
//IMS       DD      DSN=*.QUS2X1.L.SYSLMOD,DISP=(OLD,PASS)
//          DD      DSN=IMS2.PSBLIB,DISP=SHR
//          DD      DSN=IMS2.DBDLIB,DISP=SHR
//QFF       DD      DSN=IQFIFFDB,DISP=SHR
//QFFOVF    DD      DSN=IQFOFFDB,DISP=SHR
//IEFRDER   DD      DUMMY
//SYSPRINT  DD      SYSOUT=&SOUT
//SYSOUT    DD      SYSOUT=&SOUT
//SORTLIB   DD      DSN=SYS1.SORTLIB,DISP=SHR
//SHRTIN    DD      DSN=*.IU1.HOLDS,DISP=(OLD,DELETE),
//                  DCB=(BLKSIZE=&SSTLB,LRECL=&SSTLB,RECFM=FB,DSORG=PS)
//SHRTOUT   DD      UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,PASS),
//                  DCB=(BLKSIZE=&SSTLB,LRECL=&SSTLB,RECFM=FB,DSORG=PS)
//SHRTWK01  DD      UNIT=SYSDA,SPACE=(TRK,(10),,CONTIG)
//SHRTWK02  DD      UNIT=SYSDA,SPACE=(TRK,(10),,CONTIG)
//SHRTWK03  DD      UNIT=SYSDA,SPACE=(TRK,(10),,CONTIG)
//LONGIN    DD      DSN=*.IU1.HOLDL,DISP=(OLD,DELETE),
//                  DCB=(BLKSIZE=&LSTLB,LRECL=&LSTLB,RECFM=FB,DSORG=PS)
//LONGOUT   DD      UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,PASS),
//                  DCB=(BLKSIZE=&LSTLB,LRECL=&LSTLB,RECFM=FB,DSORG=PS)
//LONGWK01  DD      UNIT=SYSDA,SPACE=(TRK,(10),,CONTIG)
//LONGWK02  DD      UNIT=SYSDA,SPACE=(TRK,(10),,CONTIG)
//LONGWK03  DD      UNIT=SYSDA,SPACE=(TRK,(10),,CONTIG)
//IU3       EXEC    PGM=DFSRRC00,PARM='&IMSREG,DMGIU2,DMGIU1',REGION=200K,
//                  COND=((4,LT,IU1),(4,LT,IU2))
//STEPLIB   DD      DSN=IMS2.RESLIB,DISP=SHR
//IMS       DD      DSN=*.QUS2X1.L.SYSLMOD,DISP=(OLD,DELETE)
//          DD      DSN=IMS2.PSBLIB,DISP=SHR
//          DD      DSN=IMS2.DBDLIB,DISP=SHR
//QFF       DD      DSN=IQFIFFDB,DISP=SHR
//QFFOVF    DD      DSN=IQFOFFDB,DISP=SHR
//QXS1      DD      DSN=IQFXS1DB,DISP=&DISPS
//QXS1OV    DD      DSN=IQFXOVS1,DISP=&DISPS
//QXL1      DD      DSN=IQFXL1DB,DISP=&DISPS
//QXL1OV    DD      DSN=IQFXOVL1,DISP=&DISPS
//HOLDS     DD      DSN=*.IU2.SHRTOUT,UNIT=SYSDA,DISP=(OLD,DELETE)
//HOLDL     DD      DSN=*.IU2.LONGOUT,UNIT=SYSDA,DISP=(OLD,DELETE)
//SYSPRINT  DD      SYSOUT=&SOUT
//SYSOUT    DD      SYSOUT=&SOUT
```

The *.QUS2X1.L.SYSLMOD data set for the IMS DD card refers back to the SYSLMOD card in the DMGIU1 PSBGEN step generated by DMGSI2.

Assumes

Prior to executing the IQF Utility at IQF/IMS installation time, the user modifies this procedure to tailor it to his IQF indexing requirements. The modifications required are:

1. Add DD statements to the IU1 step for the user's IMS/360 data bases to be indexed

MEMBER NAME CBLTDLI

LIBRARY RESLIB (CBLTDLI) DL/I INTERFACE ENTRY DLITCBL

MEMBER NAME PLITDLI

LIBRARY RESLIB (PLITDLI) DL/I LANGUAGE INTERFACE ENTRY IHESAPD

SYSTEM DEFINITION ERROR CONDITIONS

IMS/360 system definition error conditions are listed in the IMS/360 Messages and Codes Reference Manual (SH20-0914).

SYSTEM DEFINITION EXAMPLES

Two examples of system definition are given, one for a complete IMS/360 data communication - data base system and the other for a batch-only IMS/360 data base system.

The data communication - data base example, which includes the IMS/360 control program region and message processing and batch message processing regions, assumes the following:

- OS programming system, MVT configuration
- Thirty-four application programs
- Thirty-six transaction codes against those application programs
- Sixteen data bases
- Line groups:

  One 2740 nonswitched line group with two lines.

    The first line has two 2740 Model 1 terminals.

    The second line has two 2740 Model 2 terminals.

  One 2740 non-station control line group containing one line and one terminal

  One 3270 remote line group containing one line, one 3275 terminal, and one 3271 control unit supporting five terminals

  One local 3270 line group containing one line and two terminals

  One local 3270 line group containing one line and one 3286 printer terminal

  One 1050 nonswitched line group containing one line and two terminals

  One 2260-remote line group containing one line, and two control units, one 2265 terminal, two 2260 terminals, and one 1053 printer

  One 2770 nonswitched line group containing one line and one terminal

  One 2980 nonswitched line group containing one line, with one control unit supporting three terminals

  One 2741 nonswitched line group containing one line and one terminal

  Two 2741 switched line groups

  One 1030 nonswitched line group with one line and two terminals

  Four Local SYSOUT line groups

  One 7770 line group with two lines

  One TWX switched line group

  One 2740 switched line group

  One 1050 switched line group with one line and one pool with four subpools

DATA BASE - DATA COMMUNICATION EXAMPLE

This example illustrates the output from Stage 1 of the IMS/360 system definition utility program. The input to Stage 1 (that is, the control cards) is provided in the output listing followed by a summary of the Featgroup Specifications, the Application Specifications, the Communication Specifications, and the Data Set and Data Base Specifications. Next is the punch statements in this example followed by the comments considered warnings.

Figure 7 shows, in summary form, the various transaction codes, programs, and data bases, including their relationship to each other, as they exist in the following example of system definition.

Figure 8 shows, in summary form, the teleprocessing relationship as it exists in the following example of system definition.

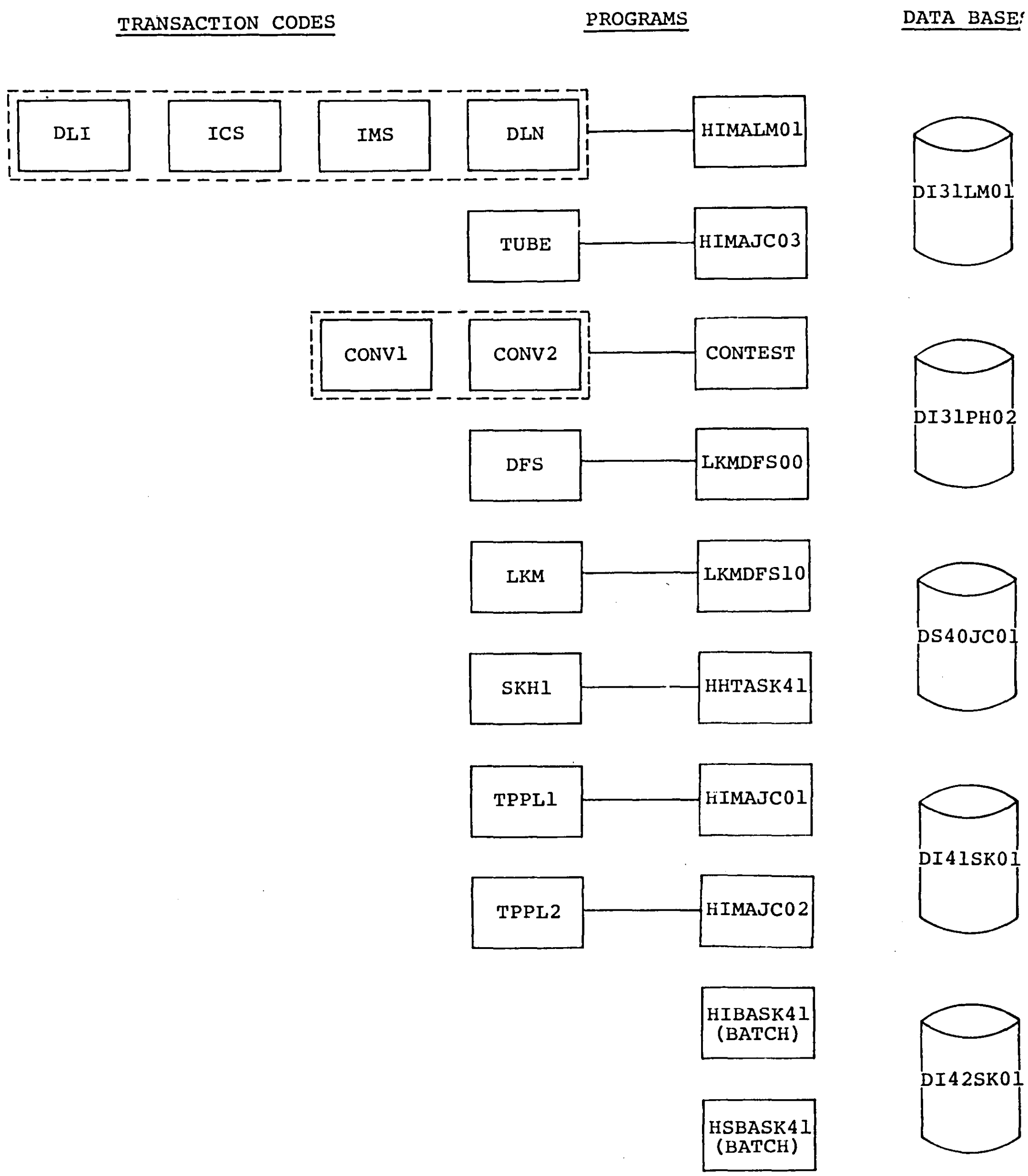


Figure 7. (Sheet 1 of 2.) System definition example summary - transaction codes, programs, and data bases

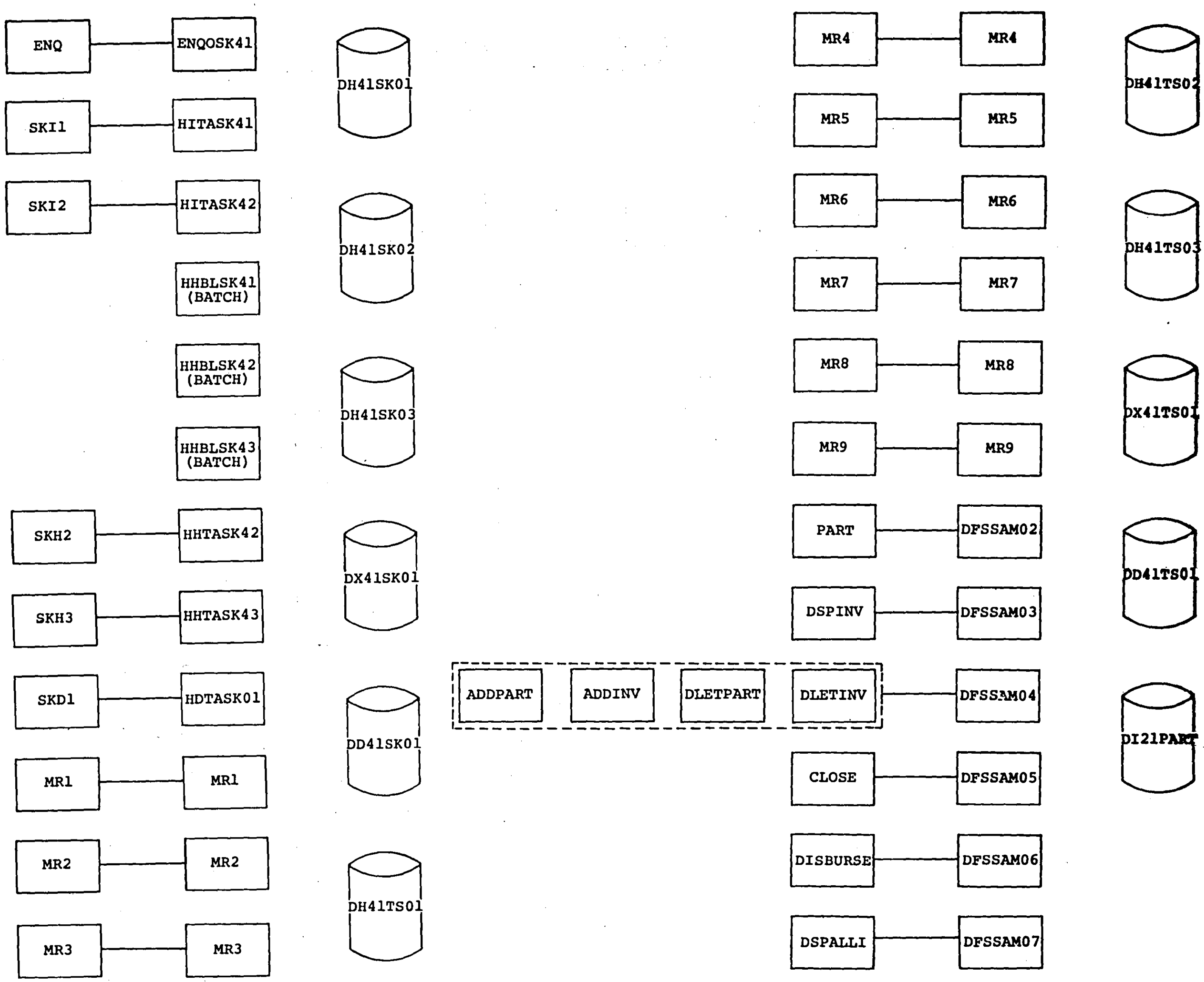


Figure 7. (Sheet 2 of 2.) System definition example summary - transaction codes, programs, and data bases

2740 NONSWITCHED LINE GROUP CONFIGURATION

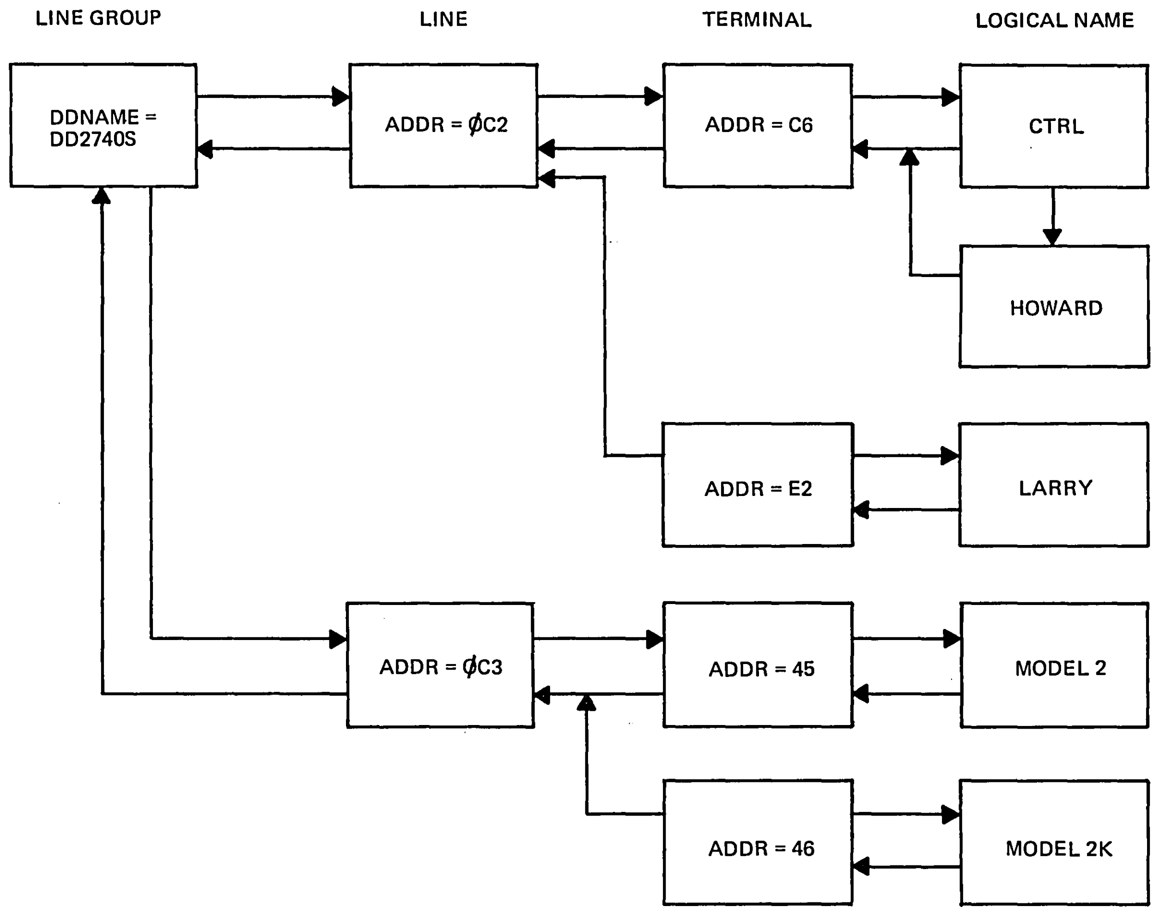


Figure 8. (page 1 of 16) System definition example summary - teleprocessing relationship

2740 NON-STATION CONTROL CONFIGURATION

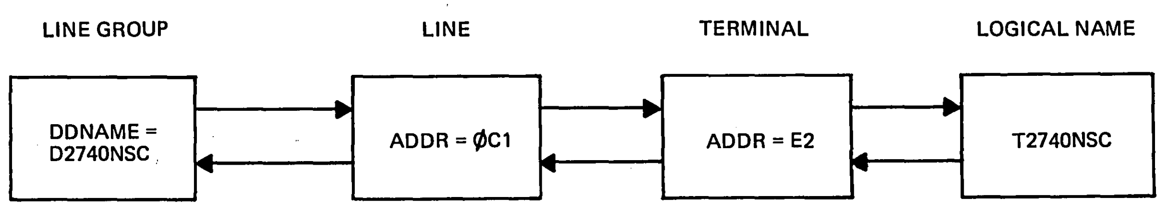


Figure 8 (page 2 of 16)

3270 REMOTE LINE CONFIGURATION

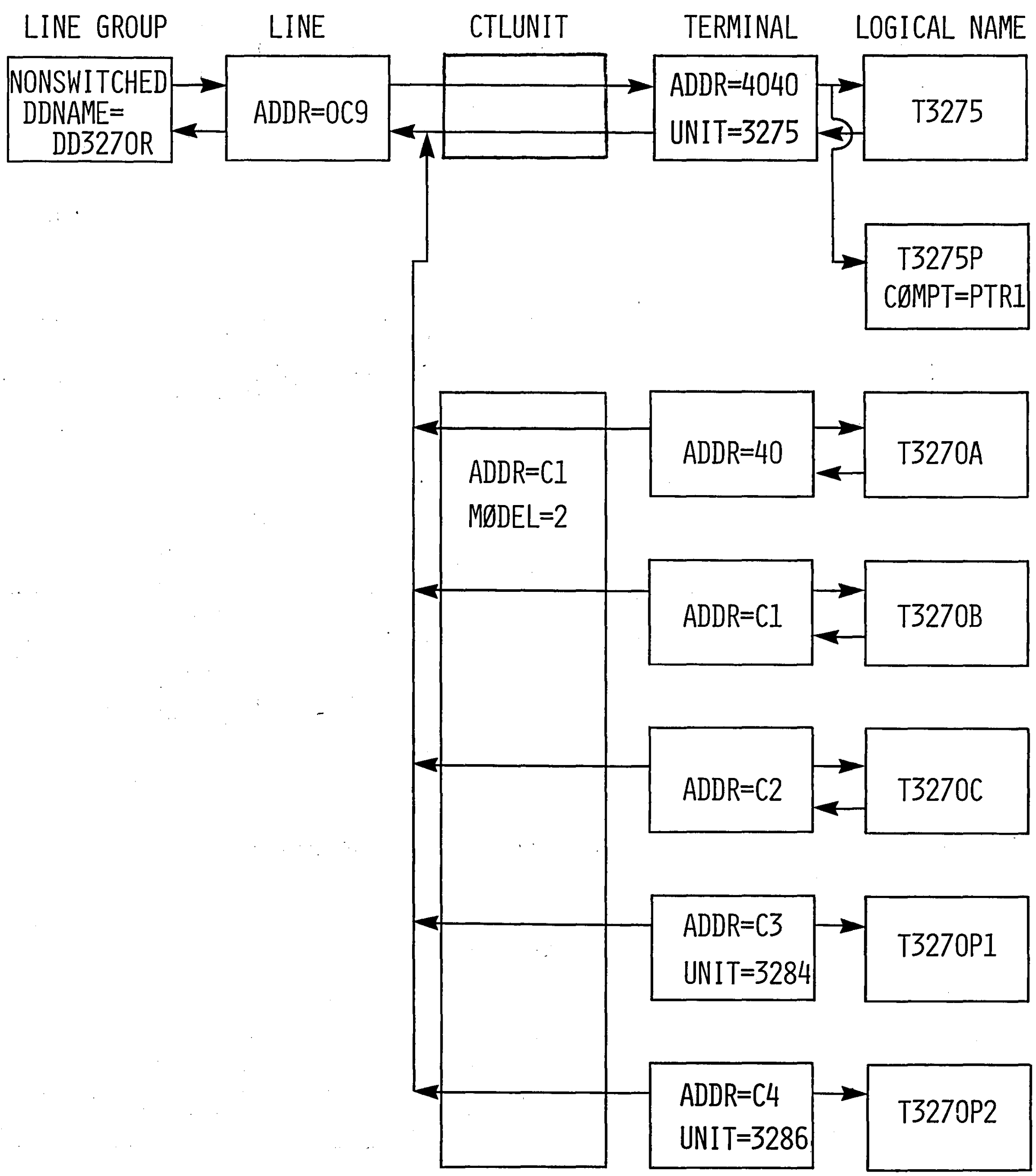
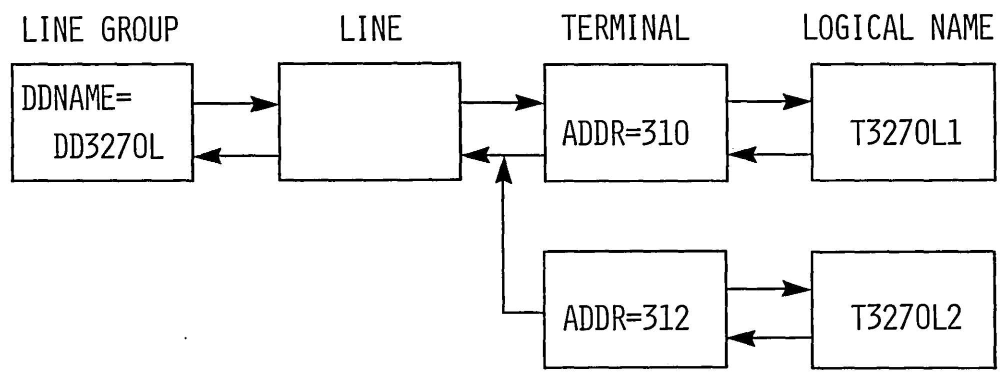


Figure 8 (page 3 of 16)

3270 LOCAL LINE CONFIGURATION



3270 LOCAL LINE CONFIGURATION

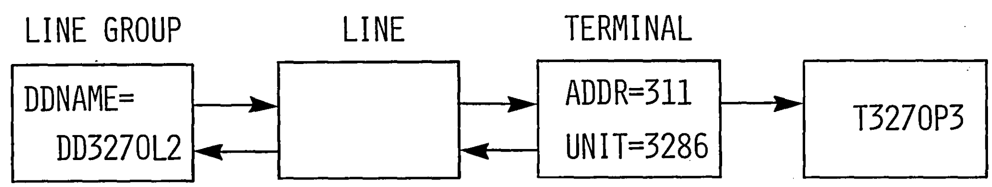


Figure 8 (page 4 of 16)

1050 NONSWITCHED LINE GROUP CONFIGURATION

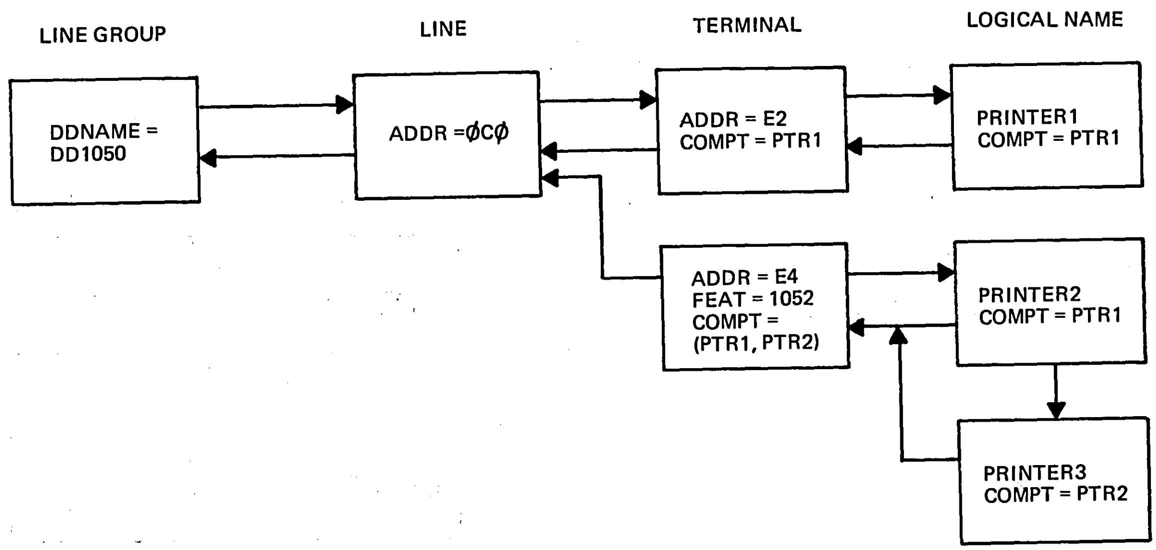


Figure 8 (page 5 of 16)

REMOTE 2260 LINE GROUP CONFIGURATION

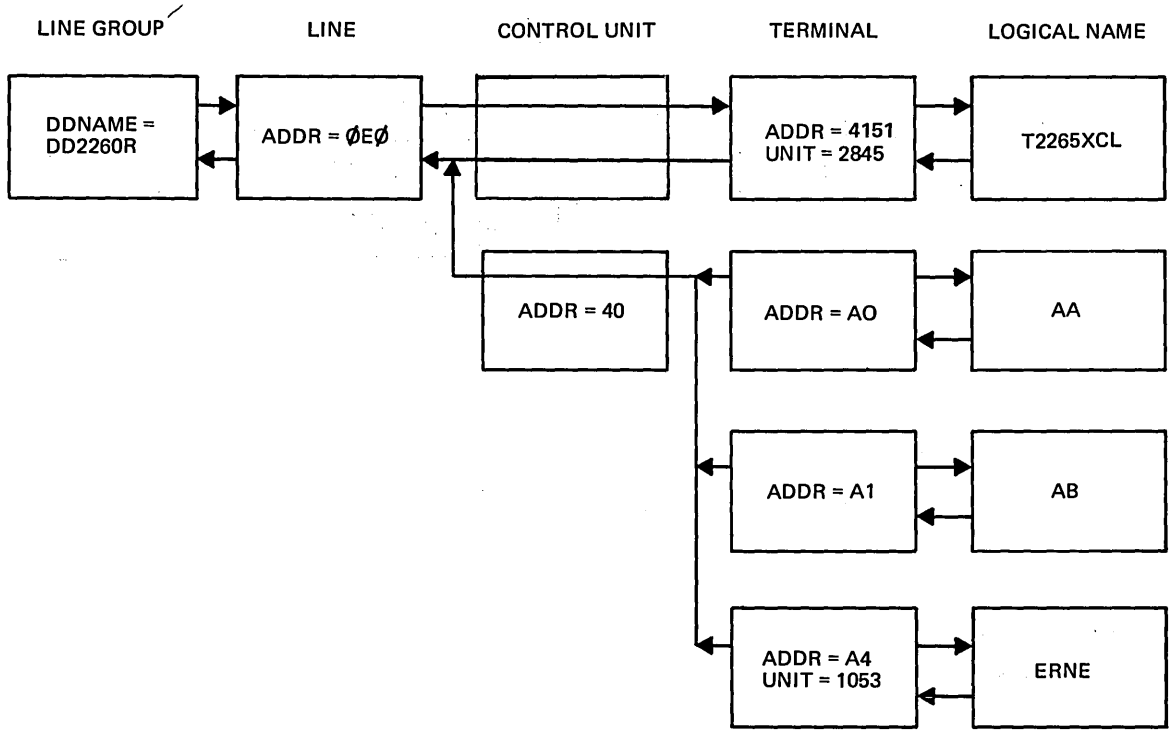


Figure 8 (page 6 of 16)

2770 NONSWITCHED LINE GROUP CONFIGURATION

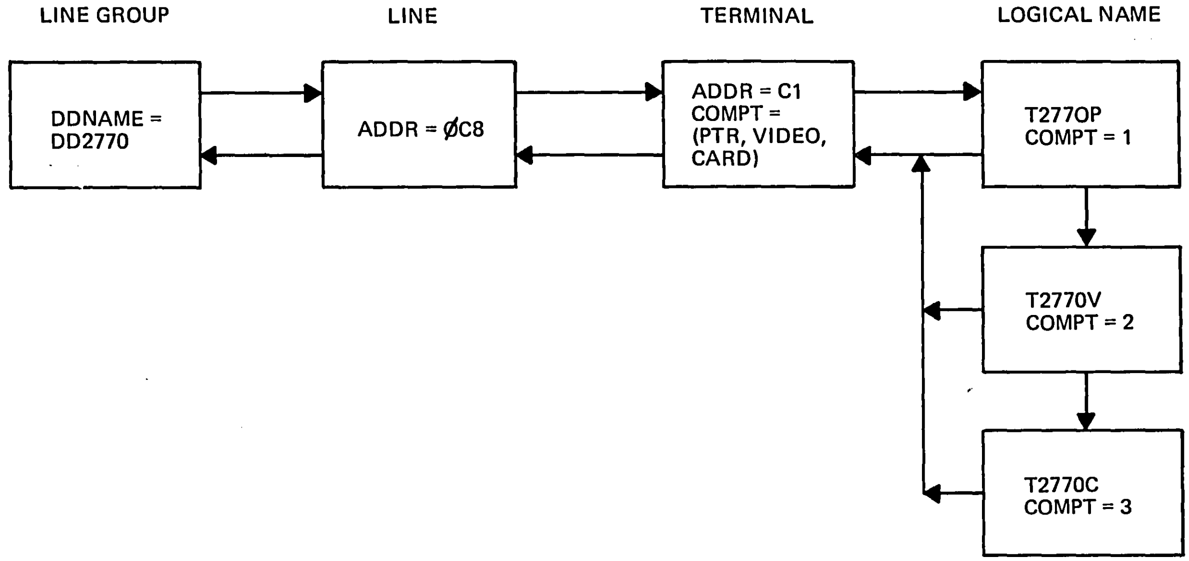


Figure 8 (page 7 of 16)

MULTIPOINT 2980 LINE GROUP CONFIGURATION

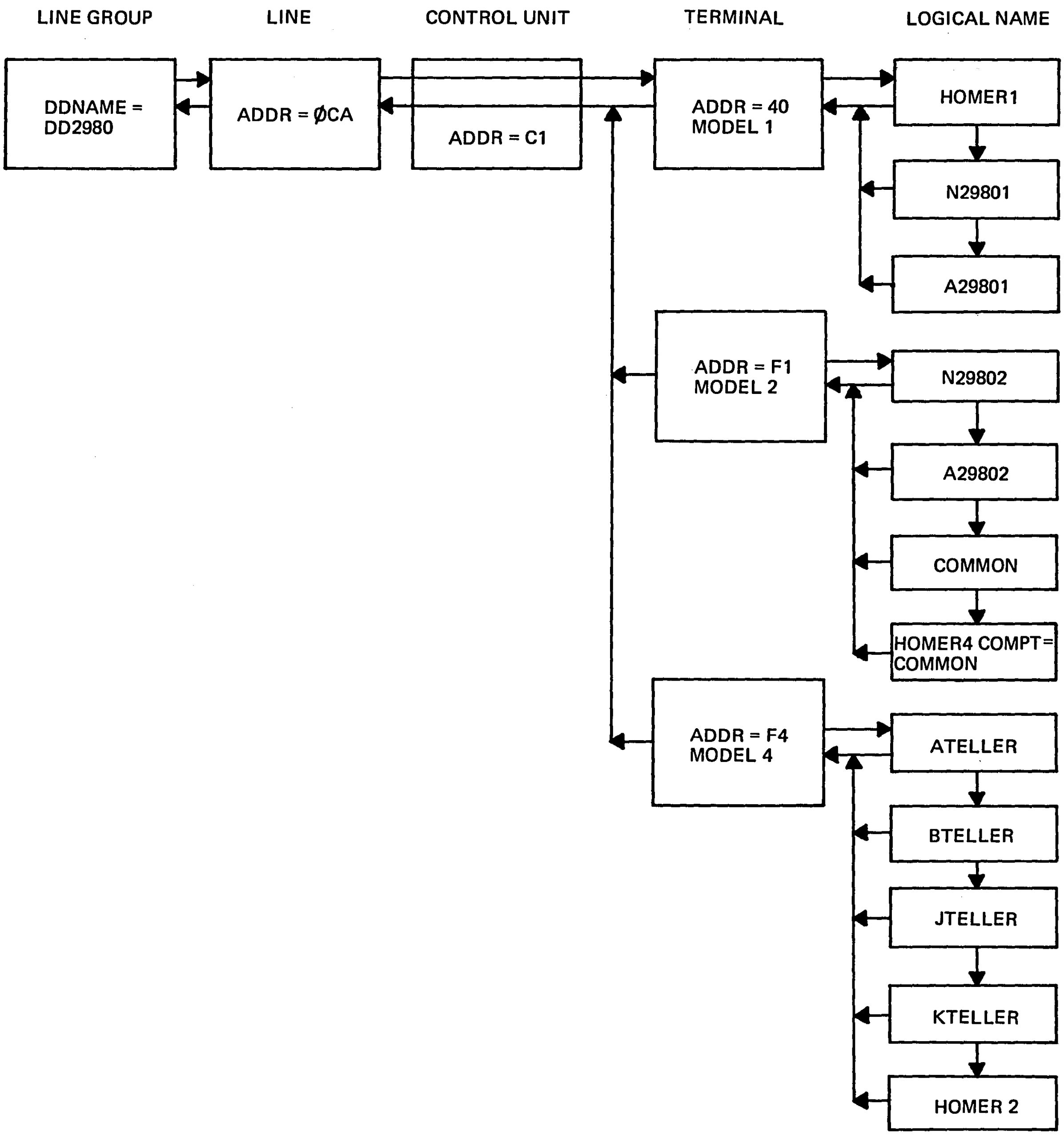


Figure 8 (page 8 of 16)

2741 NONSWITCHED LINE GROUP CONFIGURATION

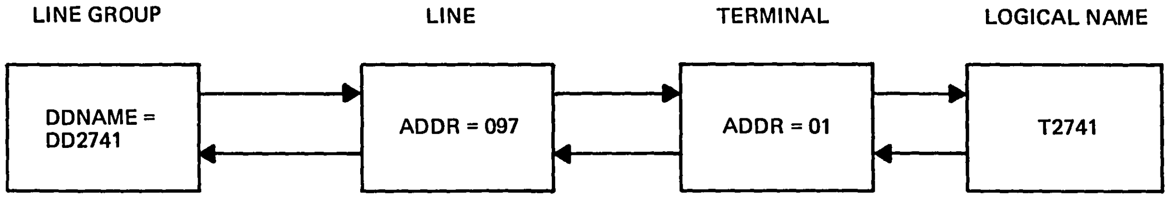


Figure 8 (page 9 of 16)

2741 SWITCHED LINE GROUP CONFIGURATION

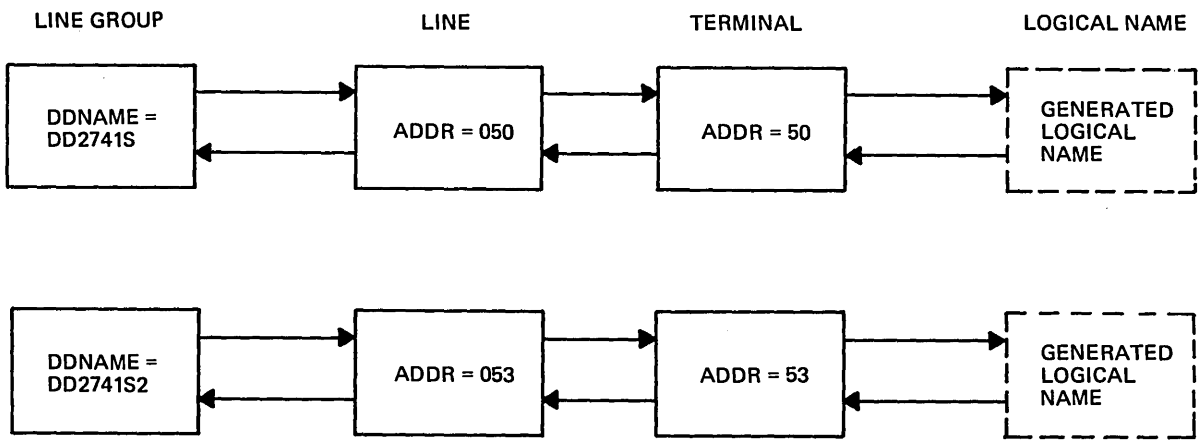


Figure 8 (page 10 of 16)

1030 LINE GROUP CONFIGURATION

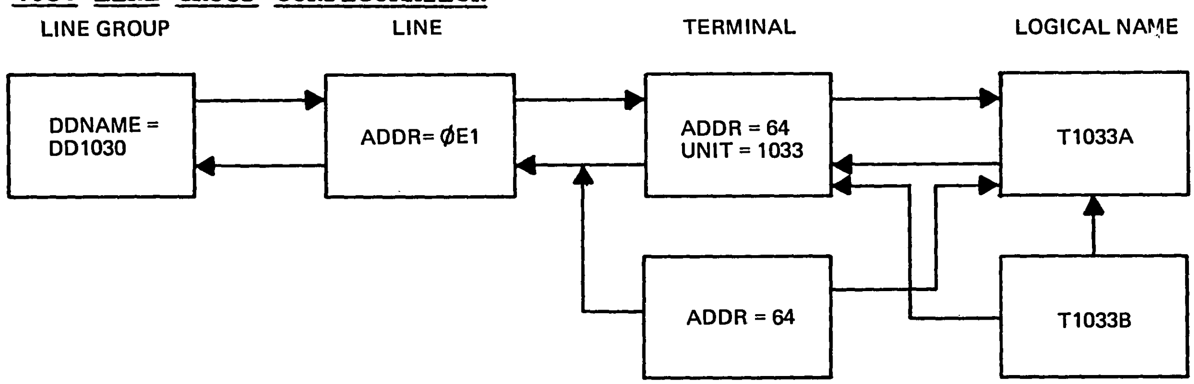


Figure 8 (page 11 of 16)

LOCAL SYSOUT LINE GROUP CONFIGURATIONS

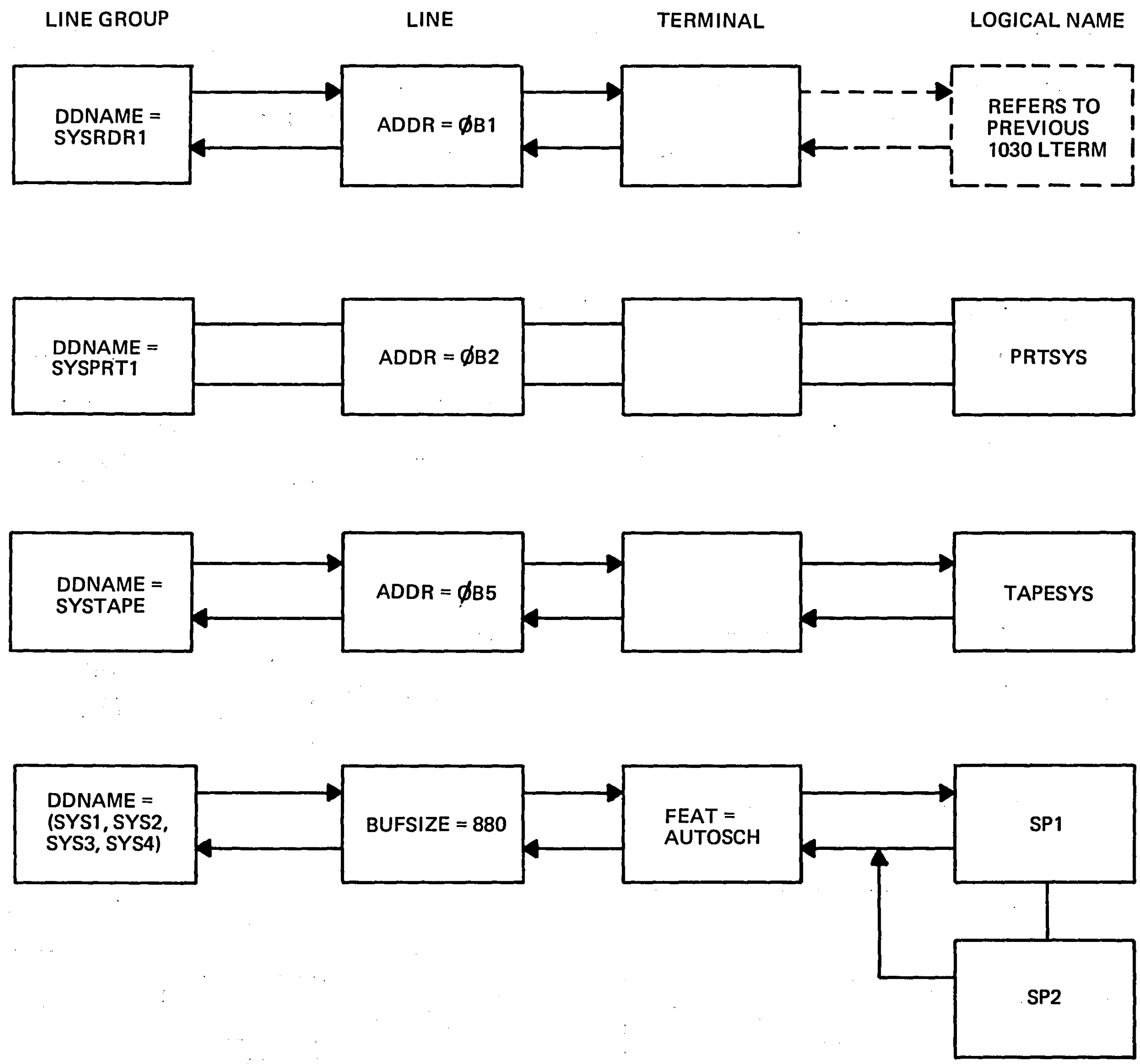


Figure 8 (page 12 of 16)

7770 LINE GROUP CONFIGURATION

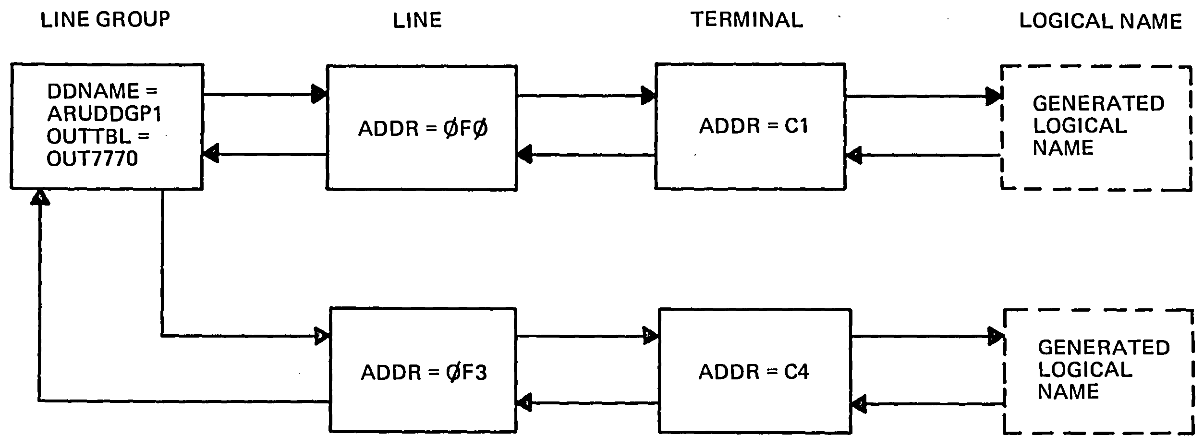


Figure 8 (page 13 of 16)

TWX LINE GROUP CONFIGURATION

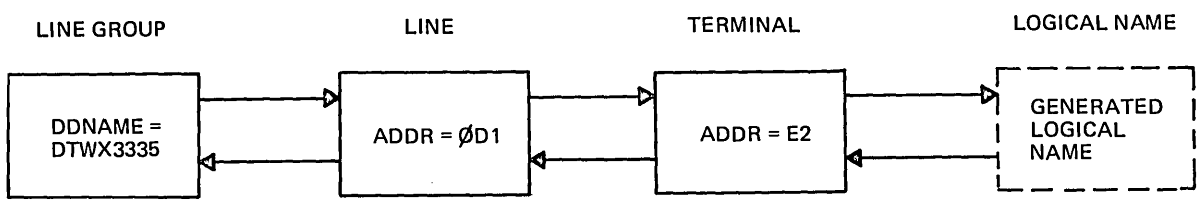


Figure 8 (page 14 of 16)

2740 SWITCHED LINE GROUP CONFIGURATION

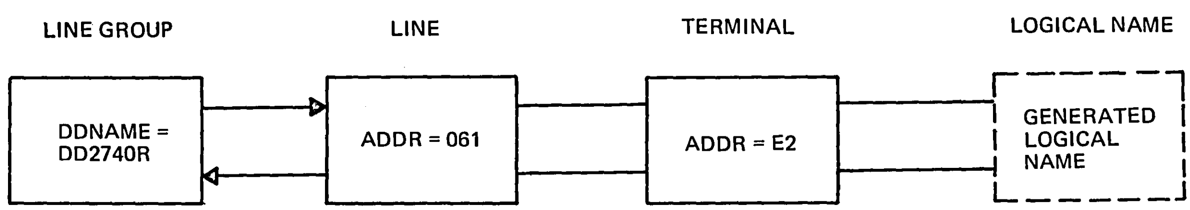


Figure 8 (page 15 of 16)

1050 SWITCHED LINE GROUP CONFIGURATION

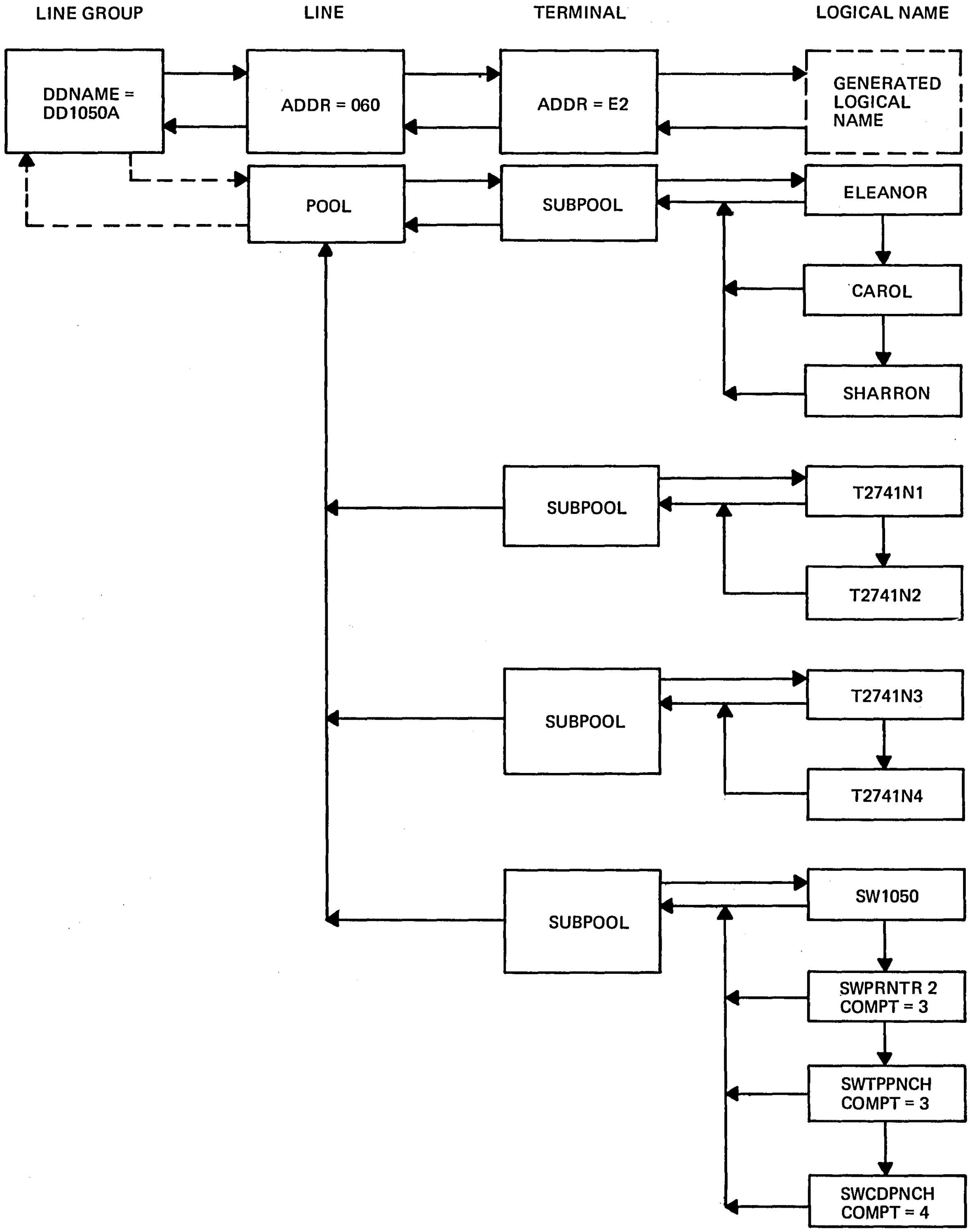


Figure 8 (page 16 of 16)

SYSTEM DEFINITION EXAMPLE

This example illustrates the output from Stage 1 of IMS/360 system definition. The input to Stage 1 (that is, the control cards) is provided in the output listing as is a summary of the Data Communication and Data Base specifications, followed by the punch statements and warning comments at the end.

```
                                                                                                PAGE    1

LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  F01OCT71  10/02/72

                                   1 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                                   2 *                                                                        *
                                   3 *        SAMPLE SYSTEM DEFINITION FOR IMS VERSION 2.3                    *
                                   4 *                                                                        *
                                   5 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

                                   7        IMSCTRL SYSTEM=(MVT,ALL),MAXREGN=(5,30K,A,3),                     C
                                                  MCS=13,DESC=2,MAXCLAS=5

                                   9              *,       ALL IMS SYSTEM FUNCTIONS ARE SELECTED
                                  10              *,       FOR A MVT OS PROGRAMMING SYSTEM.
                                  11              *,       5 REGIONS MAY OPERATE SIMULTANEOUSLY.

                                  13              *,          THE NUMBER OF CONCURRENTLY OPERATING
                                  14              *,       SUBTASKS WILL BE OPTIMIZED TO EQUAL THE
                                  15              *,       NUMBER OF SPECIFIED COMMUNICATION LINES
                                  16              *,       DIVIDED BY TWO.

                                  18              *,       5 TRANSACTION CODE
                                  19              *,       CLASSES WILL BE SUPPORTED.

                                  21         IMSCTF SVCNO=(234,248,213),APNDG=(Z5,Z6),                        C
                                                  CPLOG=2500,CORE=(,2000)


                                  23 *
                                  24 *            DEFINE SYSTEM BUFFERS AND DATA SETS
                                  25 *

                                  27         SPAREA CORE=(3,100),DASD=(6,150)
                                  28       BUFPOOLS PSB=8000,DMB=10000,DBASE=11000,GENERAL=5000
                                  29       MSGQUEUE DSETS=2314
                                                                                                PAGE    2

LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  F01OCT71  10/02/72

                                  31 *
                                  32 *            DEFINE DL/I DATABASES
                                  33 *
                                  34       DATABASE DBD=DI31LM01
                                  35       DATABASE DBD=DI31PH02
                                  36       DATABASE DBD=DI21PART
                                  37       DATABASE DBD=DS40JC01
                                  38       DATABASE DBD=DI41SK01
                                  39       DATABASE DBD=DI42SK01
                                  40       DATABASE DBD=DH41SK01
                                  41       DATABASE DBD=DH41SK02
                                  42       DATABASE DBD=DH41SK03
                                  43       DATABASE DBD=DX41SK01
                                  44       DATABASE DBD=DD41SK01
                                  45       DATABASE DBD=DH41TS01
                                  46       DATABASE DBD=DH41TS02
                                  47       DATABASE DBD=DH41TS03
                                  48       DATABASE DBD=DX41TS01
                                  49       DATABASE DBD=DD41TS01


                                  51 *
                                  52 *            DEFINE SAMPLE APPLICATION
                                  53 *
                                  54        APPLCTN      PSB=DFSSAM02,PGMTYPE=(TP,,4)
                                  55           TRANSACT CODE=PART,PRTY=(7,10,2),INQ=YES
                                  56       APPLCTN      PSB=DFSSAM03
                                  57           TRANSACT CODE=DSPINV,PRTY=(7,10,2),INQ=YES,MSGTYPE=(,,4)
                                  58       APPLCTN      PSB=DFSSAM04,PGMTYPE=(,,4)
                                  59           TRANSACT CODE=ADDPART,PRTY=(7,10,2)
                                  60           TRANSACT CODE=ADDINV,PRTY=(7,10,2),MSGTYPE=(,,5)
                                  61           TRANSACT CODE=DLETPART,PRTY=(7,10,2)
                                  62           TRANSACT CODE=DLETINV,PRTY=(7,10,2),MSGTYPE=(,,5)
                                  63       APPLCTN      PSB=DFSSAM05,PGMTYPE=(,,3)
                                  64           TRANSACT CODE=CLOSE,PRTY=(7,10,2)
                                  65       APPLCTN      PSB=DFSSAM06
                                  66           TRANSACT CODE=DISBURSE,PRTY=(7,10,2),MSGTYPE=(,,2)
                                  67       APPLCTN      PSB=DFSSAM07,PGMTYPE=(,,2)
                                  68           TRANSACT CODE=DSPALLI,PRTY=(7,10,2),INQ=YES
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                    F01OCT71  10/02/72

                                  70 *
                                  71 *            DEFINE DL/I APPLICATIONS
                                  72 *
                                  73      APPLCTN   PSB=CONTEST
                                  74       TRANSACT CODE=CONV1,PRTY=(8,8),PROCLIM=(3,60),SPA=80,MODE=SNGL
                                  75       TRANSACT CODE=CONV2,PRTY=(8,8),PROCLIM=(3,60),                  C
                                                    SPA=(80,CORE),MODE=SNGL
                                  76      APPLCTN   PSB=ENQOSK41
                                  77       TRANSACT CODE=ENQ,PRTY=(8,8)
                                  78      APPLCTN   PSB=HDTASK01
                                  79       TRANSACT CODE=SKD1,PRTY=(8,8)
                                  80      APPLCTN   PSB=HHBLSK41,PGMTYPE=BATCH
                                  81      APPLCTN   PSB=HHBLSK42,PGMTYPE=BATCH
                                  82      APPLCTN   PSB=HHBLSK43,PGMTYPE=BATCH
                                  83      APPLCTN   PSB=HHTASK41
                                  84       TRANSACT CODE=SKH1,PRTY=(8,8)
                                  85      APPLCTN   PSB=HHTASK42
                                  86       TRANSACT CODE=SKH2,PRTY=(8,8)
                                  87      APPLCTN   PSB=HHTASK43
                                  88       TRANSACT CODE=SKH3,PRTY=(8,8)
                                  89      APPLCTN   PSB=HIBASK41,PGMTYPE=BATCH
                                  90      APPLCTN   PSB=HITASK41
                                  91       TRANSACT CODE=SKI1,PRTY=(8,8)
                                  92      APPLCTN   PSB=HITASK42
                                  93       TRANSACT CODE=SKI2,PRTY=(8,8)
                                  94      APPLCTN   PSB=HSBASK41,PGMTYPE=BATCH
                                  95      APPLCTN   PSB=HIMAJC01
                                  96       TRANSACT CODE=TPPL1,PRTY=(8,8)
                                  97      APPLCTN   PSB=HIMAJC02
                                  98       TRANSACT CODE=TPPL2,PRTY=(8,8)
                                  99      APPLCTN   PSB=HIMAJC03
                                 100       TRANSACT CODE=TUBE,PRTY=(8,8),SPA=100,MODE=SNGL
                                 101      APPLCTN   PSB=HIMALM01
                                 102       TRANSACT CODE=DLI,PRTY=(5,10,5),PROCLIM=(10,10),                 C
                                                    MSGTYPE=(SNGLSEG,RESPONSE)
                                 103       TRANSACT CODE=ICS,PRTY=(5,12,5),PROCLIM=(10,100),MODE=SNGL
                                 104       TRANSACT CODE=IMS,PRTY=(2,5,10),PROCLIM=(1,100),MSGTYPE=SNGLSEG
                                 105       TRANSACT CODE=DLN,PRTY=(0,8,3),PROCLIM=(10,100)
                                 106      APPLCTN   PSB=LKMDFS00
                                 107       TRANSACT CODE=DFS,PRTY=(5,12,5),PROCLIM=(8,100),INQ=YES
                                 108      APPLCTN   PSB=LKMDFS10,PGMTYPE=BATCH
                                 109       TRANSACT CODE=LKM,PRTY=(0,0),MSGTYPE=SNGLSEG
```



```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                    F01OCT71  10/02/72

                                 111      APPLCTN   PSB=MR1
                                 112       TRANSACT CODE=MR1,PRTY=(8,8)
                                 113      APPLCTN   PSB=MR2
                                 114       TRANSACT CODE=MR2,PRTY=(8,8)
                                 115      APPLCTN   PSB=MR3
                                 116       TRANSACT CODE=MR3,PRTY=(8,8)
                                 117      APPLCTN   PSB=MR4
                                 118       TRANSACT CODE=MR4,PRTY=(8,8)
                                 119      APPLCTN   PSB=MR5
                                 120       TRANSACT CODE=MR5,PRTY=(8,8)
                                 121      APPLCTN   PSB=MR6
                                 122       TRANSACT CODE=MR6,PRTY=(8,8)
                                 123      APPLCTN   PSB=MR7
                                 124       TRANSACT CODE=MR7,PRTY=(8,8)
                                 125      APPLCTN   PSB=MR8
                                 126       TRANSACT CODE=MR8,PRTY=(8,8)
                                 127      APPLCTN   PSB=MR9
                                 128       TRANSACT CODE=MR9,PRTY=(8,8)
```



```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                    F01OCT71  10/02/72

                                 130 *
                                 131 *            DEFINE COMMUNICATIONS NETWORK
                                 132 *

                                 134 *        *    *    *    *    *    *    *    *    *    *    *
                                 135 *        * NONSWITCHED AND POINT-TO-POINT 2740 LINE GROUPS *
                                 136 *        *    *    *    *    *    *    *    *    *    *    *

                                 138              LINEGRP      DDNAME=DD2740S,UNITYPE=2740
                                 139                LINE       ADDR=0C2
                                 140                  TERMINAL ADDR=C6
                                 141                    NAME   (CTRL,MASTER)
                                 142                    NAME   HOWARD
                                 143                  TERMINAL ADDR=E2
                                 144                    NAME   LARRY
                                 145                LINE       ADDR=0C3,MODEL=(2,440)
                                 146                  TERMINAL ADDR=45,BUFSIZE=248
                                 147                    NAME   MODEL2
                                 148                  TERMINAL ADDR=46
                                 149                    NAME   MODEL2K

                                 151              LINEGRP      DDNAME=D2740NSC,UNITYPE=(2740,NOSTACTL)
                                 152                LINE       ADDR=0C1
                                 153                  TERMINAL ADDR=E2
                                 154                    NAME   T2740NSC
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                        F01OCT71  10/02/72

                                 156 *         *    *    *    *    *    *    *                          00146010
                                 157 *         *    REMOTE 3270 LINE GROUP    *                          00146020
                                 158 *         *    *    *    *    *    *    *                          00146030

                                 160                LINEGRP      DDNAME=DD3270R,UNITYPE=3270,CODE=EBCDIC   00146050
                                 161                 LINE        ADDR=0C9                                  00146060
                                 162                  TERMINAL  ADDR=4040,UNIT=3275,COMPT=PTR1,          X00146070
                                                    FEAT=(,COPY,PFK)                                       00146080
                                 163                   NAME      T3275                                     00146090
                                 164                    NAME     T3275P,COMPT=PTR1                         00146095
                                 165                 CTLUNIT     ADDR=C1,MODEL=2                           00146100
                                 166                  TERMINAL  ADDR=40,FEAT=(2,COPY,PFK,CARD,PEN)         00146110
                                 167                   NAME      T3270A                                    00146120
                                 168                  TERMINAL  ADDR=C1,FEAT=(2,COPY,PFK,CARD,PEN)         00146130
                                 169                   NAME      T3270B                                    00146140
                                 170                  TERMINAL  ADDR=C2,FEAT=(1,COPY,PFK,CARD,PEN)         00146150
                                 171                   NAME      T3270C                                    00146160
                                 172                  TERMINAL  ADDR=C3,UNIT=3284,FEAT=2,BUFSIZE=132       00146170
                                 173                   NAME      T3270P1                                   00146180
                                 174                  TERMINAL ADDR=C4,UNIT=3286,FEAT=2,BUFSIZE=132        00146190
                                 175                   NAME      T3270P2                                   00146200


                                 177 *        *    *    *    *    *    *    *                            00146220
                                 178 *        *    LOCAL 3270 LINE GROUP    *                            00146230
                                 179 *        *    *    *    *    *    *    *                            00146240

                                 181                LINEGRP DDNAME=DD3270L,UNITYPE=(3270,LOCAL)            00146260
                                 182                 LINE        BUFSIZE=300                               00146270
                                 183                  TERMINAL  ADDR=310,FEAT=2                            00146280
                                 184                   NAME      T3270L1                                   00146290
                                 185                   TERMINAL ADDR=312,FEAT=(1,,PFK,CARD,PEN)            00146300
                                 186                   NAME      T3270L2                                   00146310

                                 188                LINEGRP DDNAME=DD3270L2,UNITYPE=(3270,LOCAL)           00146318
                                 189                LINE
                                 190                  TERMINAL  ADDR=311,FEAT=2,UNIT=3286                  00146325
                                 191                   NAME      T3270P3                                   00146330
```

PAGE 7

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                        F01OCT71  10/02/72

                                 193 *            *    *    *    *    *    *    *
                                 194 *            * NONSWITCHED 1050 LINE GROUP *
                                 195 *            *    *    *    *    *    *    *

                                 197                LINEGRP      DDNAME=DD1050,UNITYPE=1050
                                 198                  LINE       ADDR=0C0
                                 199                     TERMINAL ADDR=E2,FEAT=1052,COMPT=PTR1
                                 200                        NAME    PRINTER1,COMPT=PTR1
                                 201                     TERMINAL ADDR=E4,FEAT=1052,COMPT=(PTR1,PTR2)
                                 202                        NAME    PRINTER2,COMPT=PTR1
                                 203                        NAME    PRINTER3,COMPT=PTR2

                                 205 *            *    *    *    *    *    *
                                 206 *            * REMOTE 2260 LINE GROUP *
                                 207 *            *    *    *    *    *    *

                                 209                LINEGRP      DDNAME=DD2260R,UNITYPE=2260
                                 210                  LINE       ADDR=0E0
                                 211                     TERMINAL ADDR=4151,FEAT=(2265,2,WLA),UNIT=2845
                                 212                        NAME    T2265XCL
                                 213                   CTLUNIT    ADDR=40,WLA=(YES,1053),MODEL=3
                                 214                     TERMINAL ADDR=A0
                                 215                        NAME    AA
                                 216                     TERMINAL ADDR=A1
                                 217                        NAME    AB
                                 218                     TERMINAL ADDR=A4,UNIT=1053
                                 219                        NAME    ERNE
```

PAGE 8

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                        F01OCT71  10/02/72

                                 221 *            *    *    *    *    *    *    *    *    *
                                 222 *            *       2770 LINE GROUP                *
                                 223 *            *    *    *    *    *    *    *    *    *

                                 225                LINEGRP     DDNAME=DD2770,UNITYPE=2770,CODE=EBCDIC
                                 226                 LINE        ADDR=0C8,FEAT=BUFEXP
                                 227                   TERMINAL ADDR=C1,FEAT=(,BUFEXP),COMPT=(PTR,VIDEO,CARD)
                                 228                        NAME T2770P,COMPT=1
                                 229                        NAME T2770V,COMPT=2
                                 230                        NAME T2770C,COMPT=3

                                 232 *            *    *    *    *    *    *    *
                                 233 *            * MULTI-POINT 2980 LINE GROUP *
                                 234 *            *    *    *    *    *    *    *

                                 236                LINEGRP      DDNAME=DD2980,UNITYPE=2980
                                 237                  LINE       ADDR=0CA
                                 238                   CTLUNIT   ADDR=C1
                                 239                     TERMINAL ADDR=40,FEAT=(1,4)
                                 240                       NAME    HOMER1
                                 241                       NAME    N29801
                                 242                       NAME    A29801
                                 243                    TERMINAL ADDR=F1,FEAT=2
                                 244                       NAME    N29802
                                 245                       NAME    A29802
                                 246                       NAME    COMMON
                                 247                       NAME    HOMER4,COMPT=COMMON
                                 248                    TERMINAL ADDR=F2,FEAT=(4,2)
                                 249                       NAME    ATELLER
                                 250                       NAME    BTELLER
                                 251                       NAME    JTELLER      TELLER A WITH SUPERVISOR KEY
                                 252                       NAME    KTELLER      TELLER B WITH SUPERVISOR KEY
                                 253                       NAME    HOMER2
```

```
                                                                                                  PAGE    9

LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                        F01OCT71  10/02/72
                                  255 *             *     *     *     *   *
                                  256 *             * 2741 LINE GROUPS *
                                  257 *             *     *     *     *   *

                                  259              LINEGRP        DDNAME=DD2741,UNITYPE=2741,CODE=CORRS
                                  260                  LINE       ADDR=097
                                  261                    TERMINAL ADDR=C1,FEAT=NOINTRPT
                                  262                      NAME    T2741

                                  264              LINEGRP        DDNAME=DD2741S,UNITYPE=2741,                C
                                                     CODE=(CORRS,EBCDIC),FEAT=SWITCHED
                                  265                  LINE       ADDR=050
                                  266                    TERMINAL ADDR=50

                                  268              LINEGRP        DDNAME=DD2741S2,UNITYPE=2741,               C
                                                     CODE=(CORRS,BCD),FEAT=SWITCHED
                                  269                  LINE       ADDR=053
                                  270                    TERMINAL ADDR=53

                                  272 *             *    *     *     *    *
                                  273 *             * 1030 LINE GROUP *
                                  274 *             *    *     *     *    *

                                  276                LINEGRP      DDNAME=DD1030,UNITYPE=1030
                                  277                  LINE       ADDR=0E1,BUFSIZE=182
                                  278 TERMA              TERMINAL ADDR=64,UNIT=1033
                                  279                      NAME    T1033A
                                  280                      NAME    T1C33B
                                  281                    TERMINAL ADDR=64,LTERM=T1033A
```

```
                                                                                                  PAGE   10

LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                        F01OCT71  10/02/72
                                  283 *               *    *    *    *    *    *    *    *
                                  284 *               * LOCAL SYSOUT LINE GROUPS *
                                  285 *               *    *    *    *    *    *    *    *

                                  287                LINEGRP   DDNAME=SYSRDR1,UNITYPE=READER
                                  288                LINE      ADDR=CB1
                                  289                TERMINAL  LTERM=T1033A

                                  291                LINEGRP   DDNAME=SYSPRT1,UNITYPE=PRINTER
                                  292                LINE      ADDR=CB2
                                  293                TERMINAL
                                  294                NAME      PRTSYS

                                  296                LINEGRP   DDNAME=SYSTAPE,UNITYPE=TAPE
                                  297                LINE      ADDR=085,BUFSIZE=220
                                  298                TERMINAL
                                  299                NAME      TAPESYS

                                  301                LINEGRP   UNITYPE=SPOOL,DDNAME=(SYS1,SYS2,SYS3,SYS4)
                                  302                LINE      BUFSIZE=880
                                  303                TERMINAL  FEAT=AUTOSCH
                                  304                NAME      SP1
                                  305                NAME      SP2
```

```
                                                                                                  PAGE   11

LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                        F01OCT71  10/02/72
                                  307 *             *    *    *    *    *    *    *    *
                                  308 *             * 777C SWITCHED LINE GROUP *
                                  309 *             *    *    *    *    *    *    *    *

                                  311              LINEGRP       DDNAME=ARUDDGP1,UNITYPE=777C,OUTTBL=OUT777C
                                  312                LINE        ADDR=0F0,BUFSIZE=(246,246),FEAT=(ABB,REPEAT)
                                  313                  TERMINAL ADDR=C1
                                  314                LINE        ADDR=0F3,BUFSIZE=(50,200),FEAT=(ABC,NOREPEAT)
                                  315                  TERMINAL ADDR=C4

                                  317 *             *    *    *    *    *    *    *
                                  318 *             * TWX SWITCHED LINE GROUP *
                                  319 *             *    *    *    *    *    *    *

                                  321              LINEGRP       DDNAME=DTWX3335,UNITYPE=TWX,FEAT=SWITCHED
                                  322                LINE        ADDR=CD1
                                  323                  TERMINAL ADDR=E2

                                  325 *             *    *    *    *    *    *    *
                                  326 *             * 2740 SWITCHED LINE GROUP *
                                  327 *             *    *    *    *    *    *    *

                                  329              LINEGRP       DDNAME=DD2740A,UNITYPE=2740,FEAT=SWITCHED
                                  330                LINE        ADDR=061
                                  331                  TERMINAL ADDR=E2
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  F01OCT71  10/02/72

                                   333 *              *   *    *    *    *    *    *    *
                                   334 *              * 1050 SWITCHED LINE GROUP *
                                   335 *              *   *    *    *    *    *    *    *

                                   337                  LINEGRP       DDNAME=DD1050A,UNITYPE=1050,FEAT=SWITCHED
                                   338                    LINE        ADDR=060
                                   339                      TERMINAL ADDR=E2

                                   341                    POOL
                                   342                      SUBPOOL
                                   343                         NAME  ELEANOR
                                   344                         NAME  CAROL
                                   345                         NAME  SHARRON
                                   346                      SUBPOOL
                                   347                         NAME  T2741N1
                                   348                         NAME  T2741N2
                                   349                      SUBPOOL
                                   350                         NAME  T2741N3
                                   351                         NAME  T2741N4
                                   352                      SUBPOOL
                                   353                         NAME  SW1050
                                   354                         NAME  SWPRNTR2,COMPT=3
                                   355                         NAME  SWTPPNCH,COMPT=3
                                   356                         NAME  SWCDPNCH,COMPT=4


                                   358            IMSGEN ASMPRT=ON,PROCLIB=(YES,8),                                  C
                                                        LKPRT=(XREF,LIST),SUFFIX=B,                                  C
                                                        OBJDSET=IMS2.BLKS50,                                         C
                                                        USERLIB=ICS.CLOD,                                            C
                                                        PRTY=8,                                                      C
                                                        ASM=H,                                                       C
                                                        PAGE=YES,                                                    C
                                                        JOBCTL=(4,D,A,(82C,6443))
                                   359+           TITLE 'IMS SYSTEM DEFINITION.                                      X
```



```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  F01OCT71  10/02/72
                                   360                 *,  OS SYSTEM OPTIONS:

                                   362                 *,        MVT SYSTEM
                                   363                 *,        MCS OPTIONS:  ROUTCDE=13
                                   364                 *,                      DESC=2
                                   365                 *,        RESV'D SVC'S (234,248,213)
                                   366                 *,        RESV'D APNDG (IGG019Z5,IGG019Z6)
                                   367                 *,        RESV'D SQS    (500,2000)

                                   369                 *,  IMS/360 SYSTEM OPTIONS:

                                   371                 *,        5 REGIONS
                                   372                 *,        15 SUBTASKS
                                   373                 *,        23 CONCURRENT EXCTL REQUESTS
                                   374                 *,        2500 CHECKPOINT LOG FREQUENCY
                                   375                 *,        GENERATE FOR ALL
                                   376                 *,        REGION JCL:
                                   377                 *,             CLASS=A,MSGCLASS=3,REGION=30K

                                   379                 *,  SPA OPTIONS:

                                   381                 *,        CORE:    NUMBER=3
                                   382                 *,                 SIZE=100

                                   384                 *,        DASD:    NUMBER=6
                                   385                 *,                 SIZE=150

                                   387                 *,  BUFFER POOLS:

                                   389                 *,        PSB      - SIZE=8000
                                   390                 *,        DMB      - SIZE=10000
                                   391                 *,        DBASE    - SIZE=11000
                                   392                 *,        GENERAL  - SIZE=5000
                                   393                 *,        COMM.    - SIZE=10600

                                   395                 *,        FORMAT - SIZE=13000
                                   396                 *,        FRE  - NUMBER=13

                                   398                 *,  MESSAGE QUEUE OPTIONS:

                                   400                 *,        DATA SETS:  IMS2.QBLKS  DEVICE=2314
                                   401                 *,                    IMS2.SHMSG  DEVICE=2314
                                   402                 *,                    IMS2.LGMSG  DEVICE=2314

                                   404                 *,        SHUTDOWN MARGIN IS 30 RECORDS

                                   406                 *,        SHORT MESSAGE LENGTH IS 192 BYTES
                                   407                 *,        LONG MESSAGE LENGTH IS 576 BYTES
                                   408                 *,        BUFFERS:    NUMBER=13
                                   409                 *,                    SIZE=576
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                               F010CT71  10/02/72

                                   411                 *,  APPLICATION SPECIFICATIONS:

                                   413                 *,       PSB=DFSSAM02  PGMTYPE=TP
                                   414                 *,                OPTION=N/A

                                   416                 *,           TRANSACTION CODE - PART
                                   417                 *,               MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                   418                 *,               PRTY=(7,10,2)
                                   419                 *,               PROCLIM=(65535,65535)
                                   420                 *,               CLASS=4
                                   421                 *,               EDIT=(UC)

                                   423                 *,       PSB=DFSSAM03  PGMTYPE=TP
                                   424                 *,                OPTION=N/A

                                   426                 *,           TRANSACTION CODE - DSPINV
                                   427                 *,               MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                   428                 *,               PRTY=(7,10,2)
                                   429                 *,               PROCLIM=(65535,65535)
                                   430                 *,               CLASS=4
                                   431                 *,               EDIT=(UC)

                                   433                 *,       PSB=DFSSAM04  PGMTYPE=TP
                                   434                 *,                OPTION=N/A

                                   436                 *,           TRANSACTION CODE - ADDPART
                                   437                 *,               MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                   438                 *,               PRTY=(7,10,2)
                                   439                 *,               PROCLIM=(65535,65535)
                                   440                 *,               CLASS=4
                                   441                 *,               EDIT=(UC)
                                   442                 *,           TRANSACTION CODE - ADDINV
                                   443                 *,               MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                   444                 *,               PRTY=(7,10,2)
                                   445                 *,               PROCLIM=(65535,65535)
                                   446                 *,               CLASS=5
                                   447                 *,               EDIT=(UC)
                                   448                 *,           TRANSACTION CODE - DLETPART
                                   449                 *,               MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                   450                 *,               PRTY=(7,10,2)
                                   451                 *,               PROCLIM=(65535,65535)
                                   452                 *,               CLASS=4
                                   453                 *,               EDIT=(UC)
                                   454                 *,           TRANSACTION CODE - DLETINV
                                   455                 *,               MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                   456                 *,               PRTY=(7,10,2)
                                   457                 *,               PROCLIM=(65535,65535)
                                   458                 *,               CLASS=5
                                   459                 *,               EDIT=(UC)

                                   461                 *,       PSB=DFSSAM05  PGMTYPE=TP
                                   462                 *,                OPTION=N/A

                                   464                 *,           TRANSACTION CODE - CLOSE
                                   465                 *,               MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                      FC10CT71  10/02/72
                                  466           *,           PRTY=(7,10,2)
                                  467           *,           PROCLIM=(65535,65535)
                                  468           *,           CLASS=3
                                  469           *,           EDIT=(UC)

                                  471           *,       PSB=DFSSAM06  PGMTYPE=TP
                                  472           *,               OPTION=N/A

                                  474           *,         TRANSACTION CODE - DISBURSE
                                  475           *,           MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                  476           *,           PRTY=(7,10,2)
                                  477           *,           PROCLIM=(65535,65535)
                                  478           *,           CLASS=2
                                  479           *,           EDIT=(UC)

                                  481           *,       PSB=DFSSAM07  PGMTYPE=TP
                                  482           *,               OPTION=N/A

                                  484           *,         TRANSACTION CODE - DSPALLI
                                  485           *,           MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                  486           *,           PRTY=(7,10,2)
                                  487           *,           PROCLIM=(65535,65535)
                                  488           *,           CLASS=2
                                  489           *,           EDIT=(UC)

                                  491           *,       PSB=CONTEST   PGMTYPE=TP
                                  492           *,               OPTION=N/A

                                  494           *,         TRANSACTION CODE - CONV1
                                  495           *,           MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=SNGL
                                  496           *,           PRTY=(8,8,65535)
                                  497           *,           PROCLIM=(3,60)
                                  498           *,           EDIT=(UC)
                                  499           *,          ************ CONVERSATIONAL ************
                                  500           *,          *                                      *
                                  501           *,                    SPA=(80,DISK)
                                  502           *,          *                                      *
                                  503           *,          ****************************************
                                  504           *,         TRANSACTION CODE - CONV2
                                  505           *,           MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=SNGL
                                  506           *,           PRTY=(8,8,65535)
                                  507           *,           PROCLIM=(3,60)
                                  508           *,           EDIT=(UC)
                                  509           *,          ************ CONVERSATIONAL ************
                                  510           *,          *                                      *
                                  511           *,                    SPA=(80,CORE)
                                  512           *,          *                                      *
                                  513           *,          ****************************************

                                  515           *,       PSB=ENQDSK41  PGMTYPE=TP
                                  516           *,               OPTION=N/A

                                  518           *,         TRANSACTION CODE - ENQ
                                  519           *,           MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                  520           *,           PRTY=(8,8,65535)
```

```
LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F01OCT71  10/02/72
                                   521             *,          PROCLIM=(65535,65535)
                                   522             *,          EDIT=(UC)

                                   524             *,      PSB=HDTASK01   PGMTYPE=TP
                                   525             *,               OPTION=N/A

                                   527             *,        TRANSACTION CODE - SKD1
                                   528             *,          MSGTYPE=(MULTSEG,NONRESPONSE)   MODE=MULT
                                   529             *,          PRTY=(8,8,65535)
                                   530             *,          PROCLIM=(65535,65535)
                                   531             *,          EDIT=(UC)

                                   533             *,      PSB=HHBLSK41   PGMTYPE=BATCH
                                   534             *,               OPTION=N/A


                                   537             *,      PSB=HHBLSK42   PGMTYPE=BATCH
                                   538             *,               OPTION=N/A


                                   541             *,      PSB=HHBLSK43   PGMTYPE=BATCH
                                   542             *,               OPTION=N/A


                                   545             *,      PSB=HHTASK41   PGMTYPE=TP
                                   546             *,               OPTION=N/A

                                   548             *,        TRANSACTION CODE - SKH1
                                   549             *,          MSGTYPE=(MULTSEG,NONRESPONSE)   MODE=MULT
                                   550             *,          PRTY=(8,8,65535)
                                   551             *,          PROCLIM=(65535,65535)
                                   552             *,          EDIT=(UC)

                                   554             *,      PSB=HHTASK42   PGMTYPE=TP
                                   555             *,               OPTION=N/A

                                   557             *,        TRANSACTION CODE - SKH2
                                   558             *,          MSGTYPE=(MULTSEG,NONRESPONSE)   MODE=MULT
                                   559             *,          PRTY=(8,6,65535)
                                   560             *,          PROCLIM=(65535,65535)
                                   561             *,          EDIT=(UC)

                                   563             *,      PSB=HHTASK43   PGMTYPE=TP
                                   564             *,               OPTION=N/A

                                   566             *,        TRANSACTION CODE - SKH3
                                   567             *,          MSGTYPE=(MULTSEG,NONRESPONSE)   MODE=MULT
                                   568             *,          PRTY=(8,8,65535)
                                   569             *,          PROCLIM=(65535,65535)
                                   570             *,          EDIT=(UC)

                                   572             *,      PSB=HIBASK41   PGMTYPE=BATCH
                                   573             *,               OPTION=N/A
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                   F010CT71  10/02/72
                                  576            *,        PSB=HITASK41  PGMTYPE=TP
                                  577            *,                OPTION=N/A

                                  579            *,           TRANSACTION CODE - SKI1
                                  580            *,              MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                  581            *,              PRTY=(8,8,65535)
                                  582            *,              PROCLIM=(65535,65535)
                                  583            *,              EDIT=(UC)

                                  585            *,        PSB=HITASK42  PGMTYPE=TP
                                  586            *,                OPTION=N/A

                                  588            *,           TRANSACTION CODE - SKI2
                                  589            *,              MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                  590            *,              PRTY=(8,8,65535)
                                  591            *,              PROCLIM=(65535,65535)
                                  592            *,              EDIT=(UC)

                                  594            *,        PSB=HSBASK41  PGMTYPE=BATCH
                                  595            *,                OPTION=N/A


                                  598            *,        PSB=HIMAJC01  PGMTYPE=TP
                                  599            *,                OPTION=N/A

                                  601            *,           TRANSACTION CODE - TPPL1
                                  602            *,              MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                  603            *,              PRTY=(8,8,65535)
                                  604            *,              PROCLIM=(65535,65535)
                                  605            *,              EDIT=(UC)

                                  607            *,        PSB=HIMAJC02  PGMTYPE=TP
                                  608            *,                OPTION=N/A

                                  610            *,           TRANSACTION CODE - TPPL2
                                  611            *,              MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                  612            *,              PRTY=(8,8,65535)
                                  613            *,              PROCLIM=(65535,65535)
                                  614            *,              EDIT=(UC)

                                  616            *,        PSB=HIMAJC03  PGMTYPE=TP
                                  617            *,                OPTION=N/A

                                  619            *,           TRANSACTION CODE - TUBE
                                  620            *,              MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=SNGL
                                  621            *,              PRTY=(8,8,65535)
                                  622            *,              PROCLIM=(65535,65535)
                                  623            *,              EDIT=(UC)
                                  624            *,             ************ CONVERSATIONAL ************
                                  625            *,             *                                      *
                                  626            *,                          SPA=(100,DISK)
                                  627            *,             *                                      *
                                  628            *,             ****************************************

                                  630            *,        PSB=HIMALM01  PGMTYPE=TP
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT    SOURCE STATEMENT                                              F01OCT71  10/02/72
                                  631               *,                OPTION=N/A

                                  633               *,          TRANSACTION CODE - DLI
                                  634               *,             MSGTYPE=(SNGLSEG,RESPONSE)  MODE=MULT
                                  635               *,             PRTY=(5,10,5)
                                  636               *,             PROCLIM=(10,10)
                                  637               *,             EDIT=(UC)
                                  638               *,          TRANSACTION CODE - ICS
                                  639               *,             MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=SNGL
                                  640               *,             PRTY=(5,12,5)
                                  641               *,             PROCLIM=(10,100)
                                  642               *,             EDIT=(UC)
                                  643               *,          TRANSACTION CODE - IMS
                                  644               *,             MSGTYPE=(SNGLSEG,NONRESPONSE)  MODE=MULT
                                  645               *,             PRTY=(2,5,10)
                                  646               *,             PROCLIM=(1,100)
                                  647               *,             EDIT=(UC)
                                  648               *,          TRANSACTION CODE - DLN
                                  649               *,             MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                  650               *,             PRTY=(0,8,3)
                                  651               *,             PROCLIM=(10,100)
                                  652               *,             EDIT=(UC)

                                  654               *,       PSB=LKMDFS00  PGMTYPE=TP
                                  655               *,                OPTION=N/A

                                  657               *,          TRANSACTION CODE - DFS
                                  658               *,             MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                  659               *,             PRTY=(5,12,5)
                                  660               *,             PROCLIM=(8,100)
                                  661               *,             EDIT=(UC)

                                  663               *,       PSB=LKMDFS10  PGMTYPE=BATCH
                                  664               *,                OPTION=N/A

                                  666               *,          TRANSACTION CODE - LKM
                                  667               *,             MSGTYPE=(SNGLSEG,NONRESPONSE)  MODE=MULT
                                  668               *,             PRTY=(0,0,65535)
                                  669               *,             PROCLIM=(65535,65535)
                                  670               *,             EDIT=(UC)

                                  672               *,       PSB=MR1        PGMTYPE=TP
                                  673               *,                OPTION=N/A

                                  675               *,          TRANSACTION CODE - MR1
                                  676               *,             MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
                                  677               *,             PRTY=(8,8,65535)
                                  678               *,             PROCLIM=(65535,65535)
                                  679               *,             EDIT=(UC)

                                  681               *,       PSB=MR2        PGMTYPE=TP
                                  682               *,                OPTION=N/A

                                  684               *,          TRANSACTION CODE - MR2
                                  685               *,             MSGTYPE=(MULTSEG,NONRESPONSE)  MODE=MULT
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71  10/02/72
                                  686            *,           PRTY=(8,8,65535)
                                  687            *,           PROCLIM=(65535,65535)
                                  688            *,           EDIT=(UC)

                                  690            *,      PSB=MR3        PGMTYPE=TP
                                  691            *,                OPTION=N/A

                                  693            *,         TRANSACTION CODE - MR3
                                  694            *,           MSGTYPE=(MULTSEG,NONRESPONSE)   MODE=MULT
                                  695            *,           PRTY=(8,8,65535)
                                  696            *,           PROCLIM=(65535,65535)
                                  697            *,           EDIT=(UC)

                                  699            *,      PSB=MR4        PGMTYPE=TP
                                  700            *,                OPTION=N/A

                                  702            *,         TRANSACTION CODE - MR4
                                  703            *,           MSGTYPE=(MULTSEG,NONRESPONSE)   MODE=MULT
                                  704            *,           PRTY=(8,8,65535)
                                  705            *,           PROCLIM=(65535,65535)
                                  706            *,           EDIT=(UC)

                                  708            *,      PSB=MR5        PGMTYPE=TP
                                  709            *,                OPTION=N/A

                                  711            *,         TRANSACTION CODE - MR5
                                  712            *,           MSGTYPE=(MULTSEG,NONRESPONSE)   MODE=MULT
                                  713            *,           PRTY=(8,8,65535)
                                  714            *,           PROCLIM=(65535,65535)
                                  715            *,           EDIT=(UC)

                                  717            *,      PSB=MR6        PGMTYPE=TP
                                  718            *,                OPTION=N/A

                                  720            *,         TRANSACTION CODE - MR6
                                  721            *,           MSGTYPE=(MULTSEG,NONRESPONSE)   MODE=MULT
                                  722            *,           PRTY=(8,8,65535)
                                  723            *,           PROCLIM=(65535,65535)
                                  724            *,           EDIT=(UC)

                                  726            *,      PSB=MR7        PGMTYPE=TP
                                  727            *,                OPTION=N/A

                                  729            *,         TRANSACTION CODE - MR7
                                  730            *,           MSGTYPE=(MULTSEG,NONRESPONSE)   MODE=MULT
                                  731            *,           PRTY=(8,8,65535)
                                  732            *,           PROCLIM=(65535,65535)
                                  733            *,           EDIT=(UC)

                                  735            *,      PSB=MR8        PGMTYPE=TP
                                  736            *,                OPTION=N/A

                                  738            *,         TRANSACTION CODE - MR8
                                  739            *,           MSGTYPE=(MULTSEG,NONRESPONSE)   MODE=MULT
                                  740            *,           PRTY=(8,8,65535)
```



```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71  10/02/72
                                  741            *,           PROCLIM=(65535,65535)
                                  742            *,           EDIT=(UC)

                                  744            *,      PSB=MR9        PGMTYPE=TP
                                  745            *,                OPTION=N/A

                                  747            *,         TRANSACTION CODE - MR9
                                  748            *,           MSGTYPE=(MULTSEG,NONRESPONSE)   MODE=MULT.
                                  749            *,           PRTY=(8,8,65535)
                                  750            *,           PROCLIM=(65535,65535)
                                  751            *,           EDIT=(UC)


                                  753            *,          ****************************************
                                  754            *,          *                                      *
                                  755            *,          *    THE FOLLOWING TRANSACTION CODES   *
                                  756            *,          *    ARE DESIGNATED AS INQUIRY ONLY:   *
                                  757            *,          *                                      *
                                  758            *,          *              PART                    *
                                  759            *,          *              DSPINV                  *
                                  760            *,          *              DSPALLI                 *
                                  761            *,          *              DFS                     *
                                  762            *,          *                                      *
                                  763            *,          ****************************************
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F01OCT71  10/02/72
                                  765               *,  DATABASE SPECIFICATIONS:
                                  767               *,                    DBD=DI31LM01
                                  769               *,                    DBD=DI31PHC2
                                  771               *,                    DBD=DI21PART
                                  773               *,                    DBD=DS40JC01
                                  775               *,                    DBD=DI41SK01
                                  777               *,                    DBD=DI42SK01
                                  779               *,                    DBD=DH41SK01
                                  781               *,                    DBD=DH41SK02
                                  783               *,                    DBD=DH41SK03
                                  785               *,                    DBD=DX41SK01
                                  787               *,                    DBD=DD41SK01
                                  789               *,                    DBD=DH41TS01
                                  791               *,                    DBD=DH41TSC2
                                  793               *,                    DBD=DH41TSC3
                                  795               *,                    DBD=DX41TS01
                                  797               *,                    DBD=DD41TSC1
```



```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F01OCT71  10/02/72
                                  799               *,  COMMUNICATION SPECIFICATIONS:
                                  801               *,         LINE 1     SYSTEM/360 OPERATOR'S CONSOLE.
                                  803               *,                      TERMINAL 1    ADDR=N/A  EDIT=ULC
                                  804               *,                        WTOR        EDIT=(NO,ULC)

                                  807               *,     LINEGRP 1  DDNAME=DD2740S  UNITYPE=2740
                                  808               *,                  FEAT=(NONSWITCHED,POLL)

                                  810               *,        LINE 2  ADDR=0C2
                                  811               *,                  TYPLIST=WRAPLST

                                  814               *,                TERMINAL 1   ADDR=C6  EDIT=NO

                                  816               *,                ********** MASTER TERMINAL **********
                                  817               *,                **                                 **
                                  818               *,                  CTRL       EDIT=(NO,ULC)
                                  819               *,                **                                 **
                                  820               *,                *************************************
                                  821               *,                  HOWARD     EDIT=(NO,ULC)

                                  823               *,                TERMINAL 2   ADDR=E2   EDIT=NO
                                  824               *,                  LARRY      EDIT=(NO,ULC)

                                  827               *,        LINE 3  ADDR=0C3
                                  828               *,                  TYPLIST=WRAPLST

                                  831               *,                TERMINAL 1   ADDR=45   EDIT=NO
                                  832               *,                  MODEL2     EDIT=(NO,ULC)

                                  834               *,                TERMINAL 2   ADDR=46   EDIT=NO
                                  835               *,                  MODEL2K    EDIT=(NO,ULC)

                                  839               *,     LINEGRP 2  DDNAME=D2740NSC   UNITYPE=2740,NSC
                                  840               *,                  FEAT=(NONSWITCHED,POLL)

                                  842               *,        LINE 4  ADDR=0C1
                                  843               *,                  TYPLIST=N/A

                                  846               *,                TERMINAL 1   ADDR=E2   EDIT=NO
                                  847               *,                  T2740NSC   EDIT=(NO,ULC)

                                  851               *,     LINEGRP 3  DDNAME=DD3270R  UNITYPE=3270
                                  852               *,                  FEAT=(NONSWITCHED,AUTOPOLL)
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                         F01OCT71  10/02/72
                                   854                *,        LINE 5  ADDR=0C9
                                   855                *,                 TYPLIST=AUTOWLST

                                   858                *,                 TERMINAL 1   ADDR=4040  EDIT=NO
                                   859                *,                   T3275     EDIT=(NO,ULC)
                                   860                *,                   T3275P    EDIT=(NO,ULC)  COMPT=PTR1

                                   862                *,             CTLUNIT    ADDR=C1

                                   864                *,                 TERMINAL 2   ADDR=40  EDIT=NO
                                   865                *,                   T3270A    EDIT=(NO,ULC)

                                   867                *,                 TERMINAL 3   ADDR=C1  EDIT=NO
                                   868                *,                   T3270B    EDIT=(NO,ULC)

                                   870                *,                 TERMINAL 4   ADDR=C2  EDIT=NO
                                   871                *,                   T3270C    EDIT=(NO,ULC)

                                   873                *,                 TERMINAL 5   ADDR=C3  EDIT=NO
                                   874                *,                   T3270P1   EDIT=(NO,ULC)

                                   876                *,                 TERMINAL 6   ADDR=C4  EDIT=NO
                                   877                *,                   T3270P2   EDIT=(NO,ULC)

                                   881                *,    LINEGRP 4  DDNAME=DD3270L  UNITYPE=3270,LOCAL

                                   883                *,        LINE 6  ADDR=N/A
                                   884                *,                 TYPLIST=N/A

                                   887                *,                 TERMINAL 1   ADDR=310  EDIT=NO
                                   888                *,                   T3270L1   EDIT=(NO,ULC)

                                   890                *,                 TERMINAL 2   ADDR=312  EDIT=NO
                                   891                *,                   T3270L2   EDIT=(NO,ULC)

                                   895                *,    LINEGRP 5  DDNAME=DD3270L2  UNITYPE=3270,LOCAL

                                   897                *,        LINE 7  ADDR=N/A
                                   898                *,                 TYPLIST=N/A

                                   901                *,                 TERMINAL 1   ADDR=311  EDIT=NO
                                   902                *,                   T3270P3   EDIT=(NO,ULC)

                                   906                *,    LINEGRP 6  DDNAME=DD1050  UNITYPE=1050
                                   907                *,                 FEAT=(NONSWITCHED,POLL)
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                      F01OCT71  10/02/72
                                 909              *,      LINE 8  ADDR=0C0
                                 910              *,                TYPLIST=WRAPLST

                                 913              *,              TERMINAL 1    ADDR=E2  EDIT=NO
                                 914              *,                PRINTER1  EDIT=(NO,ULC)  COMPT=PTR1

                                 916              *,              TERMINAL 2    ADDR=E4  EDIT=NO
                                 917              *,                PRINTER2  EDIT=(NO,ULC)  COMPT=PTR1
                                 918              *,                PRINTER3  EDIT=(NO,ULC)  COMPT=PTR2

                                 922              *,   LINEGRP 7  DDNAME=DD2260R  UNITYPE=2260
                                 923              *,                FEAT=(NONSWITCHED,POLL)

                                 925              *,      LINE 9  ADDR=0E0
                                 926              *,                TYPLIST=WRAPLST

                                 929              *,              TERMINAL 1    ADDR=4151  EDIT=NO
                                 930              *,                T2265XCL  EDIT=(NO,ULC)

                                 932              *,          CTLUNIT   ADDR=40  WLA=YES

                                 934              *,              TERMINAL 2    ADDR=A0  EDIT=NO
                                 935              *,                AA        EDIT=(NO,ULC)

                                 937              *,              TERMINAL 3    ADDR=A1  EDIT=NO
                                 938              *,                AB        EDIT=(NO,ULC)

                                 940              *,              TERMINAL 4    ADDR=A4  EDIT=NO
                                 941              *,                ERNE      EDIT=(NO,ULC)

                                 945              *,   LINEGRP 8  DDNAME=DD2770  UNITYPE=2770
                                 946              *,                FEAT=(NONSWITCHED,AUTOPOLL)

                                 948              *,      LINE 10  ADDR=0C8
                                 949              *,                TYPLIST=AUTOWLST

                                 952              *,              TERMINAL 1    ADDR=C1  EDIT=NO
                                 953              *,                T2770P    EDIT=(NO,ULC)  COMPT=1
                                 954              *,                T2770V    EDIT=(NO,ULC)  COMPT=2
                                 955              *,                T2770C    EDIT=(NO,ULC)  COMPT=3

                                 959              *,   LINEGRP 9  DDNAME=DD2980  UNITYPE=2980
                                 960              *,                FEAT=(NONSWITCHED,AUTOPOLL)

                                 962              *,      LINE 11  ADDR=0CA
                                 963              *,                TYPLIST=AUTOWLST
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  F01OCT71  10/02/72
                                  965              *,           CTLUNIT     ADDR=C1

                                  967              *,                TERMINAL 1   ADDR=40  EDIT=NO
                                  968              *,                  HOMER1     EDIT=(NO,UC)
                                  969              *,                  N29801     EDIT=(NO,UC)
                                  970              *,                  A29801     EDIT=(NO,UC)

                                  972              *,                TERMINAL 2   ADDR=F1  EDIT=NO
                                  973              *,                  N29802     EDIT=(NO,UC)
                                  974              *,                  A29802     EDIT=(NO,UC)
                                  975              *,                  COMMON     EDIT=(NO,UC)
                                  976              *,                  HOMER4     EDIT=(NO,UC)  COMPT=COMMON

                                  978              *,                TERMINAL 3   ADDR=F2  EDIT=NO
                                  979              *,                  ATELLER    EDIT=(NO,UC)
                                  980              *,                  BTELLER    EDIT=(NO,UC)
                                  981              *,                  JTELLER    EDIT=(NO,UC)
                                  982              *,                  KTELLER    EDIT=(NO,UC)
                                  983              *,                  HOMER2     EDIT=(NO,UC)


                                  987              *,     LINEGRP 10  DDNAME=DD2741  UNITYPE=2741
                                  988              *,                 FEAT=(NONSWITCHED,POLL)

                                  990              *,        LINE 12  ADDR=097
                                  991              *,                 TYPLIST=N/A


                                  994              *,                TERMINAL 1   ADDR=N/A  EDIT=NO
                                  995              *,                  T2741      EDIT=(NO,ULC)


                                  999              *,     LINEGRP 11  DDNAME=DD2741S  UNITYPE=2741
                                 1000              *,                 FEAT=(SWITCHED)

                                 1002              *,        LINE 13  ADDR=C50
                                 1003              *,                 TYPLIST=N/A


                                 1006              *,                TERMINAL 1   ADDR=50  EDIT=NO
                                 1007              *,                  INQU0393   EDIT=(NO,ULC)


                                 1011              *,     LINEGRP 12  DDNAME=DD2741S2  UNITYPE=2741
                                 1012              *,                 FEAT=(SWITCHED)

                                 1014              *,        LINE 14  ADDR=053
                                 1015              *,                 TYPLIST=N/A


                                 1018              *,                TERMINAL 1   ADDR=53  EDIT=NO
                                 1019              *,                  INQU0401   EDIT=(NO,ULC)
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT    SOURCE STATEMENT                                        F01OCT71  10/02/72
                                 1023                  *,    LINEGRP 13  DDNAME=DD1030  UNITYPE=1030
                                 1024                  *,                FEAT=(NONSWITCHED,POLL)

                                 1026                  *,       LINE 15  ADDR=0E1
                                 1027                  *,                TYPLIST=WRAPLST

                                 1030                  *,   TERMA       TERMINAL 1    ADDR=64  EDIT=NO
                                 1031                  *,                T1033A    EDIT=(NO,ULC)
                                 1032                  *,                T1033B    EDIT=(NO,ULC)

                                 1034                  *,               TERMINAL 2    ADDR=64  EDIT=NO
                                 1035                  *,                              LTERM=T1033A

                                 1039                  *,    LINEGRP 14  DDNAME=SYSRDR1  UNITYPE=LOCAL-READER
                                 1040                  *,                FEAT=(NONSWITCHED,POLL)

                                 1042                  *,       LINE 16  ADDR=0B1
                                 1043                  *,                TYPLIST=N/A

                                 1046                  *,               TERMINAL 1    ADDR=N/A  EDIT=NO
                                 1047                  *,                              LTERM=T1033A

                                 1051                  *,    LINEGRP 15  DDNAME=SYSPRT1  UNITYPE=LOCAL-PRINTER
                                 1052                  *,                FEAT=(NONSWITCHED,POLL)

                                 1054                  *,       LINE 17  ADDR=0B2
                                 1055                  *,                TYPLIST=N/A

                                 1058                  *,               TERMINAL 1    ADDR=N/A  EDIT=NO
                                 1059                  *,                PRTSYS    EDIT=(NO,ULC)

                                 1063                  *,    LINEGRP 16  DDNAME=SYSTAPE  UNITYPE=LOCAL-TAPE
                                 1064                  *,                FEAT=(NONSWITCHED,POLL)

                                 1066                  *,       LINE 18  ADDR=0B5
                                 1067                  *,                TYPLIST=N/A

                                 1070                  *,               TERMINAL 1    ADDR=N/A  EDIT=NO
                                 1071                  *,                TAPESYS   EDIT=(NO,ULC)

                                 1075                  *,    LINEGRP 17  DDNAME=SYS1   UNITYPE=SPOOL
                                 1076                  *,                DDNAME=SYS2
                                 1077                  *,                DDNAME=SYS3
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                        F01OCT71  10/02/72
                                  1078              *,                DDNAME=SYS4
                                  1079              *,                FEAT=(NONSWITCHED,POLL)

                                  1081              *,        LINE 19  ADDR=C0C
                                  1082              *,                TYPLIST=N/A

                                  1085              *,               TERMINAL 1   ADDR=N/A  EDIT=NO
                                  1086              *,                 SP1        EDIT=(NO,ULC)
                                  1087              *,                 SP2        EDIT=(NO,ULC)

                                  1091              *,     LINEGRP 21  DDNAME=ARUDDGP1  UNITYPE=7770
                                  1092              *,                FEAT=(SWITCHED)

                                  1094              *,        LINE 2C  ADDR=0F0
                                  1095              *,                TYPLIST=N/A

                                  1098              *,               TERMINAL 1   ADDR=C1  EDIT=NO
                                  1099              *,                 INQU0464   EDIT=(NO,ULC)

                                  1102              *,        LINE 21  ADDR=0F3
                                  11C3              *,                TYPLIST=N/A

                                  11C6              *,               TERMINAL 1   ADDR=C4  EDIT=NO
                                  1107              *,                 INQU0469   EDIT=(NO,ULC)

                                  1111              *,     LINEGRP 22  DDNAME=DTWX3335  UNITYPE=TWX
                                  1112              *,                FEAT=(SWITCHED)

                                  1114              *,        LINE 22  ADDR=0D1
                                  1115              *,                TYPLIST=IDLST

                                  1118              *,               TERMINAL 1   ADDR=E2  EDIT=NO
                                  1119              *,                 INQU0477   EDIT=(NO,ULC)

                                  1123              *,     LINEGRP 23  DDNAME=DD2740A  UNITYPE=2740
                                  1124              *,                FEAT=(SWITCHED)

                                  1126              *,        LINE 23  ADDR=061
                                  1127              *,                TYPLIST=DIALST

                                  113C              *,               TERMINAL 1   ADDR=E2  EDIT=NO
                                  1131              *,                 INQU0485   EDIT=(NO,ULC)
```



```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                        F01OCT71  10/02/72
                                  1135              *,     LINEGRP 24  DDNAME=DD1050A  UNITYPE=1050
                                  1136              *,                FEAT=(SWITCHED)

                                  1138              *,        LINE 24  ADDR=060
                                  1139              *,                TYPLIST=DIALST

                                  1142              *,               TERMINAL 1   ADDR=E2  EDIT=NO
                                  1143              *,                 INQU0493   EDIT=(NO,ULC)

                                  1146              *,                 *********************
                                  1147              *,                 **                 **
                                  1148              *,        LINE 25        DIAL POOL
                                  1149              *,                 **                 **
                                  1150              *,                 *********************

                                  1152              *,                TERMINAL 1   ****  SUBPOOL  ****
                                  1153              *,                 ELEANOR    EDIT=(NO,ULC)
                                  1154              *,                 CAROL      EDIT=(NO,ULC)
                                  1155              *,                 SHARRON    EDIT=(NO,ULC)

                                  1157              *,                TERMINAL 2   ****  SUBPOOL  ****
                                  1158              *,                 T2741N1    EDIT=(NO,ULC)
                                  1159              *,                 T2741N2    EDIT=(NO,ULC)

                                  1161              *,                TERMINAL 3   ****  SUBPOOL  ****
                                  1162              *,                 T2741N3    EDIT=(NO,ULC)
                                  1163              *,                 T2741N4    EDIT=(NO,ULC)

                                  1165              *,                TERMINAL 4   ****  SUBPOOL  ****
                                  1166              *,                 SW1050     EDIT=(NO,ULC)
                                  1167              *,                 SWPRNTR2   EDIT=(NO,ULC)  COMPT=PCH1
                                  1168              *,                 SWTPPNCH   EDIT=(NO,ULC)  COMPT=PCH1
                                  1169              *,                 SWCDPNCH   EDIT=(NO,ULC)  COMPT=PCH2
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                               F01OCT71  10/02/72
                                  1173+          PUNCH '//IMSGEN1 JOB (82C,6443),IMS,MSGLEVEL=1,MSGCLASS=A,CLASX
                                     +                S=D,PRTY=8'
                                  1174+          PUNCH '//STEP1 EXEC PGM=IEBCOPY'
                                  1175+          PUNCH '//SYSPRINT DD SYSOUT=A'
                                  1176+          PUNCH '//SYSUT1 DD DSN=IMS2.GENLIB,DISP=SHR'
                                  1177+          PUNCH '//SYSUT2 DD DSN=IMS2.MACLIB,DISP=OLD'
                                  1178+          PUNCH '//SYSUT3 DD UNIT=SYSDA,SPACE=(80,(10,15))  '
                                  1179+          PUNCH '//SYSUT4 DD UNIT=SYSDA,SPACE=(256,(5,1))   '
                                  1180+          PUNCH '//SYSIN DD *'
                                  1181+          PUNCH '  COPY OUTDD=SYSUT2,INDD=SYSUT1  '
                                  1182+          PUNCH '  SELECT MEMBER=ALPHA '
                                  1183+          PUNCH '  SELECT MEMBER=CKODN '
                                  1184+          PUNCH '  SELECT MEMBER=CKOPT '
                                  1185+          PUNCH '  SELECT MEMBER=CONVERT '
                                  1186+          PUNCH '  SELECT MEMBER=DATASET '
                                  1187+          PUNCH '  SELECT MEMBER=DBD '
                                  1188+          PUNCH '  SELECT MEMBER=DBDGEN '
                                  1189+          PUNCH '  SELECT MEMBER=DBDLRECL '
                                  1190+          PUNCH '  SELECT MEMBER=DEVSIZE '
                                  1191+          PUNCH '  SELECT MEMBER=DMAN '
                                  1192+          PUNCH '  SELECT MEMBER=EXPARMS '
                                  1193+          PUNCH '  SELECT MEMBER=EXTDBD '
                                  1194+          PUNCH '  SELECT MEMBER=FIELD '
                                  1195+          PUNCH '  SELECT MEMBER=FINISH '
                                  1196+          PUNCH '  SELECT MEMBER=FLD '
                                  1197+          PUNCH '  SELECT MEMBER=FLDK '
                                  1198+          PUNCH '  SELECT MEMBER=GLOBALS '
                                  1199+          PUNCH '  SELECT MEMBER=HIERSEQ '
                                  1200+          PUNCH '  SELECT MEMBER=LCHILD '
                                  1201+          PUNCH '  SELECT MEMBER=PCB '
                                  1202+          PUNCH '  SELECT MEMBER=PCBPDV '
                                  1203+          PUNCH '  SELECT MEMBER=PSBGEN '
                                  1204+          PUNCH '  SELECT MEMBER=SEGM '
                                  1205+          PUNCH '  SELECT MEMBER=SEGPTRS '
                                  1206+          PUNCH '  SELECT MEMBER=SENSEG '
                                  1207+          PUNCH '  SELECT MEMBER=SETFLGS '
                                  1208+          PUNCH '  SELECT MEMBER=SETFREQ '
                                  1209+          PUNCH '  SELECT MEMBER=SOURSEG '
                                  1210+          PUNCH '  SELECT MEMBER=XDFLD '
                                  1211+          PUNCH '  SELECT MEMBER=CURSOR'
                                  1212+          PUNCH '  SELECT MEMBER=DEV'
                                  1213+          PUNCH '  SELECT MEMBER=DFLD'
                                  1214+          PUNCH '  SELECT MEMBER=DFSERR'
                                  1215+          PUNCH '  SELECT MEMBER=DIV'
                                  1216+          PUNCH '  SELECT MEMBER=DPAGE'
                                  1217+          PUNCH '  SELECT MEMBER=FMT'
                                  1218+          PUNCH '  SELECT MEMBER=FMTEND'
                                  1219+          PUNCH '  SELECT MEMBER=LPAGE'
                                  1220+          PUNCH '  SELECT MEMBER=MFLD'
                                  1221+          PUNCH '  SELECT MEMBER=MSG'
                                  1222+          PUNCH '  SELECT MEMBER=MSGEND'
                                  1223+          PUNCH '  SELECT MEMBER=NUM'
                                  1224+          PUNCH '  SELECT MEMBER=SEG'
                                  1225+          PUNCH '  SELECT MEMBER=SUBLT'
                                  1226+          PUNCH '//STEP2 EXEC PGM=IEBUPDTE,'
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                F01OCT71  10/02/72

                                  1227+         PUNCH '// PARM=NEW'
                                  1228+         PUNCH '//SYSPRINT DD  SYSOUT=A'
                                  1229+         PUNCH '//SYSUT2 DD DSN=IMS2.PROCLIB,DISP=OLD'
                                  1230+         PUNCH '//SYSIN  DD    DATA'
                                  1231+         PUNCH './          ADD   NAME=CBLTDLI'
                                  1232+         PUNCH '  LIBRARY RESLIB(CBLTDLI)     DL/I LANGUAGE INTERFACE'
                                  1233+         PUNCH '  ENTRY DLITCBL'
                                  1234+         PUNCH './          ADD   NAME=PLITDLI'
                                  1235+         PUNCH '  LIBRARY RESLIB(PLITDLI)     DL/I LANGUAGE INTERFACE'
                                  1236+         PUNCH '  ENTRY IHESAPD'
                                  1237+         PUNCH './          ADD   NAME=DLIBATCH'
                                  1238+         PUNCH './          NUMBER NEW1=10,INCR=10'
                                  1239+         PUNCH '//          PROC   MBR=TEMPNAME,SOUT=A,PSB=,BUF=8,SPIE=C,TX
                                      +               EST=0'
                                  1240+         PUNCH '//G         EXEC  PGM=DFSRRC00,REGION=130K,'
                                  1241+         PUNCH '//                PARM=''DLI,&&MBR,&&PSB,&&BUF,&&SPIE&&TESX
                                      +               T'''
                                  1242+         PUNCH '//STEPLIB DD DSN=IMS2.RESLIB,DISP=SHR'
                                  1243+         PUNCH '//           DD DSN=IMS2.PGMLIB,DISP=SHR'
                                  1244+         PUNCH '//IMS        DD DSN=IMS2.PSBLIB,DISP=SHR'
                                  1245+         PUNCH '//           DD DSN=IMS2.DBDLIB,DISP=SHR'
                                  1246+         PUNCH '//DFSLCGTT DD DSN=IMS2.LCGT,DISP=SHR'
                                  1247+         PUNCH '//IEFRDER DD DSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),UNIT=(X
                                      +               2400,,DEFER),'
                                  1248+         PUNCH '// DCB=(RECFM=VBS,BLKSIZE=1408,LRECL=1400,BUFNO=1)'
                                  1249+         PUNCH '//SYSUDUMP DD SYSOUT=&&SOUT,DCB=(RECFM=FBA,LRECL=121,BLX
                                      +               KSIZE=6C5),'
                                  1250+         PUNCH '// SPACE=(605,(500,500),RLSE,,RCUND)'
                                  1251+         PUNCH './          ADD   NAME=DBBBATCH'
                                  1252+         PUNCH './          NUMBER NEW1=10,INCR=10'
                                  1253+         PUNCH '//          PROC  MBR=TEMPNAME,SOUT=A,PSB=,BUF=8,SPIE=0,TEX
                                      +               ST=0'
                                  1254+         PUNCH '//G         EXEC  PGM=DFSRRC00,REGION=130K,'
                                  1255+         PUNCH '//                PARM=''DBB,&&MBR,&&PSB,&&BUF,&&SPIE&&TESX
                                      +               T'''
                                  1256+         PUNCH '//STEPLIB DD DSN=IMS2.RESLIB,DISP=SHR'
                                  1257+         PUNCH '//           DD DSN=IMS2.PGMLIB,DISP=SHR'
                                  1258+         PUNCH '//IMSACB  DD  DSN=IMS2.ACBLIB,DISP=SHR'
                                  1259+         PUNCH '//DFSLOGTT DD DSN=IMS2.LCGT,DISP=SHR'
                                  1260+         PUNCH '//IEFRCER DD CSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),UNIT=(X
                                      +               2400,,DEFER),'
                                  1261+         PUNCH '// DCB=(RECFM=VBS,BLKSIZE=1408,LRECL=1400,BUFNO=1)'
                                  1262+         PUNCH '//SYSUDUMP DD SYSOUT=&&SOUT,DCB=(RECFM=FBA,LRECL=121,BLX
                                      +               KSIZE=605),'
                                  1263+         PUNCH '// SPACE=(605,(500,500),RLSE,,ROUND)'
                                  1264+         PUNCH './          ADD  NAME=ACBGEN'
                                  1265+         PUNCH './          NUMBER NEW1=10,INCR=10'
                                  1266+         PUNCH '//          PROC  SOUT=A,COMP=,RGN=100K'
                                  1267+         PUNCH '//G         EXEC   PGM=DFSRRC00,PARM=''UPB,&&COMP'',REGION=X
                                      +               &&RGN'
                                  1268+         PUNCH '//SYSPRINT DD SYSOUT=&&SOUT'
                                  1269+         PUNCH '//IMS          DD DSN=IMS2.PSBLIB,DISP=SHR'
                                  1270+         PUNCH '//             DD DSN=IMS2.DBDLIB,DISP=SHR'
                                  1271+         PUNCH '//IMSACB       DD DSN=IMS2.ACBLIB,DISP=OLD'
                                  1272+         PUNCH '//SYSUT3       DD UNIT=SYSDA,SPACE=(80,(100,100))'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71  10/02/72

                                 1273+         PUNCH '//SYSUT4   DD UNIT=SYSDA,SPACE=(256,(100,100)),DCB=KEYLX
                                   +                 EN=8'
                                 1274+         PUNCH '//COMPCTL  DD DSN=IMS2.PROCLIB(DFSACBCP),DISP=SHR'
                                 1275+         PUNCH './          ADD  NAME=DFSACBCP'
                                 1276+         PUNCH './          NUMBER NEW1=10,INCR=10'
                                 1277+         PUNCH '            COPY  INDD=IMSACB,OUTDD=IMSACB'
                                 1278+         PUNCH './          ADD    NAME=PSBGEN'
                                 1279+         PUNCH './          NUMBER NEW1=10,INCR=10'
                                 1280+         PUNCH '//          PROC   MBR=TEMPNAME,SOUT=A'
                                 1281+         PUNCH '//C         EXEC   PGM=IEUASM,REGION=120K,PARM=''LOAD,NODECX
                                   +                 K'''
                                 1282+         PUNCH '//SYSLIB DD        DSN=IMS2.MACLIB,DISP=SHR'
                                 1283+         PUNCH '//SYSGO   DD       UNIT=SYSDA,DISP=(,PASS),SPACE=(80,(100,1X
                                   +                 00),RLSE),'
                                 1284+         PUNCH '//                 DCB=(BLKSIZE=400,RECFM=FB,LRECL=80)'
                                 1285+         PUNCH '//SYSPRINT DD      SYSOUT=&&SOUT,DCB=(LRECL=121,RECFM=FBM,BX
                                   +                 LKSIZE=605),'
                                 1286+         PUNCH '//                 SPACE=(121,(500,500),RLSE,,ROUND)'
                                 1287+         PUNCH '//SYSUT1 DD        UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(1X
                                   +                 00,50))'
                                 1288+         PUNCH '//SYSUT2 DD        UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(1X
                                   +                 00,50))'
                                 1289+         PUNCH '//SYSUT3 DD        UNIT=(SYSDA,SEP=(SYSLIB,SYSUT1,SYSUT2)),X
                                   +                 '
                                 1290+         PUNCH '//                 SPACE=(1700,(100,50))'
                                 1291+         PUNCH '//L         EXEC   PGM=DFSILNK0,PARM=''XREF,LIST'',COND=(0,X
                                   +                 LT,C),REGION=120K'
                                 1292+         PUNCH '//STEPLIB DD       DSN=IMS2.RESLIB,DISP=SHR'
                                 1293+         PUNCH '//SYSLIN DD        DSN=*.C.SYSGO,DISP=(OLD,DELETE)'
                                 1294+         PUNCH '//SYSPRINT DD      SYSOUT=&&SOUT,DCB=(LRECL=121,RECFM=FBA,BX
                                   +                 LKSIZE=605),'
                                 1295+         PUNCH '//                 SPACE=(121,(100,100),RLSE)'
                                 1296+         PUNCH '//SYSLMOD DD       DSN=IMS2.PSBLIB(&&MBR),DISP=SHR'
                                 1297+         PUNCH '//SYSUT1 DD        UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),DISP=(X
                                   +                 ,DELETE),'
                                 1298+         PUNCH '//                 SPACE=(1024,(100,10),RLSE)'
                                 1299+         PUNCH './          ADD    NAME=DBDGEN'
                                 1300+         PUNCH './          NUMBER NEW1=10,INCR=10'
                                 1301+         PUNCH '//          PROC   MBR=TEMPNAME,SOUT=A'
                                 1302+         PUNCH '//C         EXEC   PGM=IEUASM,REGION=120K,PARM=''LOAD,NODECX
                                   +                 K'''
                                 1303+         PUNCH '//SYSLIB DD        DSN=IMS2.MACLIB,DISP=SHR'
                                 1304+         PUNCH '//SYSGO  DD        UNIT=SYSDA,DISP=(,PASS),SPACE=(80,(100,1X
                                   +                 00),RLSE),'
                                 1305+         PUNCH '//                 DCB=(BLKSIZE=400,RECFM=FB,LRECL=80)'
                                 1306+         PUNCH '//SYSPRINT DD      SYSOUT=&&SOUT,DCB=(LRECL=121,RECFM=FBM,BX
                                   +                 LKSIZE=605),'
                                 1307+         PUNCH '//                 SPACE=(121,(500,500),RLSE,,ROUND)'
                                 1308+         PUNCH '//SYSUT1 DD        UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(1X
                                   +                 00,50))'
                                 1309+         PUNCH '//SYSUT2 DD        UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(1X
                                   +                 00,50))'
                                 1310+         PUNCH '//SYSUT3 DD        UNIT=(SYSDA,SEP=(SYSLIB,SYSUT1,SYSUT2)),X
                                   +                 '
                                 1311+         PUNCH '//                 SPACE=(1700,(100,50))'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71  10/02/72
                                 1312+          PUNCH '//L         EXEC   PGM=DFSILNKO,PARM=''XREF,LIST'',COND=(0,X
                                    +                 LT,C),REGION=120K'
                                 1313+          PUNCH '//STEPLIB DD     DSN=IMS2.RESLIB,DISP=SHR'
                                 1314+          PUNCH '//SYSLIN DD      DSN=*.C.SYSGO,DISP=(OLD,DELETE)'
                                 1315+          PUNCH '//SYSPRINT DD    SYSOUT=&&SOUT,DCB=(LRECL=121,RECFM=FBA,BX
                                    +                 LKSIZE=605),'
                                 1316+          PUNCH '//                 SPACE=(121,(100,100),RLSE)'
                                 1317+          PUNCH '//SYSLMOD DD     DSN=IMS2.DBDLIB(&&MBR),DISP=SHR'
                                 1318+          PUNCH '//SYSUT1 DD      UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),DISP=(X
                                    +                 ,DELETE),'
                                 1319+          PUNCH '//                 SPACE=(1024,(100,10),RLSE)'
                                 1320+          PUNCH './          ADD    NAME=IMSCOBOL'
                                 1321+          PUNCH './          NUMBER NEW1=10,INCR=10'
                                 1322+          PUNCH '//          PROC   MBR=,PAGES=60,SOUT=A'
                                 1323+          PUNCH '//C         EXEC   PGM=IKFCBL00,REGION=150K,'
                                 1324+          PUNCH '//                 PARM=''SIZE=130K,BUF=10K,LINECNT=50'''
                                 1325+          PUNCH '//SYSLIN DD      DSN=&&&&LIN,DISP=(MOD,PASS),UNIT=SYSDA,'
                                 1326+          PUNCH '//                 DCB=(LRECL=80,RECFM=FB,BLKSIZE=400),'
                                 1327+          PUNCH '//                 SPACE=(3520,(40,10),RLSE,,ROUND)'
                                 1328+          PUNCH '//SYSPRINT DD    SYSOUT=&&SOUT,DCB=(LRECL=121,BLKSIZE=605X
                                    +                 ,RECFM=FBA),'
                                 1329+          PUNCH '//                 SPACE=(605,(&&PAGES.0,&&PAGES),RLSE,,ROUX
                                    +                 ND)'
                                 1330+          PUNCH '//SYSUT1 DD      UNIT=SYSDA,DISP=(,DELETE),'
                                 1331+          PUNCH '//                 SPACE=(3520,(100,10),RLSE,,ROUND)'
                                 1332+          PUNCH '//SYSUT2 DD      UNIT=SYSDA,DISP=(,DELETE),'
                                 1333+          PUNCH '//                 SPACE=(3520,(100,10),RLSE,,ROUND)'
                                 1334+          PUNCH '//SYSUT3 DD      UNIT=SYSDA,DISP=(,DELETE),'
                                 1335+          PUNCH '//                 SPACE=(3520,(100,10),RLSE,,ROUND)'
                                 1336+          PUNCH '//SYSUT4 DD      UNIT=SYSDA,DISP=(,DELETE),'
                                 1337+          PUNCH '//                 SPACE=(3520,(100,10),RLSE,,ROUND)'
                                 1338+          PUNCH '//L         EXEC   PGM=DFSILNKO,REGION=120K,PARM=''XREF,LETX
                                    +                 ,LIST'','
                                 1339+          PUNCH '//                 COND=(4,LT,C)'
                                 1340+          PUNCH '//STEPLIB DD     DSN=IMS2.RESLIB,DISP=SHR'
                                 1341+          PUNCH '//SYSLIB DD      DSN=SYS1.COBLIB,DISP=SHR'
                                 1342+          PUNCH '//          DD     DSN=SYS1.PL1LIB,DISP=SHR'
                                 1343+          PUNCH '//RESLIB DD      DSN=IMS2.RESLIB,DISP=SHR'
                                 1344+          PUNCH '//SYSLIN DD      DSN=&&&&LIN,DISP=(OLD,DELETE),VOL=REF=*.X
                                    +                 C.SYSLIN'
                                 1345+          PUNCH '//          DD     DSN=IMS2.PROCLIB(CBLTDLI),DISP=SHR'
                                 1346+          PUNCH '//          DD     DDNAME=SYSIN'
                                 1347+          PUNCH '//SYSLMOD DD     DSN=IMS2.PGMLIB(&&MBR),DISP=SHR'
                                 1348+          PUNCH '//SYSPRINT DD    SYSOUT=&&SOUT,DCB=(RECFM=FBA,LRECL=121,BX
                                    +                 LKSIZE=605),'
                                 1349+          PUNCH '//                 SPACE=(605,(&&PAGES.0,&&PAGES),RLSE,,ROUX
                                    +                 ND)'
                                 1350+          PUNCH '//SYSUT1 DD      UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),DISP=(X
                                    +                 ,DELETE),'
                                 1351+          PUNCH '//                 SPACE=(3520,(100,10),RLSE,,ROUND)'
                                 1352+          PUNCH './          ADD    NAME=IMSCOBGO'
                                 1353+          PUNCH './          NUMBER NEW1=10,INCR=10'
                                 1354+          PUNCH '//          PROC   MBR=,PAGES=60,SOUT=A,PSB=,SPIE=0,TEST=0,X
                                    +                 BUF=8'
                                 1355+          PUNCH '//C         EXEC   PGM=IKFCBL00,REGION=150K,'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT    SOURCE STATEMENT                                                  F01OCT71  10/02/72

                                1356+          PUNCH '//           PARM=''SIZE=130K,BUF=10K,LINECNT=50'''
                                1357+          PUNCH '//SYSLIN DD  DSN=&&&&LIN,DISP=(MOD,PASS),UNIT=SYSDA,'
                                1358+          PUNCH '//           DCB=(LRECL=80,RECFM=FB,BLKSIZE=400),'
                                1359+          PUNCH '//           SPACE=(3520,(40,10),RLSE,,ROUND)'
                                1360+          PUNCH '//SYSPRINT DD  SYSOUT=&&SOUT,DCB=(LRECL=121,BLKSIZE=605X
                                   +                ,RECFM=FBA),'
                                1361+          PUNCH '//           SPACE=(605,(&&PAGES.0,&&PAGES),RLSE,,ROUX
                                   +                ND)'
                                1362+          PUNCH '//SYSUT1 DD  UNIT=SYSDA,DISP=(,DELETE),'
                                1363+          PUNCH '//           SPACE=(3520,(100,10),RLSE,,ROUND)'
                                1364+          PUNCH '//SYSUT2 DD  UNIT=SYSDA,DISP=(,DELETE),'
                                1365+          PUNCH '//           SPACE=(3520,(100,10),RLSE,,ROUND)'
                                1366+          PUNCH '//SYSUT3 DD  UNIT=SYSDA,DISP=(,DELETE),'
                                1367+          PUNCH '//           SPACE=(3520,(100,10),RLSE,,ROUND)'
                                1368+          PUNCH '//SYSUT4 DD  UNIT=SYSDA,DISP=(,DELETE),'
                                1369+          PUNCH '//           SPACE=(3520,(100,10),RLSE,,ROUND)'
                                1370+          PUNCH '//L       EXEC  PGM=DFSILNK0,REGION=120K,PARM=''XREF,LETX
                                   +                ,LIST'','
                                1371+          PUNCH '//           COND=(4,LT,C)'
                                1372+          PUNCH '//STEPLIB DD  DSN=IMS2.RESLIB,DISP=SHR'
                                1373+          PUNCH '//SYSLIB DD  DSN=SYS1.COBLIB,DISP=SHR'
                                1374+          PUNCH '//        DD  DSN=SYS1.PLILIB,DISP=SHR'
                                1375+          PUNCH '//RESLIB DD  DSN=IMS2.RESLIB,DISP=SHR'
                                1376+          PUNCH '//SYSLIN DD  DSN=&&&&LIN,DISP=(OLD,DELETE),VOL=REF=*.X
                                   +                C.SYSLIN'
                                1377+          PUNCH '//        DD  DSN=IMS2.PROCLIB(CBLTDLI),DISP=SHR'
                                1378+          PUNCH '//        DD  DDNAME=SYSIN'
                                1379+          PUNCH '//SYSLMOD DD  DSN=IMS2.PGMLIB(&&MBR),DISP=SHR'
                                1380+          PUNCH '//SYSPRINT DD  SYSOUT=&&SOUT,DCB=(RECFM=FBA,LRECL=121,BX
                                   +                LKSIZE=605),'
                                1381+          PUNCH '//           SPACE=(605,(&&PAGES.0,&&PAGES),RLSE,,ROUX
                                   +                ND)'
                                1382+          PUNCH '//SYSUT1 DD  UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),DISP=(X
                                   +                ,DELETE),'
                                1383+          PUNCH '//           SPACE=(3520,(100,10),RLSE,,ROUND)'
                                1384+          PUNCH '//G       EXEC  PGM=DFSRRC00,REGION=150K,TIME=2,COND=(4,X
                                   +                LT),'
                                1385+          PUNCH '//           PARM=''DLI,&&MBR,&&PSB,&&BUF,&&SPIE&&TESX
                                   +                T'''
                                1386+          PUNCH '//STEPLIB DD  DSN=IMS2.RESLIB,DISP=SHR'
                                1387+          PUNCH '//        DD  DSN=IMS2.PGMLIB,DISP=SHR'
                                1388+          PUNCH '//IMS     DD  DSN=IMS2.PSBLIB,DISP=SHR'
                                1389+          PUNCH '//        DD  DSN=IMS2.DBDLIB,DISP=SHR'
                                1390+          PUNCH '//IEFRDER DD DSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),UNIT=(X
                                   +                2400,,DEFER),'
                                1391+          PUNCH '//          DCB=(RECFM=VBS,BLKSIZE=1408,LRECL=1400,BUX
                                   +                FNO=1)'
                                1392+          PUNCH '//SYSOUT DD   SYSOUT=&&SOUT,SPACE=(CYL,(1,1)),DCB=(LREX
                                   +                CL=133,RECFM=FA)'
                                1393+          PUNCH '//SYSUDUMP DD  SYSOUT=&&SOUT,DCB=(LRECL=121,RECFM=FBA,BX
                                   +                LKSIZE=3025),'
                                1394+          PUNCH '//           SPACE=(3025,(200,100),RLSE,,ROUND)'
                                1395+          PUNCH './       ADD   NAME=IMSPLI'
                                1396+          PUNCH './       NUMBER NEW1=10,INCR=10'
                                1397+          PUNCH '//       PROC  MBR=,PAGES=50,SOUT=A'
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT                                              F01OCT71  10/02/72
                                  1398+         PUNCH '//C          EXEC  PGM=IEMAA,REGION=114K,'
                                  1399+         PUNCH '//                 PARM=''XREF,ATR,LOAD,NODECK,NOMACRO,,OPTX
                                     +                =1'''
                                  1400+         PUNCH '//SYSUT1 DD        UNIT=SYSDA,SPACE=(1024,(60,60),RLSE,,ROUX
                                     +                ND),'
                                  1401+         PUNCH '//                 DCB=BLKSIZE=1024,DISP=(,DELETE)'
                                  1402+         PUNCH '//SYSUT3 DD        UNIT=SYSDA,SPACE=(1024,(60,60),RLSE,,ROUX
                                     +                ND),'
                                  1403+         PUNCH '//                 DCB=BLKSIZE=1024,DISP=(,DELETE)'
                                  1404+         PUNCH '//SYSPRINT DD      SYSOUT=&&SOUT,DCB=(LRECL=125,BLKSIZE=629X
                                     +                ,RECFM=VBA),'
                                  1405+         PUNCH '//                 SPACE=(605,(&&PAGES.0,&&PAGES),RLSE)'
                                  1406+         PUNCH '//SYSLIN DD        UNIT=SYSDA,SPACE=(80,(250,80),RLSE),DCB=X
                                     +                BLKSIZE=80,'
                                  1407+         PUNCH '//                 DISP=(,PASS)'
                                  1408+         PUNCH '//L          EXEC  PGM=DFSILNK0,PARM=''XREF,LIST,LET'',CONDX
                                     +                =(4,LT,C),'
                                  1409+         PUNCH '//                 REGION=120K'
                                  1410+         PUNCH '//STEPLIB DD       DSN=IMS2.RESLIB,DISP=SHR'
                                  1411+         PUNCH '//SYSLIB DD        DSN=SYS1.PL1LIB,DISP=SHR'
                                  1412+         PUNCH '//          DD     DSN=SYS1.COBLIB,DISP=SHR'
                                  1413+         PUNCH '//RESLIB DD        DSN=IMS2.RESLIB,DISP=SHR'
                                  1414+         PUNCH '//SYSLIN DD        DSN=*.C.SYSLIN,DISP=(OLD,DELETE)'
                                  1415+         PUNCH '//          DD     DSN=IMS2.PROCLIB(PLITDLI),DISP=SHR'
                                  1416+         PUNCH '//          DD     DDNAME=SYSIN'
                                  1417+         PUNCH '//SYSLMOD DD       DSN=IMS2.PGMLIB(&&MBR),DISP=SHR'
                                  1418+         PUNCH '//SYSPRINT DD      SYSOUT=&&SOUT,DCB=(LRECL=121,RECFM=FBA,BX
                                     +                LKSIZE=605),'
                                  1419+         PUNCH '//                 SPACE=(605,(&&PAGES.0,&&PAGES),RLSE)'
                                  1420+         PUNCH '//SYSUT1 DD        UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(5,X
                                     +                1),RLSE)'
                                  1421+         PUNCH './           ADD   NAME=IMSPLIGO'
                                  1422+         PUNCH './           NUMBER NEW1=10,INCR=10'
                                  1423+         PUNCH '//           PROC  MBR=,PAGES=50,SOUT=A,PSB=,SPIE=0,TEST=0,X
                                     +                BUF=8'
                                  1424+         PUNCH '//C          EXEC  PGM=IEMAA,REGION=114K,'
                                  1425+         PUNCH '//                 PARM=''XREF,ATR,LOAD,NODECK,NOMACRO,,OPTX
                                     +                =1'''
                                  1426+         PUNCH '//SYSUT1 DD        UNIT=SYSDA,SPACE=(1024,(60,60),RLSE,,ROUX
                                     +                ND),'
                                  1427+         PUNCH '//                 DCB=BLKSIZE=1024,DISP=(,DELETE)'
                                  1428+         PUNCH '//SYSUT3 DD        UNIT=SYSDA,SPACE=(1024,(60,60),RLSE,,ROUX
                                     +                ND),'
                                  1429+         PUNCH '//                 DCB=BLKSIZE=1024,DISP=(,DELETE)'
                                  1430+         PUNCH '//SYSPRINT DD      SYSOUT=&&SOUT,DCB=(LRECL=125,BLKSIZE=629X
                                     +                ,RECFM=VBA),'
                                  1431+         PUNCH '//                 SPACE=(605,(&&PAGES.0,&&PAGES),RLSE)'
                                  1432+         PUNCH '//SYSLIN DD        UNIT=SYSDA,SPACE=(80,(250,80),RLSE),DCB=X
                                     +                BLKSIZE=80,'
                                  1433+         PUNCH '//                 DISP=(,PASS)'
                                  1434+         PUNCH '//L          EXEC  PGM=DFSILNK0,PARM=''XREF,LIST,LET'',CONDX
                                     +                =(4,LT,C),'
                                  1435+         PUNCH '//                 REGION=120K'
                                  1436+         PUNCH '//STEPLIB DD       DSN=IMS2.RESLIB,DISP=SHR'
                                  1437+         PUNCH '//SYSLIB DD        DSN=SYS1.PL1LIB,DISP=SHR'
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT                                        F01OCT71  10/02/72

                                  1438+          PUNCH '//          DD     DSN=SYS1.COBLIB,DISP=SHR'
                                  1439+          PUNCH '//RESLIB DD        DSN=IMS2.RESLIB,DISP=SHR'
                                  144C+          PUNCH '//SYSLIN DD        DSN=*.C.SYSLIN,DISP=(OLD,DELETE)'
                                  1441+          PUNCH '//          DD     DSN=IMS2.PROCLIB(PLITOLI),DISP=SHR'
                                  1442+          PUNCH '//          DD     DDNAME=SYSIN'
                                  1443+          PUNCH '//SYSLMOD DD       DSN=IMS2.PGMLIB(&&MBR),DISP=SHR'
                                  1444+          PUNCH '//SYSPRINT DD      SYSOUT=&&SOUT,DCB=(LRECL=121,RECFM=FBA,BX
                                     +                LKSIZE=605),'
                                  1445+          PUNCH '//                 SPACE=(605,(&&PAGES,0,&&PAGES),RLSE)'
                                  1446+          PUNCH '//SYSUT1 DD        UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(5,X
                                     +                1),RLSE)'
                                  1447+          PUNCH '//G          EXEC  PGM=DFSRRC00,REGION=150K,TIME=5,COND=(4,X
                                     +                LT),'
                                  1448+          PUNCH '//                 PARM=''DLI,&&MBR,&&PSB,&&BUF,&&SPIE&&TESX
                                     +                T'''
                                  1449+          PUNCH '//STEPLIB DD       DSN=IMS2.RESLIB,DISP=SHR'
                                  1450+          PUNCH '//           DD    DSN=IMS2.PGMLIB,DISP=SHR'
                                  1451+          PUNCH '//IMS       DD     DSN=IMS2.PSBLIB,DISP=SHR'
                                  1452+          PUNCH '//          DD     DSN=IMS2.DBDLIB,DISP=SHR'
                                  1453+          PUNCH '//IEFRDER DD DSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),UNIT=(X
                                     +                2400,,DEFER),'
                                  1454+          PUNCH '//                DCB=(RECFM=VBS,BLKSIZE=1408,LRECL=1400,BUX
                                     +                FNO=1)'
                                  1455+          PUNCH '//SYSPRINT DD      SYSOUT=&&SOUT,DCB=(LRECL=121,BLKSIZE=605X
                                     +                ,RECFM=FBA),'
                                  1456+          PUNCH '//                 SPACE=(605,(50C,5C0),RLSE,,ROUND)'
                                  1457+          PUNCH '//SYSUCUMP DD      SYSOUT=&&SOUT,DCB=(LRECL=121,BLKSIZE=605X
                                     +                ,RECFM=FBA),'
                                  1458+          PUNCH '//                 SPACE=(605,(500,50C),RLSE,,RCUND)'
                                  1459+          PUNCH './          ADD   NAME=MFDBDUMP'
                                  1460+          PUNCH './          NUMBER NEW1=10,INCR=10 '
                                  1461+          PUNCH '//          PROC SOUT=A'
                                  1462+          PUNCH '//DUMP      EXEC  PGM=DFSRRC00,PARM=''DLI,DFSSAM08'',REGIOX
                                     +                N=130K'
                                  1463+          PUNCH '//STEPLIB DD DSN=IMS2.RESLIB,DISP=SHR'
                                  1464+          PUNCH '//            DD DSN=IMS2.PGMLIB,DISP=SHR'
                                  1465+          PUNCH '//IMS         DD DSN=IMS2.PSBLIB,DISP=SHR'
                                  1466+          PUNCH '//            DD DSN=IMS2.DBDLIB,DISP=SHR'
                                  1467+          PUNCH '//SYSUCUMP DD SYSOUT=&&SOUT'
                                  1468+          PUNCH '//DI21PART DD DSN=IMS2.DI21PART,DISP=SHR'
                                  1469+          PUNCH '//DI21PARO DD DSN=IMS2.DI21PARO,DISP=SHR'
                                  147C+          PUNCH '//OUTPUT DD     SYSOUT=&&SOUT'
                                  1471+          PUNCH './          ADD  NAME=MFDBLOAD'
                                  1472+          PUNCH './          NUMBER NEW1=10,INCR=10'
                                  1473+          PUNCH '//          PROC SOUT=A'
                                  1474+          PUNCH '//LOAD      EXEC  PGM=DFSRRC00,PARM=''DLI,DFSSAM01'',REGIOX
                                     +                N=130K'
                                  1475+          PUNCH '//STEPLIB DD DSN=IMS2.RESLIB,DISP=SHR'
                                  1476+          PUNCH '//            DD DSN=IMS2.PGMLIB,DISP=SHR'
                                  1477+          PUNCH '//IMS         DD DSN=IMS2.PSBLIB,DISP=SHR'
                                  1478+          PUNCH '//            DD DSN=IMS2.DBDLIB,DISP=SHR'
                                  1479+          PUNCH '//SYSUDUMP DD SYSOUT=&&SCUT'
                                  1480+          PUNCH '//DI21PART DD DSN=IMS2.DI21PART(PRIME),DISP=(,KEEP),DCBX
                                     +                =DSORG=IS,       X'
                                  1481+           UNCH '//                 SPACE=(CYL,3,,CONTIG),VOL=SER=&&PSER,UNIX
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71  10/02/72

                                    +          T=&&PUNIT'
                                 1482+        PUNCH '//DI21PARO DD DSN=IMS2.DI21PARO,DISP=(,KEEP),SPACE=(CYLX
                                    +          ,3,,CONTIG),     X'
                                 1483+        PUNCH '//                   VOL=SER=&&OSER,UNIT=&&OUNIT'
                                 1484+        PUNCH '//SYSOUT DD   SYSOUT=&&SOUT'
                                 1485+        PUNCH '//INPUT DD DSN=IMS2.MACLIB(MFCFSYSN),DISP=SHR'
                                 1486+        PUNCH './         ADD  NAME=IMSMSG'
                                 1487+        PUNCH './         NUMBER NEW1=10,INCR=10'
                                 1488+        PUNCH '//MESSAGE JOB 1,IMS,MSGLEVEL=1,PRTY=11,CLASS=A,MSGCLASSX
                                    +          =3,REGION=30K'
                                 1489+        PUNCH '//REGION EXEC  PGM=DFSRRC00,REGION=30K,TIME=1440,'
                                 1490+        PUNCH '//                PARM=''MSG,00100000000'''
                                 1491+        PUNCH '//STEPLIB DD   DSN=IMS2.RESLIB,DISP=SHR'
                                 1492+        PUNCH '//         DD   DSN=IMS2.PGMLIB,DISP=SHR'
                                 1493+        PUNCH '//SYSUDUMP DD  SYSOUT=3,DCB=(LRECL=125,BLKSIZE=3129,RECX
                                    +          FM=VBA),'
                                 1494+        PUNCH '//                SPACE=(125,(2500,100),RLSE,,ROUND)'
                                 1495+        PUNCH './         ADD  NAME=IMSBATCH'
                                 1496+        PUNCH './         NUMBER NEW1=10,INCR=10'
                                 1497+        PUNCH '//         PROC  MBR=TEMPNAME,SOUT=3,OPT=N,SPIE=0,TEST=0,X
                                    +          '
                                 1498+        PUNCH '//                PSB=,IN=,OUT=,CIRCA=000'
                                 1499+        PUNCH '//G        EXEC  PGM=DFSRRC00,REGION=30K,'
                                 1500+        PUNCH '//  PARM=''BMP,&&MBR,&&PSB,&&IN,&&OUT,&&OPT&&SPIE&&TESTX
                                    +          &&CIRCA'''
                                 1501+        PUNCH '//STEPLIB DD   DSN=IMS2.RESLIB,DISP=SHR'
                                 1502+        PUNCH '//         DD   DSN=IMS2.PGMLIB,DISP=SHR'
                                 1503+        PUNCH '//SYSUDUMP DD  SYSOUT=&&SOUT,DCB=(LRECL=121,RECFM=VBA,BX
                                    +          LKSIZE=3129),'
                                 1504+        PUNCH '//                SPACE=(125,(2500,100),RLSE,,ROUND)'
                                 1505+        PUNCH './         ADD  NAME=IMS'
                                 1506+        PUNCH './         NUMBER NEW1=10,INCR=10'
                                 1507+        PUNCH '//         PROC  RGN=300K,DMBP=000,PSBP=000,DBBP=000,TPDPX
                                    +          =000,WKAP=000,'
                                 1508+        PUNCH '//                S=0,PKEY=1,PTY=254,TEST=1,QCR=000,FBP=00X
                                    +          0,SOUT=A'
                                 1509+        PUNCH '//IEFPROC EXEC PGM=DFSRRC00,REGION=&&RGN,'
                                 1510+        PUNCH '// PARM=(CTL, '
                                 1511+        PUNCH '// &&PTY&&PKEY.0&&TEST&&S&&QCR&&FBP&&PSBP&&DMBP&&DBBP&&X
                                    +          TPDP.000000&&WKAP)'
                                 1512+        PUNCH '//*'
                                 1513+        PUNCH '//*'
                                 1514+        PUNCH '//*               PARM=(AAA,BBBCDEFGGGHHHIIIJJJKKKLLL00000X
                                    +          0000)' 4107
                                 1515+        PUNCH '//*'
                                 1516+        PUNCH '//*               AAA             REGION TYPE'
                                 1517+        PUNCH '//*               BBB             DISPATCHING PRIORITY'
                                 1518+        PUNCH '//*               C               REGION PROTECT KEY(0=0; 1X
                                    +          =NON ZERO)'
                                 1519+        PUNCH '//*               E               TEST OPTION'
                                 1520+        PUNCH '//*               F               CONTROL PROGRAM SUFFIX'
                                 1521+        PUNCH '//*               GGG             NUMBER OF QCR BUFFERS'
                                 1522+        PUNCH '//*               HHH             FORMAT BUFFER POOL SIZE(IX
                                    +          N 1K BLOCKS)'
                                 1523+        PUNCH '//*               III             PSB POOL SIZE(IN 1K BLOCKX
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                       F01OCT71  10/02/72
                                     +            S)'
                                  1524+         PUNCH '//*             JJJ             DMB POOL SIZE(IN 1K BLOCKX
                                     +            S)'
                                  1525+         PUNCH '//*             KKK             DATABASE BUFFER POOL SIZEX
                                     +            (IN 1K BLOCKS)'
                                  1526+         PUNCH '//*             LLL             LINE BUFFER POOL(IN 1K BLX
                                     +            OCKS)'
                                  1527+         PUNCH '//*             000000          RESERVED PARM AREA'
                                  1528+         PUNCH '//*             000             WORKING STORAGE(IN 1K BLOX
                                     +            CKS)'
                                  1529+         PUNCH '//*'
                                  1530+         PUNCH '//*             IF SYMBOLIC PARMS ARE NOT SPECIFIED;  THX
                                     +            E VALUES'
                                  1531+         PUNCH '//*             PROVIDED BY SYSTEM DEFINITION ARE USED.'
                                  1532+         PUNCH '//*'
                                  1533+         PUNCH '//IEFRDER DD DSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),UNIT=(X
                                     +            2400,,DEFER),'
                                  1534+         PUNCH '//             DCB=(RECFM=VBS,BLKSIZE=1408,LRECL=1400,BUX
                                     +            FNO=1)'
                                  1535+         PUNCH '//IMSLOGR DD    DSN=IMSLOG,DISP=(OLD,KEEP),VOL=SER=C0000X
                                     +            0,'
                                  1536+         PUNCH '//              UNIT=AFF=IEFRDER'
                                  1537+         PUNCH '//QBLKS DD DSN=IMS2.QBLKS,DISP=OLD'
                                  1538+         PUNCH '//SHMSG DD DSN=IMS2.SHMSG,DISP=OLD'
                                  1539+         PUNCH '//LGMSG DD DSN=IMS2.LGMSG,DISP=OLD'
                                  1540+         PUNCH '//IMSACB DD DSN=IMS2.ACBLIB,DISP=SHR'
                                  1541+         PUNCH '//DFSLOGTT DD DSN=IMS2.LOGT,DISP=SHR'
                                  1542+         PUNCH '//IMSDILIB DD DSN=IMS2.FORMAT,DISP=SHR'
                                  1543+         PUNCH '//IMSSPA DD     DSN=IMS2.SPA,DISP=OLD'
                                  1544+         PUNCH '//SYSUDUMP DD   SYSOUT=&&SOUT,DCB=(LRECL=125,RECFM=FBA,BX
                                     +            LKSIZE=3129),'
                                  1545+         PUNCH '//              SPACE=(6050,300,,,ROUND)'
                                  1546+         PUNCH '//IMSDBL DD     DSN=IMS2.DBLLOG,DISP=SHR'
                                  1547+         PUNCH '//DD2740S  DD     UNIT=0C2            ***  IMS LINE 2'
                                  1548+         PUNCH '//         DD     UNIT=0C3            ***  IMS LINE 3'
                                  1549+         PUNCH '//D2740NSC DD     UNIT=0C1            ***  IMS LINE 4'
                                  1550+         PUNCH '//DD3270R  DD     UNIT=0C9            ***  IMS LINE 5'
                                  1551+         PUNCH '//DD3270L  DD     UNIT=310            ***  IMS LINE 6   TEX
                                     +            RMINAL 1'
                                  1552+         PUNCH '//         DD     UNIT=312            ***  IMS LINE 6   TEX
                                     +            RMINAL 2'
                                  1553+         PUNCH '//DC327CL2 DD     UNIT=311            ***  IMS LINE 7   TEX
                                     +            RMINAL 1'
                                  1554+         PUNCH '//DD1050   DD     UNIT=0CC            ***  IMS LINE 8'
                                  1555+         PUNCH '//DD2260R  DD     UNIT=0E0            ***  IMS LINE 9'
                                  1556+         PUNCH '//DD277C   DD     UNIT=0C8            ***  IMS LINE 10'
                                  1557+         PUNCH '//DC2980   DD     UNIT=0CA            ***  IMS LINE 11'
                                  1558+         PUNCH '//DC2741   DD     UNIT=097            ***  IMS LINE 12'
                                  1559+         PUNCH '//DD2741S  DD     UNIT=05C            ***  IMS LINE 13'
                                  1560+         PUNCH '//DC2741S2 DD     UNIT=053            ***  IMS LINE 14'
                                  1561+         PUNCH '//DD1030   DD     UNIT=0E1            ***  IMS LINE 15'
                                  1562+         PUNCH '//SYSRDR1  DD     UNIT=081            ***  IMS LINE 16'
                                  1563+         PUNCH '//SYSPRT1  DD     UNIT=082            ***  IMS LINE 17'
                                  1564+         PUNCH '//SYSTAPE  DD     VOL=(PRIVATE,,,99),UNIT=0B5 IMS LINE 1X
                                     +            8'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT    SOURCE STATEMENT                                                   F01OCT71  10/02/72
                                 1565+           PUNCH '//SYS1       DD DISP=SHR,DSN=IMS2.SYSO1'
                                 1566+           PUNCH '//SYS2       DD DISP=SHR,DSN=IMS2.SYSO2'
                                 1567+           PUNCH '//SYS3       DD DISP=SHR,DSN=IMS2.SYSO3'
                                 1568+           PUNCH '//SYS4       DD DISP=SHR,DSN=IMS2.SYSO4'
                                 1569+           PUNCH '//ARUDDGP1 DD     UNIT=0F0              ***  IMS LINE 20'
                                 1570+           PUNCH '//         DD     UNIT=0F3              ***  IMS LINE 21'
                                 1571+           PUNCH '//DTWX3335 DD     UNIT=0D1              ***  IMS LINE 22'
                                 1572+           PUNCH '//DD2740A  DD     UNIT=061              ***  IMS LINE 23'
                                 1573+           PUNCH '//DC1050A  DD     UNIT=060              ***  IMS LINE 24'
                                 1574+           PUNCH '//*'
                                 1575+           PUNCH '//*                    USER MUST SUPPLY THE DD STATEMENTS'
                                 1576+           PUNCH '//*                    FOR THE ON-LINE DATABASES TO BE'
                                 1577+           PUNCH '//*                    INSERTED HERE PRIOR TO ATTEMPTING'
                                 1578+           PUNCH '//*                    AN ON-LINE SYSTEM EXECUTION USING'
                                 1579+           PUNCH '//*                    THIS PROCEDURE.'
                                 1580+           PUNCH './          ADD    NAME=SECURITY'
                                 1581+           PUNCH './          NUMBER NEW1=10,INCR=10'
                                 1582+           PUNCH '//          PROC  OPTN=UPDATE,IMS='',C'',SOUT=A'
                                 1583+           PUNCH '//S         EXEC  PGM=DFSISMP0,PARM=''&&OPTN.&&IMS.'''
                                 1584+           PUNCH '//STEPLIB DD DSN=IMS2.RESLIB,DISP=SHR'
                                 1585+           PUNCH '//SYSPRINT DD SYSOUT=&&SOUT,DCB=(RECFM=VBA,BLKSIZE=400,X
                                     +                 BUFL=404)'
                                 1586+           PUNCH '//SYSPUNCH DD  UNIT=SYSDA,SPACE=(80,(800,400),,,ROUND),X
                                     +                                 X'
                                 1587+           PUNCH '//                     DCB=(RECFM=FB,LRECL=80,BLKSIZE=400),DISPX
                                     +                 =(,PASS)'
                                 1588+           PUNCH '//SYSLIN    DD  UNIT=SYSDA,SPACE=(TRK,(1,1)),DCB=(RECFM=X
                                     +                 F,BLKSIZE=80),'
                                 1589+           PUNCH '//                     DISP=(,PASS)'
                                 1590+           PUNCH '//SYSUT1    DD  UNIT=SYSDA,SPACE=(100,(400,400),,,ROUND)X
                                     +                 ,'
                                 1591+           PUNCH '//                     DCB=(BLKSIZE=500,RECFM=FB)'
                                 1592+           PUNCH '//SYSUT2    DD  UNIT=(SYSDA,SEP=SYSUT1),SPACE=(100,(400,X
                                     +                 400),,,ROUND),'
                                 1593+           PUNCH '//                     DCB=*.S.SYSUT1'
                                 1594+           PUNCH '//SYSIN     DD  DSN=NO.SYSIN.DD.ASTERISK'
                                 1595+           PUNCH '//C         EXEC  PGM=IEUASM,PARM=''LOAD,NODECK'',COND=(12X
                                     +                 ,LT,S),REGION=96K'
                                 1596+           PUNCH '//SYSPRINT DD  SYSOUT=&&SOUT,DCB=(RECFM=FBM,LRECL=121,BX
                                     +                 LKSIZE=605)'
                                 1597+           PUNCH '//SYSGO     DD  UNIT=(SYSDA,SEP=SYSPRINT),DISP=(,PASS),'
                                 1598+           PUNCH '//                     DCB=*.S.SYSPUNCH,SPACE=(80,(400,400),,,RX
                                     +                 OUND)'
                                 1599+           PUNCH '//SYSUT1    DD  UNIT=SYSDA,SPACE=(CYL,(5,1))'
                                 1600+           PUNCH '//SYSUT2    DD  UNIT=SYSDA,SPACE=(CYL,(5,1))'
                                 1601+           PUNCH '//SYSUT3    DD  UNIT=(SYSDA,SEP=(SYSUT1,SYSUT2)),SPACE=(X
                                     +                 CYL,(5,1))'
                                 1602+           PUNCH '//SYSIN     DD  DSN=*.S.SYSPUNCH,DISP=(OLD,DELETE)'
                                 1603+           PUNCH '//L         EXEC  PGM=DFSILNK0,PARM=''LIST,NE,OL'',REGION=X
                                     +                 110K,COND=(4,LT,S)'
                                 1604+           PUNCH '//STEPLIB DD DSN=IMS2.RESLIB,DISP=SHR'
                                 1605+           PUNCH '//SYSPRINT DD  SYSOUT=&&SOUT,DCB=(RECFM=FBA,LRECL=121,BX
                                     +                 LKSIZE=605)'
                                 1606+           PUNCH '//SYSLMOD DD DSN=IMS2.RESLIB,DISP=SHR'
                                 1607+           PUNCH '//INPUT     DD  DSN=*.C.SYSGO,DISP=(OLD,DELETE)'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT    SOURCE STATEMENT                                         F01OCT71  10/02/72

                                 1608+            PUNCH '//SYSUT1     DD  UNIT=(SYSDA,SEP=INPUT),SPACE=(CYL,(5,1))X
                                    +                   '
                                 1609+            PUNCH '//SYSLIN     DD  DSN=*.S.SYSLIN,DISP=(OLD,DELETE)'
                                 1610+            PUNCH './         ADD   NAME=FMTUTL'
                                 1611+            PUNCH './         NUMBER NEW1=10,INCR=10'
                                 1612+            PUNCH '//         PROC  SNODE=IMS2,SOR=NOLIB,MBR=NOMBR,MNODE=IMSX
                                    +                   2,MAC=GENLIB,'
                                 1613+            PUNCH '//               RGN=250K,BUF=4,SPIE=0,TEST=0,SOUT=A'
                                 1614+            PUNCH '//PREPROC EXEC PGM=IEUASM,PARM=''NOLOAD,DECK'',REGION=1X
                                    +                   00K'
                                 1615+            PUNCH '//SYSPRINT DD  SYSOUT=&&SOUT'
                                 1616+            PUNCH '//SYSPUNCH DD  DSN=&&&&ITBPASS,DISP=(NEW,PASS),SPACE=(CX
                                    +                   YL,(1,1)),'
                                 1617+            PUNCH '//               UNIT=SYSDA'
                                 1618+            PUNCH '//SYSLIB     DD   DSN=&&MNODE..&&MAC,DISP=SHR'
                                 1619+            PUNCH '//           DD  DSN=SYS1.MACLIB,DISP=SHR'
                                 1620+            PUNCH '//SYSUT1     DD  UNIT=SYSDA,SPACE=(CYL,(1,1))'
                                 1621+            PUNCH '//SYSUT2     DD  UNIT=SYSDA,SPACE=(CYL,(1,1))'
                                 1622+            PUNCH '//SYSUT3     DD  UNIT=SYSDA,SPACE=(CYL,(1,1))'
                                 1623+            PUNCH '//SYSIN      DD  DSN=&&SNODE..&&SOR(&&MBR),DISP=SHR'
                                 1624+            PUNCH '//PHASE1 EXEC  PGM=DFSUNU10,COND=(8,LE,PREPROC),'
                                 1625+            PUNCH '//               PARM=''COMPRESS'',REGION=&&RGN'
                                 1626+            PUNCH '//SYSPRINT DD  SYSOUT=&&SOUT'
                                 1627+            PUNCH '//SYSUDUMP DD  SYSOUT=&&SOUT'
                                 1628+            PUNCH '//UTPRINT    DD  SYSOUT=&&SOUT'
                                 1629+            PUNCH '//SYSLIN     DD  UNIT=SYSDA,SPACE=(CYL,(1,1)),DCB=BLKSIZEX
                                    +                   =800'
                                 1630+            PUNCH '//SYSLMOD    DD  DSN=IMS2.REFERAL,DISP=OLD'
                                 1631+            PUNCH '//DUMMY      DD  DUMMY,DCB=BLKSIZE=80'
                                 1632+            PUNCH '//SYSUT1     DD  UNIT=SYSDA,SPACE=(CYL,(1,1))'
                                 1633+            PUNCH '//SYSTEXT    DD  DSN=&&&&TXTPASS,DISP=(NEW,PASS),UNIT=SYSX
                                    +                   DA,'
                                 1634+            PUNCH '//               SPACE=(CYL,(1,1)),DCB=BLKSIZE=800'
                                 1635+            PUNCH '//SYSIN      DD  DSN=&&&&ITBPASS,DISP=(OLD,DELETE)'
                                 1636+            PUNCH '//PHASE3 EXEC  PGM=DFSRRC00,COND=(4,LE,PHASE1),REGION=&X
                                    +                   &RGN,'
                                 1637+            PUNCH '//               PARM=''DLI,DFSUNU20,DFSUTS20,&&BUF,&&SPIX
                                    +                   E&&TEST'''
                                 1638+            PUNCH '//IMS        DD  DSN=IMS2.PSBLIB,DISP=SHR'
                                 1639+            PUNCH '//           DD  DSN=IMS2.DBDLIB,DISP=SHR'
                                 1640+            PUNCH '//IEFRDER    DD  DSN=&&&&IMSLOG,DISP=(,PASS),UNIT=SYSDA,'
                                 1641+            PUNCH '//               SPACE=(CYL,(1,1)),DCB=DSORG=PS'
                                 1642+            PUNCH '//DFSLOGTT DD  DSN=IMS2.LOGT,DISP=SHR'
                                 1643+            PUNCH '//SYSUDUMP DD  SYSOUT=&&SOUT'
                                 1644+            PUNCH '//SYSPRINT DD  SYSOUT=&&SOUT,DCB=(RECFM=FA,LRECL=133,BLX
                                    +                   KSIZE=133)'
                                 1645+            PUNCH '//UTPRINT    DD  SYSOUT=&&SOUT'
                                 1646+            PUNCH '//SYSTEXT    DD  DSN=&&&&TXTPASS,DISP=(OLD,DELETE)'
                                 1647+            PUNCH '//REFERAL    DD  DSN=IMS2.REFERAL,DISP=SHR'
                                 1648+            PUNCH '//SEQBLKS    DD  DSN=&&&&BLKS,DISP=(NEW,PASS),UNIT=SYSDA,X
                                    +                   SPACE=(CYL,(2,2))'
                                 1649+            PUNCH '//FMTINDX    DD  DSN=IMS2.FMTINDX,DISP=OLD'
                                 1650+            PUNCH '//FMTINDXO DD  DSN=IMS2.FMTINDXO,DISP=OLD'
                                 1651+            PUNCH '//FMTDB      DD  DSN=IMS2.FORMATS,DISP=OLD'
                                 1652+            PUNCH '//MSGINDX    DD  DSN=IMS2.MSGINDX,DISP=OLD'
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT                                              F01OCT71  10/02/72
                                  1653+            PUNCH '//MSGINDXO DD  DSN=IMS2.MSGINDXO,DISP=OLD'
                                  1654+            PUNCH '//MSGDB    DD  DSN=IMS2.MESSAGES,DISP=OLD'
                                  1655+            PUNCH '//PHASE4 EXEC  PGM=DFSUNU30,REGION=&&RGN,PARM=''COMPRESX
                                     +                   S'''
                                  1656+            PUNCH '//SEQBLKS  DD  DSN=&&&&BLKS,DISP=(OLD,DELETE)'
                                  1657+            PUNCH '//SYSPRINT DD  SYSOUT=&&SOUT'
                                  1658+            PUNCH '//UTPRINT  DD  SYSOUT=&&SOUT'
                                  1659+            PUNCH '//FORMAT   DD  DSN=IMS2.FORMAT,DISP=OLD'
                                  1660+            PUNCH '//SYSUDUMP DD  SYSOUT=&&SOUT'
                                  1661+            PUNCH '//DUMMY    DD  DUMMY,DCB=BLKSIZE=80'
                                  1662+            PUNCH './         ADD   NAME=FMTINIT'
                                  1663+            PUNCH './         NUMBER NEW1=10,INCR=10'
                                  1664+            PUNCH '//IMSFMT JOB 1,IMS,MSGLEVEL=1'
                                  1665+            PUNCH '//STEP1  EXEC  PGM=IEWL,REGION=130K,'
                                  1666+            PUNCH '//             PARM=''XREF,LIST,LET,NCAL,DCBS'''
                                  1667+            PUNCH '//SYSPRINT DD  SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZX
                                     +                   E=605),'
                                  1668+            PUNCH '// SPACE=(605,(10,10),RLSE,,ROUND)'
                                  1669+            PUNCH '//LOAD     DD  DSN=IMS2.LOAD,DISP=SHR'
                                  1670+            PUNCH '//SYSLMOD DD   DSN=IMS2.PSBLIB,DISP=OLD'
                                  1671+            PUNCH '//SYSUT1  DD   UNIT=SYSDA,SPACE=(CYL,(5,1))'
                                  1672+            PUNCH '//SYSLIN  DD   *'
                                  1673+            PUNCH '  INCLUDE LOAD(DFSUTS10)'
                                  1674+            PUNCH '  NAME DFSUTS10(R)'
                                  1675+            PUNCH '  INCLUDE LOAD(DFSUTS20)'
                                  1676+            PUNCH '  NAME DFSUTS20(R)'
                                  1677+            PUNCH '//STEP2  EXEC  PGM=IEWL,REGION=130K,'
                                  1678+            PUNCH '//             PARM=''XREF,LIST,LET,NCAL,DCBS'''
                                  1679+            PUNCH '//SYSPRINT DD  SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZX
                                     +                   E=605),'
                                  1680+            PUNCH '// SPACE=(605,(10,10),RLSE,,ROUND)'
                                  1681+            PUNCH '//LOAD     DD  DSN=IMS2.LOAD,DISP=SHR'
                                  1682+            PUNCH '//SYSLMOD DD   DSN=IMS2.DBDLIB,DISP=OLD'
                                  1683+            PUNCH '//SYSUT1  DD   UNIT=SYSDA,SPACE=(CYL,(5,1))'
                                  1684+            PUNCH '//SYSLIN  DD   *'
                                  1685+            PUNCH '  INCLUDE LOAD(DFSUTS30)'
                                  1686+            PUNCH '  NAME DFSUTS30(R)'
                                  1687+            PUNCH '  INCLUDE LOAD(DFSUTS40)'
                                  1688+            PUNCH '  NAME DFSUTS40(R)'
                                  1689+            PUNCH '  INCLUDE LOAD(DFSUTS50)'
                                  1690+            PUNCH '  NAME DFSUTS50(R)'
                                  1691+            PUNCH '  INCLUDE LOAD(DFSUTS60)'
                                  1692+            PUNCH '  NAME DFSUTS60(R)'
                                  1693+            PUNCH '//STEP3  EXEC  PGM=IEBCOPY'
                                  1694+            PUNCH '//SYSPRINT DD  SYSOUT=A'
                                  1695+            PUNCH '//SYSUT1   DD  DSN=IMS2.LOAD,DISP=SHR'
                                  1696+            PUNCH '//SYSUT2   DD  DSN=IMS2.REFERAL,DISP=SHR'
                                  1697+            PUNCH '//SYSUT3   DD  UNIT=SYSDA,SPACE=(80,(10,20))'
                                  1698+            PUNCH '//SYSUT4   DD  UNIT=SYSDA,SPACE=(256,(2,4))'
                                  1699+            PUNCH '//SYSIN   DD   *'
                                  1700+            PUNCH '  COPY OUTDD=SYSUT2,INDD=((SYSUT1,R))'
                                  1701+            PUNCH '  SELECT MEMBER=DFSMI1'
                                  1702+            PUNCH '  SELECT MEMBER=DFSMI2'
                                  1703+            PUNCH '  SELECT MEMBER=DFSMI4'
                                  1704+            PUNCH '  SELECT MEMBER=DFSMO1'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT    SOURCE STATEMENT                                              F01OCT71  10/02/72
                                 1705+          PUNCH '   SELECT MEMBER=DFSM02'
                                 1706+          PUNCH '   SELECT MEMBER=DFSM03'
                                 1707+          PUNCH '   SELECT MEMBER=DFSM04'
                                 1708+          PUNCH '   SELECT MEMBER=DFSDF1'
                                 1709+          PUNCH '   SELECT MEMBER=DFSDF2'
                                 1710+          PUNCH '   SELECT MEMBER=DFSDF4'
                                 1711+          PUNCH '//STEP4  EXEC  PGM=DFSRRC00,REGION=200K,'
                                 1712+          PUNCH '// PARM=''DLI,DFSUTS00,DFSUTS10,4,00'''
                                 1713+          PUNCH '//IMS          DD  DSN=IMS2.PSBLIB,DISP=SHR'
                                 1714+          PUNCH '//             DD  DSN=IMS2.DBDLIB,DISP=SHR'
                                 1715+          PUNCH '//MSGINDX      DD  DSN=IMS2.MSGINDX,DISP=OLD'
                                 1716+          PUNCH '//MSGINDXO DD  DSN=IMS2.MSGINDXO,DISP=OLD'
                                 1717+          PUNCH '//MSGDB        DD  DSN=IMS2.MESSAGES,DISP=OLD'
                                 1718+          PUNCH '//FMTINDX      DD  DSN=IMS2.FMTINDX,DISP=OLD'
                                 1719+          PUNCH '//FMTINDXO DD  DSN=IMS2.FMTINDXO,DISP=OLD'
                                 1720+          PUNCH '//FMTDB        DD  DSN=IMS2.FORMATS,DISP=OLD'
                                 1721+          PUNCH '//SYSPRINT DD  SYSOUT=A,DCB=BLKSIZE=80'
                                 1722+          PUNCH '//SYSIN        DD  DSN=IMS2.GENLIB(DFSUTS70),DISP=SHR'
                                 1723+          PUNCH '//STEP5  EXEC  PGM=DFSRRC00,REGION=250K,'
                                 1724+          PUNCH '// PARM=''DLI,DFSUNU20,DFSUTS20,4,00'''
                                 1725+          PUNCH '//IMS          DD  DSN=IMS2.PSBLIB,DISP=SHR'
                                 1726+          PUNCH '//             DD  DSN=IMS2.DBDLIB,DISP=SHR'
                                 1727+          PUNCH '//IEFRDER  DD  DSN=&&&&IMSLOG,DISP=(,PASS),'
                                 1728+          PUNCH '// UNIT=SYSDA,SPACE=(CYL,(1,1)),DCB=DSORG=PS'
                                 1729+          PUNCH '//DFSLOGTT DD  DSN=IMS2.LOGT,DISP=SHR'
                                 1730+          PUNCH '//SYSUDUMP DD  SYSOUT=A'
                                 1731+          PUNCH '//SYSPRINT DD  SYSOUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZEX
                                     +          =133)'
                                 1732+          PUNCH '//UTPRINT  DD  SYSOUT=A'
                                 1733+          PUNCH '//SYSTEXT  DD  DSN=IMS2.GENLIB(DFSUTS90),DISP=SHR'
                                 1734+          PUNCH '//REFERAL  DD  DSN=IMS2.REFERAL,DISP=SHR'
                                 1735+          PUNCH '//MSGINDX  DD  DSN=IMS2.MSGINDX,DISP=OLD'
                                 1736+          PUNCH '//MSGINDXO DD  DSN=IMS2.MSGINDXO,DISP=OLD'
                                 1737+          PUNCH '//MSGDB        DD  DSN=IMS2.MESSAGES,DISP=OLD'
                                 1738+          PUNCH '//FMTINDX  DD  DSN=IMS2.FMTINDX,DISP=OLD'
                                 1739+          PUNCH '//FMTINDXO DD  DSN=IMS2.FMTINDXO,DISP=OLD'
                                 1740+          PUNCH '//FMTDB        DD  DSN=IMS2.FORMATS,DISP=OLD'
                                 1741+          PUNCH '//SEQBLKS  DD  DSN=&&&&BLKS,DISP=(NEW,PASS),'
                                 1742+          PUNCH '// UNIT=SYSDA,SPACE=(CYL,(1,1))'
                                 1743+          PUNCH '//STEP6  EXEC  PGM=DFSUNU30,REGION=200K'
                                 1744+          PUNCH '//SEQBLKS  DD  DSN=&&&&BLKS,DISP=(OLD,DELETE)'
                                 1745+          PUNCH '//SYSPRINT DD  SYSOUT=A'
                                 1746+          PUNCH '//UTPRINT  DD  SYSOUT=A'
                                 1747+          PUNCH '//SYSUDUMP DD  SYSOUT=A'
                                 1748+          PUNCH '//FORMAT   DD  DSN=IMS2.FORMAT,DISP=OLD'
                                 1749+          PUNCH '//DUMMY        DD  DUMMY,DCB=BLKSIZE=80'
                                 1750+          PUNCH './         ADD    NAME=IMSRDR'
                                 1751+          PUNCH './         NUMBER NEW1=10,INCR=10'
                                 1752+          PUNCH '//           PROC  MBR=IMSMSG'
                                 1753+          PUNCH '//IEFPROC EXEC PGM=IEFIRC,          READER FIRST LOAD'
                                 1754+          PUNCH '//                 REGION=48K,          READER BASIC REGION'
                                 1755+          PUNCH '//                 PARM=''00103005001024905210SYSDA    ''  X
                                     +          DEFAULT PARM FLD'
                                 1756+          PUNCH '//*                          BPPTTTOOOMMMIIICCCRLSSSSSSSS'
                                 1757+          PUNCH '//*'
```

```
LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F010CT71  10/02/72
                                   1758+         PUNCH '//*               B            DEFINED PROGRAMMER NAME && ACCX
                                     +                T NBR NOT NEEDED'
                                   1759+         PUNCH '//*               PP           PRIORITY=01'
                                   1760+         PUNCH '//*               TTT          JOB STEP INTERVAL=30 MINUTES'
                                   1761+         PUNCH '//*               OOO          PRIMARY SYSOUT SPACE=50 TRACKSX
                                     +                '
                                   1762+         PUNCH '//*               MMM          SECONDARY SYSOUT SPACE=10 TRACX
                                     +                KS'
                                   1763+         PUNCH '//*               III          READER/INTERPRETER DISPATCHINGX
                                     +                 PRIORITY=249'
                                   1764+         PUNCH '//*               CCC          JOB STEP DEFAULT REGION=52K'
                                   1765+         PUNCH '//*               R            DISPLAY && EXECUTE COMMANDS=1'
                                   1766+         PUNCH '//*               L            BYPASS LABEL=0'
                                   1767+         PUNCH '//*               SSSSSSSS  SYSOUT UNIT NAME=SYSDA'
                                   1768+         PUNCH '//*'
                                   1769+         PUNCH '//IEFRCER DD DSN=IMS2.PROCLIB(&&MBR),DISP=SHR,DCB=BUFNOX
                                     +                =1'
                                   1770+         PUNCH '//IEFPDSI DD   DSN=IMS2.PROCLIB,DISP=SHR      PROCEDURE X
                                     +                 LIBRARY'
                                   1771+         PUNCH '//         DD   DSN=SYS1.PROCLIB,DISP=SHR'
                                   1772+         PUNCH '//IEFDATA DD   UNIT=SYSDA,                            SPOOLX
                                     +                 DEVICE'
                                   1773+         PUNCH '//              SPACE=(80,(500,500),RLSE,CONTIG),  AMOUNX
                                     +                T'
                                   1774+         PUNCH '//              DCB=(BUFNO=2,LRECL=80,BLKSIZE=80,RECFM=FX
                                     +                B,BUFL=80)'
                                   1775+         PUNCH './         ADD  NAME=IMSWT000'
                                   1776+         PUNCH './         NUMBER NEW1=10,INCR=10'
                                   1777+         PUNCH '//         PROC  SOUT=A,JOB=A'
                                   1778+         PUNCH '//SPRTC JOB 1,IMS,MSGLEVEL=1,CLASS=&&JOB'
                                   1779+         PUNCH '//PRINT  EXEC  PGM=DFSUPRT0,REGION=30K'
                                   1780+         PUNCH '//STEPLIB  DD  DSN=IMS2.RESLIB,DISP=SHR'
                                   1781+         PUNCH '//SYSPRINT DD  SYSOUT=&&SOUT,DCB=BLKSIZE=1410'
                                   1782+         PUNCH '//SYSUDUMP DD  SYSOUT=&&SOUT'
                                   1783+         PUNCH '//SPOOL1  DD  DISP=SHR,DSN=IMS2.SYS01'
                                   1784+         PUNCH '//SPOOL2  DD  DISP=SHR,DSN=IMS2.SYS02'
                                   1785+         PUNCH '//SPOOL3  DD  DISP=SHR,DSN=IMS2.SYS03'
                                   1786+         PUNCH '//SPOOL4  DD  DISP=SHR,DSN=IMS2.SYS04'
                                   1787+         PUNCH './         ENDUP'
                                   1788+         PUNCH '/*'
                                   1789+         PUNCH '//STEP3 EXEC PGM=IEV90,REGION=200K,'
                                   1790+         PUNCH '// PARM=''LOAD,NODECK'''
                                   1791+         PUNCH '//SYSLIB DD DSN=IMS2.GENLIB,DISP=SHR'
                                   1792+         PUNCH '//         DD DSN=SYS1.MACLIB,DISP=SHR'
                                   1793+         PUNCH '//SYSLIN DD DSN=IMS2.BLKS50(DFSBBLKB),DISP=OLD'
                                   1794+         PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(BLKSIZE=605),'
                                   1795+         PUNCH '// SPACE=(605,(100,50),RLSE,,ROUND)'
                                   1796+         PUNCH '//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),'
                                   1797+         PUNCH '// SPACE=(CYL,(10,5))'
                                   1798+         PUNCH '//SYSIN DD *'
                                   1799+         PUNCH '         DFSPSBD DUMMY,2018,0'
                                   1800+         PUNCH '         SPACE 3'
                                   1801+         PUNCH '         SPACE 3'
                                   1802+         PUNCH '      IMSBATCH CENDA=Z5Z6,SVCNO=(234,248,213)'
                                   1803+         PUNCH '      DFSIPST REGIONS=1'
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F01OCT71  10/02/72
                                  1804+          PUNCH '         TITLE ''DFSIOS4C - OSAM IOB.'''
                                  1805+          PUNCH '       DFSIOIOB NUMIOB=1'
                                  1806+          PUNCH '       DFSAVARA 1,SECTYPE=CSECT'
                                  1807+          PUNCH '         TITLE ''DFSIWAIT - BATCH IWAIT ROUTINE.'''
                                  1808+          PUNCH 'DFSIWAIT CSECT'
                                  1809+          PUNCH '*****'
                                  1810+          PUNCH '*'
                                  1811+          PUNCH '*        IWAIT ROUTINE'
                                  1812+          PUNCH '*'
                                  1813+          PUNCH '*        THIS CODE PUTS THE BATCH TASK'
                                  1814+          PUNCH '*        INTO OS WAIT FOR THE DURATION'
                                  1815+          PUNCH '*        OF AN I/O OPERATION.'
                                  1816+          PUNCH '*'
                                  1817+          PUNCH '*        ASSUMES UPON ENTRY:'
                                  1818+          PUNCH '*        '
                                  1819+          PUNCH '*              R1   PST ADDR'
                                  1820+          PUNCH '*              R13  SAVE AREA ADDR'
                                  1821+          PUNCH '*'
                                  1822+          PUNCH '*****'
                                  1823+          PUNCH '         SPACE 3'
                                  1824+          PUNCH '         SAVE (14,12),,DFSIWAIT'
                                  1825+          PUNCH '         L     13,8(,13)'
                                  1826+          PUNCH '         WAIT  ECB=(1)'
                                  1827+          PUNCH '         L     13,4(,13)'
                                  1828+          PUNCH '         RETURN (14,12)'
                                  1829+          PUNCH '         TITLE ''DFSISCD - SYSTEM CONTENTS DIRECTORY (SX
                                      +                CD).'''
                                  1830+          PUNCH '      ISCD     SECTYPE=CSECT'
                                  1831+          PUNCH '         ORG   SCDDBFPL'
                                  1832+          PUNCH '         DC    F''8192''        DEFAULT BUFFER POOL SIZEX
                                      +                .'
                                  1833+          PUNCH '         ORG'
                                  1834+          PUNCH '      END'
                                  1835+          PUNCH '/*'
                                  1836+          PUNCH '//STEP4 EXEC PGM=IEV90,REGION=200K,'
                                  1837+          PUNCH '// PARM=''LOAD,NODECK'''
                                  1838+          PUNCH '//SYSLIB DD DSN=IMS2.GENLIB,DISP=SHR'
                                  1839+          PUNCH '//        DD DSN=SYS1.MACLIB,DISP=SHR'
                                  1840+          PUNCH '//SYSLIN DD DSN=IMS2.BLKS50(DFSIDLIO),DISP=OLD'
                                  1841+          PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(BLKSIZE=605),'
                                  1842+          PUNCH '// SPACE=(605,(100,50),RLSE,,ROUND)'
                                  1843+          PUNCH '//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),'
                                  1844+          PUNCH '// SPACE=(CYL,(10,5))'
                                  1845+          PUNCH '//SYSIN DD *'
                                  1846+          PUNCH 'PSBD7    DFSPSBD  CONTEST,8000,0'
                                  1847+          PUNCH 'PSBD1    DFSPSBD  DFSSAM02,8000,0'
                                  1848+          PUNCH 'PSBD2    DFSPSBD  DFSSAM03,8000,0'
                                  1849+          PUNCH 'PSBD3    DFSPSBD  DFSSAM04,8000,0'
                                  1850+          PUNCH 'PSBD4    DFSPSBD  DFSSAM05,8000,0'
                                  1851+          PUNCH 'PSBD5    DFSPSBD  DFSSAM06,8000,0'
                                  1852+          PUNCH 'PSBD6    DFSPSBD  DFSSAM07,8000,0'
                                  1853+          PUNCH 'PSBD8    DFSPSBD  ENQOSK41,8000,0'
                                  1854+          PUNCH 'PSBD9    DFSPSBD  HDTASK01,8000,0'
                                  1855+          PUNCH 'PSBD10   DFSPSBD  HHBLSK41,4000,0'
                                  1856+          PUNCH 'PSBD11   DFSPSBD  HHBLSK42,4000,0'
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT                                              F01OCT71  10/02/72

                                  1857+          PUNCH 'PSBD12      DFSPSBD   HHBLSK43,4000,0'
                                  1858+          PUNCH 'PSBD13      DFSPSBD   HHTASK41,8000,0'
                                  1859+          PUNCH 'PSBD14      DFSPSBD   HHTASK42,8000,0'
                                  1860+          PUNCH 'PSBD15      DFSPSBD   HHTASK43,8000,0'
                                  1861+          PUNCH 'PSBD16      DFSPSBD   HIBASK41,4000,0'
                                  1862+          PUNCH 'PSBD20      DFSPSBD   HIMAJCC1,8000,0'
                                  1863+          PUNCH 'PSBD21      DFSPSBD   HIMAJCC2,8000,0'
                                  1864+          PUNCH 'PSBD22      DFSPSBD   HIMAJCO3,8000,0'
                                  1865+          PUNCH 'PSBD23      DFSPSBD   HIMALMO1,8000,0'
                                  1866+          PUNCH 'PSBD17      DFSPSBD   HITASK41,8000,0'
                                  1867+          PUNCH 'PSBD18      DFSPSBD   HITASK42,8000,0'
                                  1868+          PUNCH 'PSBD19      DFSPSBD   HSBASK41,4000,0'
                                  1869+          PUNCH 'PSBD24      DFSPSBD   LKMDFS00,8000,0'
                                  1870+          PUNCH 'PSBD25      DFSPSBD   LKMDFS10,4000,0'
                                  1871+          PUNCH 'PSBD26      DFSPSBD   MR1,8000,0'
                                  1872+          PUNCH 'PSBD27      DFSPSBD   MR2,8000,0'
                                  1873+          PUNCH 'PSBD28      DFSPSBD   MR3,8000,0'
                                  1874+          PUNCH 'PSBD29      DFSPSBD   MR4,8000,0'
                                  1875+          PUNCH 'PSBD30      DFSPSBD   MR5,8000,0'
                                  1876+          PUNCH 'PSBD31      DFSPSBD   MR6,8000,0'
                                  1877+          PUNCH 'PSBD32      DFSPSBD   MR7,8000,0'
                                  1878+          PUNCH 'PSBD33      DFSPSBD   MR8,8000,0'
                                  1879+          PUNCH 'PSBD34      DFSPSBD   MR9,8000,0'
                                  1880+          PUNCH '           DFSSMB   ADDINV,A70002,4540,65535,65535,0,5,0,PX
                                      +                SBD3 '
                                  1881+          PUNCH '           DFSSMB   ADDPART,A70002,4540,65535,65535,0,4,0,X
                                      +                PSBD3 '
                                  1882+          PUNCH '           DFSSMB   CLOSE,A70002,4540,65535,65535,0,3,0,PSX
                                      +                BD4 '
                                  1883+          PUNCH '           DFSSMB   CONV1,88FFFF,45C8,3,60,0,1,80,PSBD7 '
                                  1884+          PUNCH '           DFSSMB   CONV2,88FFFF,45E8,3,60,0,1,80,PSBD7 '
                                  1885+          PUNCH '           DFSSMB   DFS,C50005,4140,8,100,0,1,0,PSBD24 '
                                  1886+          PUNCH '           DFSSMB   DISBURSE,A70002,4540,65535,65535,0,2,0X
                                      +                ,PSBD5 '
                                  1887+          PUNCH '           DFSSMB   DLETINV,A70002,4540,65535,65535,0,5,0,X
                                      +                PSBD3 '
                                  1888+          PUNCH '           DFSSMB   DLETPART,A70002,4540,65535,65535,0,4,0X
                                      +                ,PSBD3 '
                                  1889+          PUNCH '           DFSSMB   DLI,A50005,8540,10,10,0,1,0,PSBD23 '
                                  1890+          PUNCH '           DFSSMB   DLN,800003,4540,10,100,0,1,0,PSBD23 '
                                  1891+          PUNCH '           DFSSMB   DSPALLI,A70002,4140,65535,65535,0,2,0,X
                                      +                PSBD6 '
                                  1892+          PUNCH '           DFSSMB   DSPINV,A70002,4140,65535,65535,0,4,0,PX
                                      +                SBD2 '
                                  1893+          PUNCH '           DFSSMB   ENQ,88FFFF,4540,65535,65535,0,1,0,PSBDX
                                      +                8 '
                                  1894+          PUNCH '           DFSSMB   ICS,C50005,4548,10,100,0,1,0,PSBD23 '
                                  1895+          PUNCH '           DFSSMB   IMS,52000A,0540,1,100,0,1,0,PSBD23 '
                                  1896+          PUNCH '           DFSSMB   LKM,00FFFF,0540,65535,65535,0,1,0,PSBDX
                                      +                25 '
                                  1897+          PUNCH '           DFSSMB   MR1,88FFFF,4540,65535,65535,0,1,0,PSBDX
                                      +                26 '
                                  1898+          PUNCH '           DFSSMB   MR2,88FFFF,4540,65535,65535,0,1,0,PSBDX
                                      +                27 '
                                  1899+          PUNCH '           DFSSMB   MR3,88FFFF,4540,65535,65535,0,1,0,PSBDX
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                            F01OCT71  1C/02/72
                                   +              28 '
                                1900+       PUNCH '            DFSSMB  MR4,88FFFF,4540,65535,65535,C,1,0,PSBDX
                                   +              29 '
                                1901+       PUNCH '            DFSSMB  MR5,88FFFF,4540,65535,65535,0,1,C,PSBDX
                                   +              30 '
                                1902+       PUNCH '            DFSSMB  MR6,88FFFF,454C,65535,65535,0,1,0,PSBDX
                                   +              31 '
                                1903+       PUNCH '            DFSSMB  MR7,88FFFF,4540,65535,65535,0,1,0,PSBDX
                                   +              32 '
                                1904+       PUNCH '            DFSSMB  MR8,88FFFF,4540,65535,65535,0,1,0,PSBDX
                                   +              33 '
                                1905+       PUNCH '            DFSSMB  MR9,88FFFF,4540,65535,65535,C,1,0,PSBDX
                                   +              34 '
                                1906+       PUNCH '            CFSSMB  PART,A70002,4140,65535,65535,0,4,0,PSBX
                                   +              D1  '
                                1907+       PUNCH '            CFSSMB  SKD1,88FFFF,4540,65535,65535,0,1,0,PSBX
                                   +              D9  '
                                1908+       PUNCH '            DFSSMB  SKH1,88FFFF,4540,65535,65535,0,1,0,PSBX
                                   +              D13 '
                                1909+       PUNCH '            DFSSMB  SKH2,88FFFF,4540,65535,65535,0,1,0,PSBX
                                   +              D14 '
                                1910+       PUNCH '            DFSSMB  SKH3,88FFFF,4540,65535,65535,0,1,C,PSBX
                                   +              D15 '
                                1911+       PUNCH '            DFSSMB  SKI1,88FFFF,4540,65535,65535,0,1,0,PSBX
                                   +              D17 '
                                1912+       PUNCH '            DFSSMB  SKI2,88FFFF,4540,65535,65535,0,1,0,PSBX
                                   +              D18 '
                                1913+       PUNCH '            DFSSMB  TPPL1,88FFFF,4540,65535,65535,0,1,0,PSX
                                   +              BD20 '
                                1914+       PUNCH '            DFSSMB  TPPL2,88FFFF,4540,65535,65535,0,1,0,PSX
                                   +              BD21 '
                                1915+       PUNCH '            DFSSMB  TUBE,88FFFF,45C8,65535,65535,0,1,10C,PX
                                   +              SBD22 '
                                1916+       PUNCH '            SPACE 3'
                                1917+       PUNCH '            ENTRY DFSLISTC'
                                1918+       PUNCH 'DFSLISTC DS     0D             USER SUPPLIED SMB EDITS'
                                1919+       PUNCH '            DFSDMD  DD41SK01,0,C,1,0'
                                1920+       PUNCH '            DFSDMD  DD41TS01,0,0,1,0'
                                1921+       PUNCH '            DFSDMD  DH41SK01,C,0,1,0'
                                1922+       PUNCH '            DFSDMD  DH41SK02,0,0,1,0'
                                1923+       PUNCH '            DFSDMD  DH41SK03,0,0,1,0'
                                1924+       PUNCH '            DFSDMD  DH41TS01,0,0,1,0'
                                1925+       PUNCH '            DFSDMD  DH41TS02,C,0,1,0'
                                1926+       PUNCH '            DFSDMD  DH41TS03,0,0,1,0'
                                1927+       PUNCH '            CFSDMD  DI21PART,0,0,1,C'
                                1928+       PUNCH '            DFSDMD  DI31LM01,0,0,1,0'
                                1929+       PUNCH '            DFSDMD  DI31PHC2,0,0,1,0'
                                1930+       PUNCH '            DFSDMD  DI41SK01,0,0,1,0'
                                1931+       PUNCH '            DFSDMD  DI42SK01,0,0,1,0'
                                1932+       PUNCH '            DFSDMD  DS40JC01,0,0,1,0'
                                1933+       PUNCH '            DFSDMD  DX41SK01,0,0,1,0'
                                1934+       PUNCH '            DFSDMD  DX41TSC1,0,0,1,C'
                                1935+       PUNCH '            END' 3852
                                1936+       PUNCH '/*'
                                1937+       PUNCH '//IMSGEN2 JOB (82C,6443),IMS,MSGLEVEL=1,MSGCLASS=A,CLASX
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                         F01OCT71  10/02/72
                                    +           S=D,PRTY=8'
                                 1938+        PUNCH '//STEP1 EXEC PGM=IEV90,REGION=200K,'
                                 1939+        PUNCH '// PARM=''LOAD,NODECK'''
                                 1940+        PUNCH '//SYSLIB DD DSN=IMS2.GENLIB,DISP=SHR'
                                 1941+        PUNCH '//         DD DSN=SYS1.MACLIB,DISP=SHR'
                                 1942+        PUNCH '//SYSLIN DD DSN=IMS2.BLKS50(DFSICLL0),DISP=OLD'
                                 1943+        PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(BLKSIZE=605),'
                                 1944+        PUNCH '// SPACE=(605,(100,50),RLSE,,ROUND)'
                                 1945+        PUNCH '//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),'
                                 1946+        PUNCH '// SPACE=(CYL,(10,5))'
                                 1947+        PUNCH '//SYSIN DD *'
                                 1948+        PUNCH '         DFSCLB 1,0,0,1,C008,0,CTB0-DFSICTB,0    SYSTEM X
                                    +           CONSOLE CLB.'
                                 1949+        PUNCH '.DFCL1  DFSCLB 2,DFDCB1,DFPL1,1,0008,C,CTB1-DFSICTB,0'
                                 1950+        PUNCH '          AGO    .DFCL2'
                                 1951+        PUNCH '.DFPL1  ANOP'
                                 1952+        PUNCH 'DFPL1 DFTRMLST WRAPLST,(C6,E2)'
                                 1953+        PUNCH '          AGO    .DFPL2'
                                 1954+        PUNCH '.DFCL2  DFSCLB 3,DFDCB1,DFPL2,2,80E8,0,CTB3-DFSICTB,0'
                                 1955+        PUNCH '          AGO    .DFCL3'
                                 1956+        PUNCH '.DFPL2  ANOP'
                                 1957+        PUNCH 'DFPL2 DFTRMLST WRAPLST,(45,46)'
                                 1958+        PUNCH '          AGO    .DFPL3'
                                 1959+        PUNCH '.LERB1 ANOP'
                                 1960+        PUNCH 'LERB1  LERB  2,(200,10,5,5)'
                                 1961+        PUNCH '          AGO    .LERB2'
                                 1962+        PUNCH '.DFCL3  DFSCLB 4,DFDCB2,0,1,00E800,0,CTB5-DFSICTB,0'
                                 1963+        PUNCH '          AGO    .DFCL4'
                                 1964+        PUNCH '.DFPL3  ANOP'
                                 1965+        PUNCH '          AGO    .DFPL4'
                                 1966+        PUNCH '.LERB2 ANOP'
                                 1967+        PUNCH 'LERB2  LERB  1,(200,10,5,5)'
                                 1968+        PUNCH '          AGO    .LERB3'
                                 1969+        PUNCH '.DFCL4  DFSCLB 5,DFDCB3,DFPL4,1,00E900,0,CTB6-DFSICTB,0X
                                    +           '
                                 1970+        PUNCH '          AGO    .DFCL5'
                                 1971+        PUNCH '.DFPL4  ANOP'
                                 1972+        PUNCH 'DFPL4  DFTRMLST AUTOWLST,(40407F7F2D,                X
                                    +                  CONTINUE'
                                 1973+        PUNCH '                     C1C17F7F2D,                      X
                                    +                 CONTINUE'
                                 1974+        PUNCH '                     3737373737)'
                                 1975+        PUNCH '          AGO    .DFPL5'
                                 1976+        PUNCH '.LERB3 ANOP'
                                 1977+        PUNCH 'LERB3  LERB  1,(200,10,5,5)'
                                 1978+        PUNCH '          AGO    .LERB4'
                                 1979+        PUNCH '.DFCL5  DFSCLB 6,DFDCB4,0,1,00E800,0,CTB12-DFSICTB,0'
                                 1980+        PUNCH '          AGO    .DFCL6'
                                 1981+        PUNCH '.DFPL5  ANOP'
                                 1982+        PUNCH '          AGO    .DFPL6'
                                 1983+        PUNCH '.LERB4 ANOP'
                                 1984+        PUNCH 'LERB4  LERB  1,(200,10,5,5)'
                                 1985+        PUNCH '          AGO    .LERB5'
                                 1986+        PUNCH '.DFCL6  DFSCLB 7,DFDCB5,0,1,00E800,0,CTB14-DFSICTB,0'
                                 1987+        PUNCH '          AGO    .DFCL7'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                   F01OCT71  10/02/72
                                 1988+         PUNCH '.DFPL6  ANOP'
                                 1989+         PUNCH '        AGO    .DFPL7'
                                 1990+         PUNCH '.LERB5 ANOP'
                                 1991+         PUNCH 'LERB5  LERB  1,(200,10,5,5)'
                                 1992+         PUNCH '        AGO    .LERB6'
                                 1993+         PUNCH '.DFCL7  DFSCLB 8,DFDCB6,DFPL7,1,00E800,0,CTB15-DFSICTB,X
                                     +               0'
                                 1994+         PUNCH '        AGO    .DFCL8'
                                 1995+         PUNCH '.DFPL7  ANOP'
                                 1996+         PUNCH 'DFPL7 DFTRMLST WRAPLST,(E215,E415)'
                                 1997+         PUNCH '        AGO    .DFPL8'
                                 1998+         PUNCH '.LERB6 ANOP'
                                 1999+         PUNCH 'LERB6  LERB  1,(200,10,5,5)'
                                 2000+         PUNCH '        AGO    .LERB7'
                                 2001+         PUNCH '.DFCL8  DFSCLB 9,DFDCB7,DFPL8,1,00E800,0,CTB17-DFSICTB,X
                                     +               0'
                                 2002+         PUNCH '        AGO    .DFCL9'
                                 2003+         PUNCH '.DFPL8  ANOP'
                                 2004+         PUNCH 'DFPL8 DFTRMLST WRAPLST,(41FF,40FF)'
                                 2005+         PUNCH '        AGO    .DFPL9'
                                 2006+         PUNCH '.LERB7 ANOP'
                                 2007+         PUNCH 'LERB7  LERB  1,(200,10,5,5)'
                                 2008+         PUNCH '        AGO    .LERB8'
                                 2009+         PUNCH '.DFCL9  DFSCLB 10,DFDCB8,DFPL9,1,80E900,0,CTB21-DFSICTBX
                                     +               ,0'
                                 2010+         PUNCH '        AGO    .DFCL10'
                                 2011+         PUNCH '.DFPL9  ANOP'
                                 2012+         PUNCH 'DFPL9  DFTRMLST AUTOWLST,(C1C1F52D,C1C1F62D,           X
                                     +                     CONTINUE'
                                 2013+         PUNCH '                     C1C1F72D,37373737)'
                                 2014+         PUNCH '        AGO    .DFPL10'
                                 2015+         PUNCH '.LERB8 ANOP'
                                 2016+         PUNCH 'LERB8  LERB  1,(200,10,5,5)'
                                 2017+         PUNCH '        AGO    .LERB9'
                                 2018+         PUNCH '.DFCL10  DFSCLB 11,DFDCB9,DFPL10,1,80E900,0,CTB22-DFSICX
                                     +               TB,0'
                                 2019+         PUNCH '        AGO    .DFCL11'
                                 2020+         PUNCH '.DFPL10  ANOP'
                                 2021+         PUNCH 'DFPL10 DFTRMLST AUTOWLST,(C1C1F02D,37373737)'
                                 2022+         PUNCH '        AGO    .DFPL11'
                                 2023+         PUNCH '.LERB9 ANOP'
                                 2024+         PUNCH 'LERB9  LERB  1,(200,10,5,5)'
                                 2025+         PUNCH '        AGO    .LERB10'
                                 2026+         PUNCH '.DFCL11  DFSCLB 12,DFDCB10,0,1,00E800,0,CTB25-DFSICTB,0X
                                     +               '
                                 2027+         PUNCH '        AGO    .DFCL12'
                                 2028+         PUNCH '.DFPL11  ANOP'
                                 2029+         PUNCH '        AGO    .DFPL12'
                                 2030+         PUNCH '.LERB10 ANOP'
                                 2031+         PUNCH 'LERB10  LERB  1,(200,10,5,5)'
                                 2032+         PUNCH '        AGO    .LERB11'
                                 2033+         PUNCH '.DFCL12  DFSCLB 13,DFDCB11,0,1,00E800,0,CTB26-DFSICTB,0X
                                     +               '
                                 2034+         PUNCH '        AGO    .DFCL13'
                                 2035+         PUNCH '.DFPL12  ANOP'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                         F01OCT71  10/02/72
                                 2036+         PUNCH '          AGO   .DFPL13'
                                 2037+         PUNCH '.LERB11 ANOP'
                                 2038+         PUNCH 'LERB11  LERB  1,(200,10,5,5)'
                                 2039+         PUNCH '          AGO   .LERB12'
                                 2040+         PUNCH '.DFCL13  DFSCLB 14,DFDCB12,0,1,00E800,0,CTB27-DFSICTB,0X
                                    +               '
                                 2041+         PUNCH '          AGO   .DFCL14'
                                 2042+         PUNCH '.DFPL13  ANOP'
                                 2043+         PUNCH '          AGO   .DFPL14'
                                 2044+         PUNCH '.LERB12 ANOP'
                                 2045+         PUNCH 'LERB12  LERB  1,(200,10,5,5)'
                                 2046+         PUNCH '          AGO   .LERB13'
                                 2047+         PUNCH '.DFCL14  DFSCLB 15,DFDCB13,DFPL14,1,00E800,0,CTB28-DFSIX
                                    +               CTB,0'
                                 2048+         PUNCH '          AGO   .DFCL15'
                                 2049+         PUNCH '.DFPL14  ANOP'
                                 2050+         PUNCH 'DFPL14 DFTRMLST WRAPLST,(64)'
                                 2051+         PUNCH '          AGO   .DFPL15'
                                 2052+         PUNCH '.LERB13 ANOP'
                                 2053+         PUNCH 'LERB13  LERB  1,(200,10,5,5)'
                                 2054+         PUNCH '          AGO   .LERB14'
                                 2055+         PUNCH '.DFCL15  DFSCLB 16,DFDCB14,0,1,00E800,0,CTB30-DFSICTB,0X
                                    +               '
                                 2056+         PUNCH '          AGO   .DFCL16'
                                 2057+         PUNCH '.DFPL15  ANOP'
                                 2058+         PUNCH '          AGO   .DFPL16'
                                 2059+         PUNCH '.LERB14 ANOP'
                                 2060+         PUNCH 'LERB14  LERB  1,(200,10,5,5)'
                                 2061+         PUNCH '          AGO   .LERB15'
                                 2062+         PUNCH '.DFCL16  DFSCLB 17,DFDCB15,0,1,00E800,0,CTB31-DFSICTB,0X
                                    +               '
                                 2063+         PUNCH '          AGO   .DFCL17'
                                 2064+         PUNCH '.DFPL16  ANOP'
                                 2065+         PUNCH '          AGO   .DFPL17'
                                 2066+         PUNCH '.LERB15 ANOP'
                                 2067+         PUNCH 'LERB15  LERB  1,(200,10,5,5)'
                                 2068+         PUNCH '          AGO   .LERB16'
                                 2069+         PUNCH '.DFCL17  DFSCLB 18,DFDCB16,0,1,00E800,0,CTB32-DFSICTB,0X
                                    +               '
                                 2070+         PUNCH '          AGO   .DFCL18'
                                 2071+         PUNCH '.DFPL17  ANOP'
                                 2072+         PUNCH '          AGO   .DFPL18'
                                 2073+         PUNCH '.LERB16 ANOP'
                                 2074+         PUNCH 'LERB16  LERB  1,(200,10,5,5)'
                                 2075+         PUNCH '          AGO   .LERB17'
                                 2076+         PUNCH '.DFCL18  DFSCLB 19,DFDCB17,0,1,00E800,0,CTB33-DFSICTB,4X
                                    +               '
                                 2077+         PUNCH '          AGO   .DFCL19'
                                 2078+         PUNCH '.DFPL18  ANOP'
                                 2079+         PUNCH '          AGO   .DFPL19'
                                 2080+         PUNCH '.LERB17 ANOP'
                                 2081+         PUNCH 'LERB17  LERB  1,(200,10,5,5)'
                                 2082+         PUNCH '          AGO   .LERB21'
                                 2083+         PUNCH '.DFCL19  DFSCLB 20,DFDCB21,0,1,00E800,0,CTB34-DFSICTB,0X
                                    +               '
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71  10/02/72
                                 2084+           PUNCH '          AGO    .DFCL20'
                                 2085+           PUNCH '.DFPL19   ANOP'
                                 2086+           PUNCH '          AGO    .DFPL20'
                                 2087+           PUNCH '.DFCL20   DFSCLB 21,DFDCB21,0,2,00E800,C,CTB35-DFSICTB,0X
                                     +                 '
                                 2088+           PUNCH '          AGO    .DFCL21'
                                 2089+           PUNCH '.DFPL20   ANOP'
                                 2090+           PUNCH '          AGO    .DFPL21'
                                 2091+           PUNCH '.LERB21 ANOP'
                                 2092+           PUNCH 'LERB21   LERB   2,(200,10,5,5)'
                                 2093+           PUNCH '          AGO    .LERB22'
                                 2094+           PUNCH '.DFCL21   DFSCLB 22,DFDCB22,0,1,00E800,0,CTB36-DFSICTB,0X
                                     +                 '
                                 2095+           PUNCH '          AGO    .DFCL22'
                                 2096+           PUNCH '.DFPL21   ANOP'
                                 2097+           PUNCH '          AGO    .DFPL22'
                                 2098+           PUNCH '.LERB22 ANOP'
                                 2099+           PUNCH 'LERB22   LERB   1,(200,10,5,5)'
                                 2100+           PUNCH '          AGO    .LERB23'
                                 2101+           PUNCH '.DFCL22   DFSCLB 23,DFDCB23,DFPL22,1,00E800,0,CTB37-DFSIX
                                     +                 CTB,0'
                                 2102+           PUNCH '          AGO    .DFCL23'
                                 2103+           PUNCH '.DFPL22   ANOP'
                                 2104+           PUNCH 'DFPL22 DFTRMLST DIALST,0,(E2)'
                                 2105+           PUNCH '          AGO    .DFPL23'
                                 2106+           PUNCH '.LERB23 ANOP'
                                 2107+           PUNCH 'LERB23   LERB   1,(200,10,5,5)'
                                 2108+           PUNCH '          AGO    .LERB24'
                                 2109+           PUNCH '.DFCL23   DFSCLB 24,DFDCB24,DFPL23,1,00E800,C,CTB38-DFSIX
                                     +                 CTB,0'
                                 2110+           PUNCH '          AGO    .DFCL24'
                                 2111+           PUNCH '.DFPL23   ANOP'
                                 2112+           PUNCH 'DFPL23 DFTRMLST DIALST,0,(E215)'
                                 2113+           PUNCH '          AGO    .DFPL24'
                                 2114+           PUNCH '.DFCL24   DFSCLB 25,DFDCB24,0,1,00E800,0,CTB39-DFSICTB,0X
                                     +                 '
                                 2115+           PUNCH '          TITLE ''DFSICLLO - COMMUNICATICN LINE POLLING X
                                     +                 LISTS.'''
                                 2116+           PUNCH '          AGO    .DFPL1'
                                 2117+           PUNCH '.DFPL24   ANOP'
                                 2118+           PUNCH '          TITLE ''DFSICLLO - COMMUNICATION LINE ERROR BLX
                                     +                 OCKS.'''
                                 2119+           PUNCH '          AGO    .LERB1'
                                 2120+           PUNCH '.LERB24 ANOP'
                                 2121+           PUNCH 'LERB24   LERB   1,(200,10,5,5)'
                                 2122+           PUNCH '          TITLE ''DFSICLLO - COMMUNICATION LINE GROUP DCX
                                     +                 B''''S.'''
                                 2123+           PUNCH 'DFDCB1 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB1,           X
                                     +                        CONTINUE'
                                 2124+           PUNCH '                     ERROPT=CTRW,DDNAME=DD2740S'
                                 2125+           PUNCH '          EJECT'
                                 2126+           PUNCH 'DFDCB2 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB2,           X
                                     +                        CONTINUE'
                                 2127+           PUNCH '                     ERROPT=CTRW,DDNAME=D2740NSC'
                                 2128+           PUNCH '          EJECT'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F01OCT71  10/02/72
                                 2129+         PUNCH 'DFDCB3 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB3,             X
                                    +                         CONTINUE'
                                 2130+         PUNCH '                    ERROPT=CT,DEVD=BS,DDNAME=DD3270R'
                                 2131+         PUNCH '             EJECT'
                                 2132+         PUNCH 'DFDCB4 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB4,             X
                                    +                         CONTINUE'
                                 2133+         PUNCH '                    ERROPT=CT,DDNAME=DD3270L'
                                 2134+         PUNCH '             EJECT'
                                 2135+         PUNCH 'DFDCB5 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB5,             X
                                    +                         CONTINUE'
                                 2136+         PUNCH '                    ERROPT=CT,DDNAME=DD327CL2'
                                 2137+         PUNCH '             EJECT'
                                 2138+         PUNCH 'DFDCB6 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB6,             X
                                    +                         CONTINUE'
                                 2139+         PUNCH '                    ERROPT=CTRW,DDNAME=DD1050'
                                 2140+         PUNCH '             EJECT'
                                 2141+         PUNCH 'DFDCB7 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB7,             X
                                    +                         CONTINUE'
                                 2142+         PUNCH '                    ERROPT=CTRW,DDNAME=DD2260R'
                                 2143+         PUNCH '             EJECT'
                                 2144+         PUNCH 'DFDCB8 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB8,             X
                                    +                         CONTINUE'
                                 2145+         PUNCH '                    ERROPT=CT,DEVD=BS,DDNAME=DD2770'
                                 2146+         PUNCH '             EJECT'
                                 2147+         PUNCH 'DFDCB9 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB9,             X
                                    +                         CONTINUE'
                                 2148+         PUNCH '                    ERROPT=CT,DEVD=BS,DDNAME=DD2980'
                                 2149+         PUNCH '             EJECT'
                                 2150+         PUNCH 'DFDCB10 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB10,           X
                                    +                           CONTINUE'
                                 2151+         PUNCH '                    ERROPT=CTW,DDNAME=DD2741'
                                 2152+         PUNCH '             EJECT'
                                 2153+         PUNCH 'DFDCB11 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB11,           X
                                    +                           CONTINUE'
                                 2154+         PUNCH '                    ERROPT=CTW,DDNAME=DD2741S'
                                 2155+         PUNCH '             EJECT'
                                 2156+         PUNCH 'DFDCB12 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB12,           X
                                    +                           CONTINUE'
                                 2157+         PUNCH '                    ERROPT=CTW,DDNAME=DD2741S2'
                                 2158+         PUNCH '             EJECT'
                                 2159+         PUNCH 'DFDCB13 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB13,           X
                                    +                           CONTINUE'
                                 2160+         PUNCH '                    ERROPT=CTRW,DDNAME=DD1030'
                                 2161+         PUNCH '             EJECT'
                                 2162+         PUNCH '             EXTRN DFSYSOUT '
                                 2163+         PUNCH 'DFDCB14 DCB DSORG=PS,MACRF=R,DEVD=DA,RECFM=F,BLKSIZE=80X
                                    +                               CONTINUE'
                                 2164+         PUNCH '                    DDNAME=SYSRDR1'
                                 2165+         PUNCH '             EJECT'
                                 2166+         PUNCH 'DFDCB15 DCB    DSORG=PS,MACRF=W,DEVD=DA,EXLST=DFSYSOUT, X
                                    +                             CONTINUE'
                                 2167+         PUNCH '                    DDNAME=SYSPRT1'
                                 2168+         PUNCH '             EJECT'
                                 2169+         PUNCH 'DFDCB16 DCB    DSORG=PS,MACRF=W,DEVD=DA,EXLST=DFSYSOUT, X
                                    +                             CONTINUE'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                             F01OCT71  1C/02/72
                                 2170+        PUNCH '                   DDNAME=SYSTAPE'
                                 2171+        PUNCH '              EJECT'
                                 2172+        PUNCH 'DFDCB17 DCB    DSORG=PS,MACRF=(R,W),DEVD=DA,EXLST=DFSYSOX
                                    +               UT,            CONTINUE'
                                 2173+        PUNCH '                   DDNAME=SYS1'
                                 2174+        PUNCH '              EJECT'
                                 2175+        PUNCH 'DFDCB18 DCB    DSORG=PS,MACRF=(R,W),DEVD=DA,EXLST=DFSYSOX
                                    +               UT,            CONTINUE'
                                 2176+        PUNCH '                   DDNAME=SYS2'
                                 2177+        PUNCH '              EJECT'
                                 2178+        PUNCH 'DFDCB19 DCB    DSORG=PS,MACRF=(R,W),DEVD=DA,EXLST=DFSYSOX
                                    +               UT,            CONTINUE'
                                 2179+        PUNCH '                   DDNAME=SYS3'
                                 2180+        PUNCH '              EJECT'
                                 2181+        PUNCH 'DFDCB20 DCB    DSORG=PS,MACRF=(R,W),DEVD=DA,EXLST=DFSYSOX
                                    +               UT,            CONTINUE'
                                 2182+        PUNCH '                   DDNAME=SYS4'
                                 2183+        PUNCH '              DC    A(0)      END OF SPOOL DCB SET '
                                 2184+        PUNCH '              EJECT'
                                 2185+        PUNCH 'DFDCB21 DCB    DSORG=PS,MACRF=(E),CENDA=Z6,XENDA=Z6,DDNAX
                                    +               ME=ARUDDGP1'
                                 2186+        PUNCH '              EJECT'
                                 2187+        PUNCH 'DFDCB22 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB22,          X
                                    +                              CONTINUE'
                                 2188+        PUNCH '                   ERROPT=CTRW,DDNAME=DTWX3335'
                                 2189+        PUNCH '              EJECT'
                                 2190+        PUNCH 'DFDCB23 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB23,          X
                                    +                              CONTINUE'
                                 2191+        PUNCH '                   ERROPT=CTRW,DDNAME=DD274CA'
                                 2192+        PUNCH '              EJECT'
                                 2193+        PUNCH 'DFDCB24 DCB    DSORG=CX,MACRF=(R,W),LERB=LERB24,          X
                                    +                              CONTINUE'
                                 2194+        PUNCH '                   ERROPT=CTRW,DDNAME=DD105CA'
                                 2195+        PUNCH '              TITLE ''DFSICLL0 - COMMUNICATION LINE GROUP OPX
                                    +               EN LIST.'''
                                 2196+        PUNCH '              ENTRY DFSICDB'
                                 2197+        PUNCH 'DFSICDB  DS    0D'
                                 2198+        PUNCH '              DC    AL1(0),AL3(DFDCB1)'
                                 2199+        PUNCH '              DC    AL1(0),AL3(DFDCB2)'
                                 2200+        PUNCH '              DC    AL1(0),AL3(DFDCB3)'
                                 2201+        PUNCH '              DC    AL1(0),AL3(DFDCB4)'
                                 2202+        PUNCH '              DC    AL1(0),AL3(DFDCB5)'
                                 2203+        PUNCH '              DC    AL1(0),AL3(DFDCB6)'
                                 2204+        PUNCH '              DC    AL1(0),AL3(DFDCB7)'
                                 2205+        PUNCH '              DC    AL1(0),AL3(DFDCB8)'
                                 2206+        PUNCH '              DC    AL1(0),AL3(DFDCB9)'
                                 2207+        PUNCH '              DC    AL1(0),AL3(DFDCB10)'
                                 2208+        PUNCH '              DC    AL1(0),AL3(DFDCB11)'
                                 2209+        PUNCH '              DC    AL1(0),AL3(DFDCB12)'
                                 2210+        PUNCH '              DC    AL1(0),AL3(DFDCB13)'
                                 2211+        PUNCH '              DC    AL1(0),AL3(DFDCB14)'
                                 2212+        PUNCH '              DC    AL1(15),AL3(DFDCB15)'
                                 2213+        PUNCH '              DC    AL1(15),AL3(DFDCB16)'
                                 2214+        PUNCH '              DC    AL1(3),AL3(DFDCB17)'
                                 2215+        PUNCH '              DC    AL1(3),AL3(DFDCB18)'
```

```
LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    F01OCT71  10/02/72
                                   2216+         PUNCH '            DC      AL1(3),AL3(DFDCB19)'
                                   2217+         PUNCH '            DC      AL1(3),AL3(DFDCB20)'
                                   2218+         PUNCH '            DC      AL1(1),AL3(DFDCB21)'
                                   2219+         PUNCH '            DC      AL1(0),AL3(DFDCB22)'
                                   2220+         PUNCH '            DC      AL1(C),AL3(DFDCB23)'
                                   2221+         PUNCH '            DC      AL1(128),AL3(DFDCB24)'
                                   2222+         PUNCH 'CTB0 DFSCTB 0,1,000000,000003C0,00,C0,CNT64-DFSICNT,0,0X
                                      +               ,0'
                                   2223+         PUNCH 'CTB1       DFSCTB 1,2,C68181,400001C0,C0,C0,CNT1-DFSICNT,X
                                      +                            CONTINUE'
                                   2224+         PUNCH '                COB1-DFSICOB,0,C'
                                   2225+         PUNCH '            ENTRY DFSCTBMT'
                                   2226+         PUNCH 'DFSCTBMT EQU   CTB1'
                                   2227+         PUNCH 'CTB2       DFSCTB 1,2,E28181,00000300,0C,C0,CNT3-DFSICNT,X
                                      +                            CONTINUE'
                                   2228+         PUNCH '                COB1-DFSICOB,0,0'
                                   2229+         PUNCH 'CTB3       DFSCTB 2,3,458181,00000100,00,00,CNT4-DFSICNT,X
                                      +                            CONTINUE'
                                   2230+         PUNCH '                COB2-DFSICOB,0,0'
                                   2231+         PUNCH 'CTB4       DFSCTB 3,3,468181,000003C0,C0,00,CNT5-DFSICNT,X
                                      +                            CONTINUE'
                                   2232+         PUNCH '                COB3-DFSICOB,0,0'
                                   2233+         PUNCH 'CTB5       DFSCTB 4,4,E28181,00C00300,0C,CC,CNT6-DFSICNT,X
                                      +                            CONTINUE'
                                   2234+         PUNCH '                COB4-DFSICOB,0,0'
                                   2235+         PUNCH 'CTB6       DFSCTB 5,5,404081,0000C100,D0,0A,CNT7-DFSICNT,X
                                      +                            CONTINUE'
                                   2236+         PUNCH '                COB5-DFSICOB,0,CIB1-DFSICIB'
                                   2237+         PUNCH 'CTB7       DFSCTB 5,5,C14081,00000100,18,2A,CNT9-DFSICNT,X
                                      +                            CONTINUE'
                                   2238+         PUNCH '                COB5-DFSICOB,0,CIB2-DFSICIB'
                                   2239+         PUNCH 'CTB8       DFSCTB 5,5,C1C181,00C00100,18,2A,CNT10-DFSICNTX
                                      +               ,             CONTINUE'
                                   2240+         PUNCH '                COB5-DFSICOB,0,CIB3-DFSICIB'
                                   2241+         PUNCH 'CTB9       DFSCTB 5,5,C1C281,00000100,1C,2A,CNT11-DFSICNTX
                                      +               ,             CONTINUE'
                                   2242+         PUNCH '                COB5-DFSICOB,C,CIB4-DFSICIB'
                                   2243+         PUNCH 'CTB10      DFSCTB 5,5,C1C381,00000140,88,2A,CNT12-DFSICNTX
                                      +               ,             CONTINUE'
                                   2244+         PUNCH '                COB5-DFSICOB,0,CIB5-DFSICIB'
                                   2245+         PUNCH 'CTB11      DFSCTB 5,5,C1C481,00C00340,88,2A,CNT13-DFSICNTX
                                      +               ,             CONTINUE'
                                   2246+         PUNCH '                COB5-DFSICOB,0,CIB6-DFSICIB'
                                   2247+         PUNCH 'CTB12      DFSCTB 6,6,108181,0000010C,08,2A,CNT14-DFSICNTX
                                      +               ,             CONTINUE'
                                   2248+         PUNCH '                COB6-DFSICOB,0,CIB7-DFSICIB'
                                   2249+         PUNCH 'CTB13      DFSCTB 6,6,128181,00C00300,0C,2A,CNT15-DFSICNTX
                                      +               ,             CONTINUE'
                                   2250+         PUNCH '                COB6-DFSICOB,0,CIB8-DFSICIB'
                                   2251+         PUNCH 'CTB14      DFSCTB 7,7,118181,00000340,88,2A,CNT16-DFSICNTX
                                      +               ,             CONTINUE'
                                   2252+         PUNCH '                COB7-DFSICOB,0,CIB9-DFSICIB'
                                   2253+         PUNCH 'CTB15      DFSCTB 8,8,E20281,00000100,20,2A,CNT17-DFSICNTX
                                      +               ,             CONTINUE'
                                   2254+         PUNCH '                COB8-DFSICOB,C,0'
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                     F01OCT71  10/02/72
                                  2255+         PUNCH 'CTB16    CFSCTB 8,8,E40281,00C00300,20,0A,CNT18-DFSICNTX
                                     +                ,            CCNTINUE'
                                  2256+         PUNCH '                 COB8-DFSICOB,0,0'
                                  2257+         PUNCH 'CTB17    DFSCTB 9,9,415181,00G00100,44,00,CNT2C-DFSICNTX
                                     +                ,            CONTINUE'
                                  2258+         PUNCH '                 COB9-DFSICOB,0,0'
                                  2259+         PUNCH 'CTB18    DFSCTB 10,9,40A081,00000100,40,00,CNT21-DFSICNX
                                     +                T,            CONTINUE'
                                  2260+         PUNCH '                 COB9-DFSICOB,0,0'
                                  2261+         PUNCH 'CTB19    DFSCTB 10,9,4CA181,00000100,40,00,CNT22-DFSICNX
                                     +                T,            CONTINUE'
                                  2262+         PUNCH '                 COB9-DFSICOB,0,0'
                                  2263+         PUNCH 'CTB20    DFSCTB 10,9,4CA481,C000034C,C0,00,CNT23-DFSICNX
                                     +                T,            CONTINUE'
                                  2264+         PUNCH '                 COB9-DFSICOB,C,C'
                                  2265+         PUNCH 'CTB21    DFSCTB 11,10,C18181,00C04700,80,02,CNT24-DFSICX
                                     +                NT,            CONTINUE'
                                  2266+         PUNCH '                 COB10-DFSICOB,CXB1-DFSICXB,C'
                                  2267+         PUNCH 'CTB22    DFSCTB 12,11,C140F4,00000100,10,00,CNT27-DFSICX
                                     +                NT,            CONTINUE'
                                  2268+         PUNCH '                 COB11-DFSICOB,0,C'
                                  2269+         PUNCH 'CTB23    DFSCTB 13,11,C1F1F5,00000100,01,00,CNT30-DFSICX
                                     +                NT,            CONTINUE'
                                  2270+         PUNCH '                 COB12-DFSICOB,0,0'
                                  2271+         PUNCH 'CTB24    DFSCTB 14,11,C1F2F6,000C030C,0A,00,CNT34-DFSICX
                                     +                NT,            CONTINUE'
                                  2272+         PUNCH '                 COB13-DFSICOB,0,0'
                                  2273+         PUNCH 'CTB25    DFSCTB 15,12,C18181,000003C0,C0,00,CNT39-DFSICX
                                     +                NT,            CONTINUE'
                                  2274+         PUNCH '                 COB14-DFSICOB,0,0'
                                  2275+         PUNCH 'CTB26    DFSCTB 16,13,508181,00C0030C,CC,0C,CNT40-DFSICX
                                     +                NT,            CONTINUE'
                                  2276+         PUNCH '                 COB15-DFSICOB,0,0'
                                  2277+         PUNCH 'CTB27    DFSCTB 16,14,538181,00C00300,0A,00,CNT41-DFSICX
                                     +                NT,            CONTINUE'
                                  2278+         PUNCH '                 COB16-DFSICOB,0,0'
                                  2279+         PUNCH 'CTB28    DFSCTB 17,15,648181,C000C100,C0,00,CNT42-DFSICX
                                     +                NT,            CONTINUE'
                                  2280+         PUNCH '                 COB17-DFSICOB,0,0'
                                  2281+         PUNCH 'CTB29    DFSCTB 17,15,648181,000CC70C,C0,00,CNT42-DFSICX
                                     +                NT,            CONTINUE'
                                  2282+         PUNCH '                 COB17-DFSICOB,0,0'
                                  2283+         PUNCH 'CTB30    DFSCTB 18,16,008181,0000C7C0,00,00,CNT42-DFSICX
                                     +                NT,            CONTINUE'
                                  2284+         PUNCH '                 COB18-DFSICOB,C,C'
                                  2285+         PUNCH 'CTB31    DFSCTB 19,17,C08181,C000034C,00,0C,CNT44-DFSICX
                                     +                NT,            CONTINUE'
                                  2286+         PUNCH '                 COB19-DFSICOB,C,0'
                                  2287+         PUNCH 'CTB32    DFSCTB 20,18,008181,C0000340,C0,0C,CNT45-DFSICX
                                     +                NT,            CONTINUE'
                                  2288+         PUNCH '                 COB20-DFSICOB,0,0'
                                  2289+         PUNCH 'CTB33    DFSCTB 21,19,00000C,000C0300,C0,00,CNT46-DFSICX
                                     +                NT,            CONTINUE'
                                  2290+         PUNCH '                 COB21-DFSICOB,0,C'
                                  2291+         PUNCH 'CTB34    DFSCTB 22,20,C18181,C0004300,40,00,CNT48-DFSICX
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F01OCT71  10/02/72
                                    +            NT,          CONTINUE'
                                 2292+    PUNCH '               COB22-DFSICOB,0,0'
                                 2293+    PUNCH 'CTB35    DFSCTB 23,21,C48181,C0000300,00,00,CNT49-DFSICX
                                    +            NT,          CONTINUE'
                                 2294+    PUNCH '               COB23-DFSICOB,0,0'
                                 2295+    PUNCH 'CTB36    DFSCTB 24,22,E28181,00000300,00,00,CNT50-DFSICX
                                    +            NT,          CONTINUE'
                                 2296+    PUNCH '               COB24-DFSICOB,0,0'
                                 2297+    PUNCH 'CTB37    DFSCTB 25,23,E28181,00000300,00,00,CNT51-DFSICX
                                    +            NT,          CONTINUE'
                                 2298+    PUNCH '               COB25-DFSICOB,C,0'
                                 2299+    PUNCH 'CTB38    DFSCTB 26,24,E2C281,00000300,20,0C,CNT52-DFSICX
                                    +            NT,          CONTINUE'
                                 2300+    PUNCH '               COB26-DFSICOB,0,C'
                                 2301+    PUNCH 'CTB39    DFSCTB 0,25,000281,200C0100,C0,00,CNT53-DFSICNX
                                    +            T,          CONTINUE'
                                 2302+    PUNCH '               COB27-DFSICOB,C,C'
                                 2303+    PUNCH 'CTB40    DFSCTB 0,25,000281,20000100,C0,00,CNT56-DFSICNX
                                    +            T,          CONTINUE'
                                 2304+    PUNCH '               COB27-DFSICOB,C,C'
                                 2305+    PUNCH 'CTB41    DFSCTB 0,25,000281,200C0100,00,00,CNT58-DFSICNX
                                    +            T,          CONTINUE'
                                 2306+    PUNCH '               COB27-DFSICOB,C,0'
                                 2307+    PUNCH 'CTB42    DFSCTB 0,25,000281,200C0300,C0,00,CNT60-DFSICNX
                                    +            T,          CONTINUE'
                                 2308+    PUNCH '               COB27-DFSICOB,0,0'
                                 2309+    PUNCH '          DFSCOBS 27'
                                 2310+    PUNCH 'CXB1 DFSCXB 0210,0000,0019,0000'
                                 2311+    PUNCH 'CIB1 DFSCIB 00C4'
                                 2312+    PUNCH 'CIB2 DFSCIB 02C7'
                                 2313+    PUNCH 'CIB3 DFSCIB 02C7'
                                 2314+    PUNCH 'CIB4 DFSCIB 00C7'
                                 2315+    PUNCH 'CIB5 DFSCIB 0360'
                                 2316+    PUNCH 'CIB6 DFSCIB 0360'
                                 2317+    PUNCH 'CIB7 DFSCIB 0240'
                                 2318+    PUNCH 'CIB8 DFSCIB 00C7'
                                 2319+    PUNCH 'CIB9 DFSCIB 0340'
                                 2320+    PUNCH 'CNT21    DFSCNT AA,0000,0,CTB18-DFSICTB,65535'
                                 2321+    PUNCH 'CNT22    DFSCNT AB,0000,0,CTB19-DFSICTB,65535'
                                 2322+    PUNCH 'CNT34    DFSCNT ATELLER,008C,0,CTB24-DFSICTB,CNT35-DFSIX
                                    +            CNT'
                                 2323+    PUNCH 'CNT29    DFSCNT A29801,0080,0,CTB22-DFSICTB,65535'
                                 2324+    PUNCH 'CNT31    DFSCNT A29802,0080,0,CTB23-DFSICTB,CNT32-DFSICX
                                    +            NT'
                                 2325+    PUNCH 'CNT35    DFSCNT BTELLER,0080,0,CTB24-DFSICTB,CNT36-DFSIX
                                    +            CNT'
                                 2326+    PUNCH 'CNT54    DFSCNT CAROL,0002,0,CTB39-DFSICTB,CNT55-DFSICNX
                                    +            T'
                                 2327+    PUNCH 'CNT32    DFSCNT COMMON,0080,0,CTB23-DFSICTB,CNT33-DFSICX
                                    +            NT'
                                 2328+    PUNCH 'CNT1     DFSCNT CTRL,400C,0,CTB1-DFSICTB,CNT2-DFSICNT'
                                 2329+    PUNCH 'CNT53    DFSCNT ELEANOR,0002,0,CTB39-DFSICTB,CNT54-DFSIX
                                    +            CNT'
                                 2330+    PUNCH 'CNT23    DFSCNT ERNE,0000,0,CTB20-DFSICTB,65535'
                                 2331+    PUNCH 'CNT27    DFSCNT HOMER1,0080,0,CTB22-DFSICTB,CNT28-DFSICX
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    FC10CT71  10/02/72
                                     +              NT'
                                  2332+             PUNCH 'CNT38       DFSCNT HOMER2,0080,0,CTB24-DFSICTB,65535'
                                  2333+             PUNCH 'CNT33       DFSCNT HOMER4,0080,1,CTB23-DFSICTB,65535'
                                  2334+             PUNCH 'CNT2        DFSCNT HOWARD,0000,0,CTB1-DFSICTB,65535'
                                  2335+             PUNCH 'CNT40       DFSCNT INQU0393,0001,0,CTB26-DFSICTB,65535'
                                  2336+             PUNCH 'CNT41       DFSCNT INQU0401,0001,0,CTB27-DFSICTB,65535'
                                  2337+             PUNCH 'CNT48       DFSCNT INQU0464,0001,0,CTB34-DFSICTB,65535'
                                  2338+             PUNCH 'CNT49       DFSCNT INQU0469,0001,0,CTB35-DFSICTB,65535'
                                  2339+             PUNCH 'CNT50       DFSCNT INQU0477,0001,0,CTB36-DFSICTB,65535'
                                  2340+             PUNCH 'CNT51       DFSCNT INQU0485,0001,0,CTB37-DFSICTB,65535'
                                  2341+             PUNCH 'CNT52       DFSCNT INQU0493,0001,0,CTB38-DFSICTB,65535'
                                  2342+             PUNCH 'CNT36       DFSCNT JTELLER,0080,0,CTB24-DFSICTB,CNT37-DFSIX
                                     +              CNT'
                                  2343+             PUNCH 'CNT37       DFSCNT KTELLER,0080,0,CTB24-DFSICTB,CNT38-DFSIX
                                     +              CNT'
                                  2344+             PUNCH 'CNT3        DFSCNT LARRY,0000,0,CTB2-DFSICTB,65535'
                                  2345+             PUNCH 'CNT4        DFSCNT MODEL2,0000,0,CTB3-DFSICTB,65535'
                                  2346+             PUNCH 'CNT5        DFSCNT MODEL2K,0000,0,CTB4-DFSICTB,65535'
                                  2347+             PUNCH 'CNT28       DFSCNT N29801,0080,0,CTB22-DFSICTB,CNT29-DFSICX
                                     +              NT'
                                  2348+             PUNCH 'CNT30       DFSCNT N29802,0080,0,CTB23-DFSICTB,CNT31-DFSICX
                                     +              NT'
                                  2349+             PUNCH 'CNT17       DFSCNT PRINTER1,0000,0,CTB15-DFSICTB,65535'
                                  2350+             PUNCH 'CNT18       DFSCNT PRINTER2,0000,0,CTB16-DFSICTB,CNT19-DFSX
                                     +              ICNT'
                                  2351+             PUNCH 'CNT19       DFSCNT PRINTER3,0000,1,CTB16-DFSICTB,65535'
                                  2352+             PUNCH 'CNT44       DFSCNT PRTSYS,0000,0,CTB31-DFSICTB,65535'
                                  2353+             PUNCH 'CNT55       DFSCNT SHARRON,0002,0,CTB39-DFSICTB,65535'
                                  2354+             PUNCH 'CNT46       DFSCNT SP1,0000,0,CTB33-DFSICTB,CNT47-DFSICNT'
                                  2355+             PUNCH 'CNT47       DFSCNT SP2,0000,0,CTB33-DFSICTB,65535'
                                  2356+             PUNCH 'CNT63       DFSCNT SWCDPNCH,0002,3,CTB42-DFSICTB,65535'
                                  2357+             PUNCH 'CNT61       DFSCNT SWPRNTR2,0002,2,CTB42-DFSICTB,CNT62-DFSX
                                     +              ICNT'
                                  2358+             PUNCH 'CNT62       DFSCNT SWTPPNCH,0002,2,CTB42-DFSICTB,CNT63-DFSX
                                     +              ICNT'
                                  2359+             PUNCH 'CNT60       DFSCNT SW1050,0002,0,CTB42-DFSICTB,CNT61-DFSICX
                                     +              NT'
                                  2360+             PUNCH 'CNT45       DFSCNT TAPESYS,0000,0,CTB32-DFSICTB,65535'
                                  2361+             PUNCH 'CNT42       DFSCNT T1033A,0000,0,CTB28-DFSICTB,CNT43-DFSICX
                                     +              NT'
                                  2362+             PUNCH 'CNT43       DFSCNT T1033B,0000,0,CTB28-DFSICTB,65535'
                                  2363+             PUNCH 'CNT20       DFSCNT T2265XCL,0000,0,CTB17-DFSICTB,65535'
                                  2364+             PUNCH 'CNT6        DFSCNT T2740NSC,0000,0,CTB5-DFSICTB,65535'
                                  2365+             PUNCH 'CNT39       DFSCNT T2741,0000,0,CTB25-DFSICTB,65535'
                                  2366+             PUNCH 'CNT56       DFSCNT T2741N1,0002,0,CTB40-DFSICTB,CNT57-DFSIX
                                     +              CNT'
                                  2367+             PUNCH 'CNT57       DFSCNT T2741N2,0002,0,CTB40-DFSICTB,65535'
                                  2368+             PUNCH 'CNT58       DFSCNT T2741N3,0002,0,CTB41-DFSICTB,CNT59-DFSIX
                                     +              CNT'
                                  2369+             PUNCH 'CNT59       DFSCNT T2741N4,0002,0,CTB41-DFSICTB,65535'
                                  2370+             PUNCH 'CNT26       DFSCNT T2770C,0000,2,CTB21-DFSICTB,65535'
                                  2371+             PUNCH 'CNT24       DFSCNT T2770P,0000,0,CTB21-DFSICTB,CNT25-DFSICX
                                     +              NT'
                                  2372+             PUNCH 'CNT25       DFSCNT T2770V,0000,1,CTB21-DFSICTB,CNT26-DFSICX
                                     +              NT'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71  10/02/72
                                 2373+         PUNCH 'CNT9      DFSCNT T3270A,0000,0,CTB7-DFSICTB,65535'
                                 2374+         PUNCH 'CNT10     DFSCNT T3270B,0000,0,CTB8-DFSICTB,65535'
                                 2375+         PUNCH 'CNT11     DFSCNT T3270C,0000,0,CTB9-DFSICTB,65535'
                                 2376+         PUNCH 'CNT14     DFSCNT T3270L1,0000,0,CTB12-DFSICTB,65535'
                                 2377+         PUNCH 'CNT15     DFSCNT T3270L2,C000,0,CTB13-DFSICTB,65535'
                                 2378+         PUNCH 'CNT12     DFSCNT T3270P1,C000,0,CTB10-DFSICTB,65535'
                                 2379+         PUNCH 'CNT13     DFSCNT T3270P2,0000,0,CTB11-DFSICTB,65535'
                                 2380+         PUNCH 'CNT16     DFSCNT T3270P3,C000,C,CTB14-DFSICTB,65535'
                                 2381+         PUNCH 'CNT7      DFSCNT T3275,0000,C,CTB6-DFSICTB,CNT8-DFSICNT'
                                 2382+         PUNCH 'CNT8      DFSCNT T3275P,0000,1,CTB6-DFSICTB,65535'
                                 2383+         PUNCH 'CNT64     DFSCNT WTOR,0000,0,CTB0-DFSICTB,65535'
                                 2384+         PUNCH '          TITLE ''DFSICLL0 - COMMUNICATION TERMINAL MATRX
                                     +                 IX (CTM).'''
                                 2385+         PUNCH '          ENTRY DFSICTM'
                                 2386+         PUNCH 'DFSICTM   DS    0D'
                                 2387+         PUNCH '          DC    B''C0000000'''
                                 2388+         PUNCH '          DC    B''10000000'''
                                 2389+         PUNCH '          DC    B''00000000'''
                                 2390+         PUNCH '          DC    B''00000000'''
                                 2391+         PUNCH '          DC    B''00000000'''
                                 2392+         PUNCH '          DC    B''00000000'''
                                 2393+         PUNCH '          DC    B''00000000'''
                                 2394+         PUNCH '          DC    B''00000001'''
                                 2395+         PUNCH '          END'
                                 2396+         PUNCH '/*'
                                 2397+         PUNCH '//STEP2 EXEC PGM=IEV90,REGION=200K,'
                                 2398+         PUNCH '// PARM=''LOAD,NODECK'''
                                 2399+         PUNCH '//SYSLIB DD DSN=IMS2.GENLIB,DISP=SHR'
                                 2400+         PUNCH '//        DD DSN=SYS1.MACLIB,DISP=SHR'
                                 2401+         PUNCH '//SYSLIN DD DSN=IMS2.BLKS50(DFSICLL1),DISP=OLD'
                                 2402+         PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(BLKSIZE=605),'
                                 2403+         PUNCH '// SPACE=(605,(100,50),RLSE,,ROUND)'
                                 2404+         PUNCH '//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),'
                                 2405+         PUNCH '// SPACE=(CYL,(10,5))'
                                 2406+         PUNCH '//SYSIN DD *'
                                 2407+         PUNCH '          DFSCCBS 6,3'
                                 2408+         PUNCH '          TITLE ''DFSICLL1 - COMMUNICATION TERMINAL TABLX
                                     +                 ES (CTT).'''
                                 2409+         PUNCH 'DFSICTTC CSECT'
                                 2410+         PUNCH 'CTT0      ICTTG OPT1=231,OPT2=0,DD=DFSDNSCO,TSND=TRCONSSX
                                     +                 ,TREC=TRCONSS,  X'
                                 2411+         PUNCH '                BUFSZ=146,INLNG=129,NTRYL=0,LTC=51'
                                 2412+         PUNCH 'CTT1      ICTTG BUFSZ=200,NTRYL=2,IDLNL=19,             X
                                     +                    CONTINUE'
                                 2413+         PUNCH '                INLNG=144,IDLTAB=13,IDLLF=2,            X
                                     +                     CONTINUE'
                                 2414+         PUNCH '                LTC=1,OPT1=0,OPT2=C,OPT3=0,             X
                                     +                 CONTINUE'
                                 2415+         PUNCH '                DD=DFSDNG10,                            X
                                     +                 CONTINUE'
                                 2416+         PUNCH '                TSND=TR27401S,                          X
                                     +                 CONTINUE'
                                 2417+         PUNCH '                TREC=TR27401R'
                                 2418+         PUNCH 'CTT2      ICTTG BUFSZ=257,NTRYL=2,IDLNL=0,              X
                                     +                  CONTINUE'
```

```
LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                F01OCT71  10/02/72
                                  2419+          PUNCH '              INLNG=451,IDLTAB=0,IDLLF=0,              X
                                     +                            CONTINUE'
                                  2420+          PUNCH '              LTC=4,OPT1=96,OPT2=0,OPT3=0,             X
                                     +                          CONTINUE'
                                  2421+          PUNCH '              DD=DFSDN040,                             X
                                     +                                CONTINUE'
                                  2422+          PUNCH '              TSND=TR27401S,                           X
                                     +                                CONTINUE'
                                  2423+          PUNCH '              TREC=TR27401R'
                                  2424+          PUNCH 'CTT3      ICTTG BUFSZ=449,NTRYL=2,IDLNL=0,             X
                                     +                           CONTINUE'
                                  2425+          PUNCH '              INLNG=451,IDLTAB=0,IDLLF=0,              X
                                     +                            CONTINUE'
                                  2426+          PUNCH '              LTC=4,OPT1=96,OPT2=0,OPT3=0,             X
                                     +                          CONTINUE'
                                  2427+          PUNCH '              DD=DFSDN040,                             X
                                     +                                CONTINUE'
                                  2428+          PUNCH '              TSND=TR27401S,                           X
                                     +                                CONTINUE'
                                  2429+          PUNCH '              TREC=TR27401R'
                                  2430+          PUNCH 'CTT4      ICTTG BUFSZ=200,NTRYL=2,IDLNL=19,            X
                                     +                            CONTINUE'
                                  2431+          PUNCH '              INLNG=144,IDLTAB=13,IDLLF=2,             X
                                     +                             CONTINUE'
                                  2432+          PUNCH '              LTC=12,OPT1=0,OPT2=48,OPT3=0,            X
                                     +                           CONTINUE'
                                  2433+          PUNCH '              DD=DFSDN120,                             X
                                     +                                CONTINUE'
                                  2434+          PUNCH '              TSND=TR27401S,                           X
                                     +                                CONTINUE'
                                  2435+          PUNCH '              TREC=TR27401R'
                                  2436+          PUNCH 'CTT5      ICTTG BUFSZ=48,NTRYL=6,IDLNL=0,              X
                                     +                          CONTINUE'
                                  2437+          PUNCH '              INLNG=344,IDLTAB=0,IDLLF=0,              X
                                     +                            CONTINUE'
                                  2438+          PUNCH '              LTC=13,OPT1=117,OPT2=2,OPT3=0,           X
                                     +                            CONTINUE'
                                  2439+          PUNCH '              DD=DFSDN130,                             X
                                     +                                CONTINUE'
                                  2440+          PUNCH '              TSND=TRCONSS,                            X
                                     +                               CONTINUE'
                                  2441+          PUNCH '              TREC=TRCONSS'
                                  2442+          PUNCH 'CTT6      ICTTG BUFSZ=8,NTRYL=0,IDLNL=0,               X
                                     +                         CONTINUE'
                                  2443+          PUNCH '              INLNG=306,IDLTAB=C,IDLLF=0,              X
                                     +                            CONTINUE'
                                  2444+          PUNCH '              LTC=14,OPT1=101,OPT2=2,OPT3=0,           X
                                     +                            CONTINUE'
                                  2445+          PUNCH '              DD=DFSDN140,                             X
                                     +                                CONTINUE'
                                  2446+          PUNCH '              TSND=TRCONSS,                            X
                                     +                               CONTINUE'
                                  2447+          PUNCH '              TREC=TRCONSS'
                                  2448+          PUNCH 'CTT7      ICTTG BUFSZ=8,NTRYL=0,ICLNL=0,               X
                                     +                         CONTINUE'
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT                                                F01OCT71  10/02/72
                                  2449+          PUNCH '             INLNG=6,IDLTAB=0,IDLLF=0,            X
                                     +                        CONTINUE'
                                  2450+          PUNCH '             LTC=14,OPT1=101,OPT2=2,OPT3=0,       X
                                     +                          CONTINUE'
                                  2451+          PUNCH '             DD=DFSDN140,                         X
                                     +                              CONTINUE'
                                  2452+          PUNCH '             TSND=TRCONSS,                        X
                                     +                             CONTINUE'
                                  2453+          PUNCH '             TREC=TRCONSS'
                                  2454+          PUNCH 'CTT8      ICTTG BUFSZ=200,NTRYL=3,IDLNL=19,        X
                                     +                          CONTINUE'
                                  2455+          PUNCH '             INLNG=144,IDLTAB=13,IDLLF=2,         X
                                     +                           CONTINUE'
                                  2456+          PUNCH '             LTC=2,OPT1=0,OPT2=0,OPT3=0,          X
                                     +                       CONTINUE'
                                  2457+          PUNCH '             DD=DFSDN020,                         X
                                     +                              CONTINUE'
                                  2458+          PUNCH '             TSND=IECTSD50,                       X
                                     +                              CONTINUE'
                                  2459+          PUNCH '             TREC=IECTRC50'
                                  2460+          PUNCH 'CTT9      ICTTG BUFSZ=970,NTRYL=3,IDLNL=0,         X
                                     +                         CONTINUE'
                                  2461+          PUNCH '             INLNG=970,IDLTAB=3,IDLLF=0,          X
                                     +                          CONTINUE'
                                  2462+          PUNCH '             LTC=3,OPT1=96,OPT2=128,OPT3=0,       X
                                     +                          CONTINUE'
                                  2463+          PUNCH '             DD=DFSDN030,                         X
                                     +                              CONTINUE'
                                  2464+          PUNCH '             TSND=TR2260RS,                       X
                                     +                              CONTINUE'
                                  2465+          PUNCH '             TREC=TR2260RR'
                                  2466+          PUNCH 'CTT10     ICTTG BUFSZ=970,NTRYL=3,IDLNL=0,         X
                                     +                         CONTINUE'
                                  2467+          PUNCH '             INLNG=970,IDLTAB=0,IDLLF=0,          X
                                     +                          CONTINUE'
                                  2468+          PUNCH '             LTC=3,OPT1=96,OPT2=128,OPT3=0,       X
                                     +                          CONTINUE'
                                  2469+          PUNCH '             DD=DFSDN030,                         X
                                     +                              CONTINUE'
                                  2470+          PUNCH '             TSND=TR2260RS,                       X
                                     +                              CONTINUE'
                                  2471+          PUNCH '             TREC=TR2260RR'
                                  2472+          PUNCH 'CTT11     ICTTG BUFSZ=276,NTRYL=5,IDLNL=0,         X
                                     +                         CONTINUE'
                                  2473+          PUNCH '             INLNG=277,IDLTAB=0,IDLLF=0,          X
                                     +                          CONTINUE'
                                  2474+          PUNCH '             LTC=9,OPT1=181,OPT2=0,OPT3=0,        X
                                     +                         CONTINUE'
                                  2475+          PUNCH '             DD=DFSDN090,                         X
                                     +                              CONTINUE'
                                  2476+          PUNCH '             TSND=TRCONSS,                        X
                                     +                             CONTINUE'
                                  2477+          PUNCH '             TREC=TRCONSS'
                                  2478+          PUNCH 'CTT12     ICTTG BUFSZ=100,NTRYL=5,IDLNL=15,        X
                                     +                         CONTINUE'
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT                                               F01OCT71  10/02/72
                                  2479+          PUNCH '                INLNG=100,IDLTAB=10,IDLLF=15,           X
                                     +                               CONTINUE'
                                  2480+          PUNCH '                LTC=11,OPT1=53,OPT2=0,OPT3=0,           X
                                     +                            CONTINUE'
                                  2481+          PUNCH '                DD=DFSDN110,                            X
                                     +                                  CONTINUE'
                                  2482+          PUNCH '                TSND=MD1EBCLC,                          X
                                     +                                  CONTINUE'
                                  2483+          PUNCH '                TREC=MD1LCEBC'
                                  2484+          PUNCH 'CTT13     ICTTG BUFSZ=100,NTRYL=5,IDLNL=15,              X
                                     +                             CONTINUE'
                                  2485+          PUNCH '                INLNG=100,IDLTAB=10,IDLLF=15,           X
                                     +                               CONTINUE'
                                  2486+          PUNCH '                LTC=11,OPT1=53,OPT2=0,OPT3=0,           X
                                     +                            CONTINUE'
                                  2487+          PUNCH '                DD=DFSDN110,                            X
                                     +                                  CONTINUE'
                                  2488+          PUNCH '                TSND=MD2EBCLC,                          X
                                     +                                  CONTINUE'
                                  2489+          PUNCH '                TREC=MD2LCEBC'
                                  2490+          PUNCH 'CTT14     ICTTG BUFSZ=100,NTRYL=5,IDLNL=15,              X
                                     +                             CONTINUE'
                                  2491+          PUNCH '                INLNG=100,IDLTAB=10,IDLLF=15,           X
                                     +                               CONTINUE'
                                  2492+          PUNCH '                LTC=11,OPT1=53,OPT2=0,OPT3=0,           X
                                     +                            CONTINUE'
                                  2493+          PUNCH '                DD=DFSDN110,                            X
                                     +                                  CONTINUE'
                                  2494+          PUNCH '                TSND=MD4EBCLC,                          X
                                     +                                  CONTINUE'
                                  2495+          PUNCH '                TREC=MD4LCEBC'
                                  2496+          PUNCH 'CTT15     ICTTG BUFSZ=200,NTRYL=0,IDLNL=13,              X
                                     +                             CONTINUE'
                                  2497+          PUNCH '                INLNG=144,IDLTAB=13,IDLLF=2,            X
                                     +                              CONTINUE'
                                  2498+          PUNCH '                LTC=8,OPT1=0,OPT2=56,OPT3=0,            X
                                     +                           CONTINUE'
                                  2499+          PUNCH '                DD=DFSDN080,                            X
                                     +                                  CONTINUE'
                                  2500+          PUNCH '                TSND=IECTSD41,                          X
                                     +                                  CONTINUE'
                                  2501+          PUNCH '                TREC=IECTRF41'
                                  2502+          PUNCH 'CTT16     ICTTG BUFSZ=200,NTRYL=0,IDLNL=13,              X
                                     +                             CONTINUE'
                                  2503+          PUNCH '                INLNG=144,IDLTAB=13,IDLLF=2,            X
                                     +                              CONTINUE'
                                  2504+          PUNCH '                LTC=69,OPT1=13,OPT2=56,OPT3=0,          X
                                     +                             CONTINUE'
                                  2505+          PUNCH '                DD=DFSDS050,                            X
                                     +                                  CONTINUE'
                                  2506+          PUNCH '                TSND=IECTSD41,                          X
                                     +                                  CONTINUE'
                                  2507+          PUNCH '                TREC=IECTRF41'
                                  2508+          PUNCH 'CTT17     ICTTG BUFSZ=200,NTRYL=2,IDLNL=52,              X
                                     +                             CONTINUE'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                       F01OCT71  10/02/72
                                 2509+         PUNCH '             INLNG=200,IDLTAB=52,IDLLF=8,              X
                                    +                          CONTINUE'
                                 2510+         PUNCH '             LTC=6,OPT1=100,OPT2=128,OPT3=0,           X
                                    +                          CONTINUE'
                                 2511+         PUNCH '             DD=DFSDN060,                              X
                                    +                             CONTINUE'
                                 2512+         PUNCH '             TSND=IECTSD30,                            X
                                    +                             CONTINUE'
                                 2513+         PUNCH '             TREC=IECTRC30'
                                 2514+         PUNCH 'CTT18     ICTTG BUFSZ=10,NTRYL=0,IDLNL=0,              X
                                    +                       CONTINUE'
                                 2515+         PUNCH '             INLNG=90,IDLTAB=0,IDLLF=0,                X
                                    +                        CONTINUE'
                                 2516+         PUNCH '             LTC=15,OPT1=103,OPT2=160,OPT3=0,          X
                                    +                           CONTINUE'
                                 2517+         PUNCH '             DD=DFSDN150,                              X
                                    +                             CONTINUE'
                                 2518+         PUNCH '             TSND=0,                                   X
                                    +                     CONTINUE'
                                 2519+         PUNCH '             TREC=0'
                                 2520+         PUNCH 'CTT19     ICTTG BUFSZ=143,NTRYL=0,IDLNL=0,             X
                                    +                        CONTINUE'
                                 2521+         PUNCH '             INLNG=10,IDLTAB=0,IDLLF=0,                X
                                    +                        CONTINUE'
                                 2522+         PUNCH '             LTC=15,OPT1=103,OPT2=32,OPT3=0,           X
                                    +                          CONTINUE'
                                 2523+         PUNCH '             DD=DFSDN150,                              X
                                    +                             CONTINUE'
                                 2524+         PUNCH '             TSND=TRPRTPN,                             X
                                    +                           CONTINUE'
                                 2525+         PUNCH '             TREC=0'
                                 2526+         PUNCH 'CTT20     ICTTG BUFSZ=224,NTRYL=0,IDLNL=0,             X
                                    +                        CONTINUE'
                                 2527+         PUNCH '             INLNG=10,IDLTAB=0,IDLLF=0,                X
                                    +                        CONTINUE'
                                 2528+         PUNCH '             LTC=15,OPT1=103,OPT2=36,OPT3=0,           X
                                    +                          CONTINUE'
                                 2529+         PUNCH '             DD=DFSDN150,                              X
                                    +                             CONTINUE'
                                 2530+         PUNCH '             TSND=0,                                   X
                                    +                     CONTINUE'
                                 2531+         PUNCH '             TREC=0'
                                 2532+         PUNCH 'CTT21     ICTTG BUFSZ=880,NTRYL=0,IDLNL=0,             X
                                    +                        CONTINUE'
                                 2533+         PUNCH '             INLNG=32,IDLTAB=0,IDLLF=0,                X
                                    +                        CONTINUE'
                                 2534+         PUNCH '             LTC=15,OPT1=103,OPT2=32,OPT3=0,           X
                                    +                          CONTINUE'
                                 2535+         PUNCH '             DD=DFSDN150,                              X
                                    +                             CONTINUE'
                                 2536+         PUNCH '             TSND=0,                                   X
                                    +                     CONTINUE'
                                 2537+         PUNCH '             TREC=0'
                                 2538+         PUNCH 'CTT22     ICTTG BUFSZ=256,NTRYL=0,IDLNL=0,             X
                                    +                        CONTINUE'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                         F01OCT71  10/02/72
                                 2539+         PUNCH '            INLNG=256,IDLTAB=0,IDLLF=0,          X
                                     +                        CONTINUE'
                                 2540+         PUNCH '            LTC=67,OPT1=12,OPT2=56,OPT3=0,       X
                                     +                        CONTINUE'
                                 2541+         PUNCH '            DD=DFSDS030,                         X
                                     +                            CONTINUE'
                                 2542+         PUNCH '            TSND=OUT7770,                        X
                                     +                           CONTINUE'
                                 2543+         PUNCH '            TREC=ABB7770'
                                 2544+         PUNCH '       ROUTINE OUT7770 '
                                 2545+         PUNCH 'CTT23   ICTTG BUFSZ=210,NTRYL=0,IDLNL=0,         X
                                     +                       CONTINUE'
                                 2546+         PUNCH '            INLNG=60,IDLTAB=0,IDLLF=0,           X
                                     +                       CONTINUE'
                                 2547+         PUNCH '            LTC=67,OPT1=12,OPT2=56,OPT3=0,       X
                                     +                        CONTINUE'
                                 2548+         PUNCH '            DD=DFSDS030,                         X
                                     +                            CONTINUE'
                                 2549+         PUNCH '            TSND=OUT7770,                        X
                                     +                           CONTINUE'
                                 2550+         PUNCH '            TREC=ABC7770'
                                 2551+         PUNCH '       ROUTINE OUT7770 '
                                 2552+         PUNCH 'CTT24   ICTTG BUFSZ=200,NTRYL=0,IDLNL=2,         X
                                     +                       CONTINUE'
                                 2553+         PUNCH '            INLNG=200,IDLTAB=2,IDLLF=2,          X
                                     +                        CONTINUE'
                                 2554+         PUNCH '            LTC=68,OPT1=40,OPT2=8,OPT3=0,        X
                                     +                       CONTINUE'
                                 2555+         PUNCH '            DD=DFSDS040,                         X
                                     +                            CONTINUE'
                                 2556+         PUNCH '            TSND=IECTSCT2,                       X
                                     +                            CONTINUE'
                                 2557+         PUNCH '            TREC=IECTRCT2'
                                 2558+         PUNCH 'CTT25   ICTTG BUFSZ=200,NTRYL=0,IDLNL=19,        X
                                     +                        CONTINUE'
                                 2559+         PUNCH '            INLNG=144,IDLTAB=13,IDLLF=2,         X
                                     +                         CONTINUE'
                                 2560+         PUNCH '            LTC=65,OPT1=8,OPT2=0,OPT3=0,         X
                                     +                      CONTINUE'
                                 2561+         PUNCH '            DD=DFSDS010,                         X
                                     +                            CONTINUE'
                                 2562+         PUNCH '            TSND=TR27401S,                       X
                                     +                            CONTINUE'
                                 2563+         PUNCH '            TREC=TR27401R'
                                 2564+         PUNCH 'CTT26   ICTTG BUFSZ=200,NTRYL=3,IDLNL=19,        X
                                     +                        CONTINUE'
                                 2565+         PUNCH '            INLNG=144,IDLTAB=13,IDLLF=2,         X
                                     +                         CONTINUE'
                                 2566+         PUNCH '            LTC=66,OPT1=8,OPT2=0,OPT3=0,         X
                                     +                      CONTINUE'
                                 2567+         PUNCH '            DD=DFSDS020,                         X
                                     +                            CONTINUE'
                                 2568+         PUNCH '            TSND=IECTSD50,                       X
                                     +                            CONTINUE'
                                 2569+         PUNCH '            TREC=IECTRC50'
```

```
LOC  OBJECT CODE      ADDR1 ADDR2  STMT    SOURCE STATEMENT                                                    F01OCT71  10/02/72
                                   2570+          PUNCH '         ROUTINE GEN    '
                                   2571+          PUNCH '         TITLE ''DFSICLL1 - COMMUNICATION TRANSLATION TX
                                       +                 ABLES.'''
                                   2572+          PUNCH 'DFSITRTO CSECT'
                                   2573+          PUNCH '         COPY TRCONSS'
                                   2574+          PUNCH '         ENTRY IECTSD41,IECTRF41       '
                                   2575+          PUNCH '         ASMTRTAB  SD41'
                                   2576+          PUNCH '         ORG    IECTSD41+55'
                                   2577+          PUNCH '         DC     X''EB'''
                                   2578+          PUNCH '         ORG'
                                   2579+          PUNCH '         ASMTRTAB  RF41'
                                   2580+          PUNCH '         ORG    IECTRF41+31'
                                   2581+          PUNCH '         DC     X''42'''
                                   2582+          PUNCH '         ORG    IECTRF41+159'
                                   2583+          PUNCH '         DC     X''42'''
                                   2584+          PUNCH '         ORG'
                                   2585+          PUNCH '         EJECT'
                                   2586+          PUNCH '         ENTRY IECTSD40,IECTRF40       '
                                   2587+          PUNCH '         ASMTRTAB  SD40'
                                   2588+          PUNCH '         ORG    IECTSD40+55'
                                   2589+          PUNCH '         DC     X''88'''
                                   2590+          PUNCH '         ORG'
                                   2591+          PUNCH '         ASMTRTAB  RF40'
                                   2592+          PUNCH '         ORG    IECTRF40+31'
                                   2593+          PUNCH '         DC     X''42'''
                                   2594+          PUNCH '         ORG'
                                   2595+          PUNCH '         EJECT'
                                   2596+          PUNCH '         ENTRY IECTSB40,IECTRU40       '
                                   2597+          PUNCH '         ASMTRTAB  SB40'
                                   2598+          PUNCH '         ORG    IECTSB40+55'
                                   2599+          PUNCH '         DC     X''88'''
                                   2600+          PUNCH '         ORG'
                                   2601+          PUNCH '         ASMTRTAB  RU40'
                                   2602+          PUNCH '         ORG    IECTRU40+31'
                                   2603+          PUNCH '         DC     X''42'''
                                   2604+          PUNCH '         ORG'
                                   2605+          PUNCH '         EJECT'
                                   2606+          PUNCH '         ASMTRTAB RCT2,SCT2 '
                                   2607+          PUNCH '         ORG    IECTSCT2+21 '
                                   2608+          PUNCH '         DC     X''B1'''
                                   2609+          PUNCH '         ORG    IECTSCT2+38 '
                                   2610+          PUNCH '         DC     X''C9'''
                                   2611+          PUNCH '         ORG '
                                   2612+          PUNCH '         EJECT'
                                   2613+          PUNCH '         COPY TRPRTPN '
                                   2614+          PUNCH '         COPY TRN2740'
                                   2615+          PUNCH '         COPY TRN1050'
                                   2616+          PUNCH '         COPY TRN2260'
                                   2617+          PUNCH '         ASMTRTAB RC30,SD30'
                                   2618+          PUNCH '         ORG    IECTSD30+3'
                                   2619+          PUNCH '         DC     X''23'''
                                   2620+          PUNCH '         ORG    IECTSD30+55'
                                   2621+          PUNCH '         DC     X''23'''
                                   2622+          PUNCH '         ORG'
                                   2623+          PUNCH '         EJECT'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71  10/02/72
                                 2624+         PUNCH '          COPY TRN2742'
                                 2625+         PUNCH '          COPY  ABC7770'
                                 2626+         PUNCH '          COPY  ABB7770'
                                 2627+         PUNCH '          COPY  TRAN2980'
                                 2628+         PUNCH '          COPY  TRN29801'
                                 2629+         PUNCH '          COPY  TRN29802'
                                 2630+         PUNCH '          COPY  TRN29804'
                                 2631+         PUNCH '          END'
                                 2632+         PUNCH '/*'
                                 2633+         PUNCH '//STEP3 EXEC PGM=IEV90,REGION=200K,'
                                 2634+         PUNCH '// PARM=''LOAD,NODECK'''
                                 2635+         PUNCH '//SYSLIB DD DSN=IMS2.GENLIB,DISP=SHR'
                                 2636+         PUNCH '//        DD DSN=SYS1.MACLIB,DISP=SHR'
                                 2637+         PUNCH '//SYSLIN DD DSN=IMS2.BLKS50(DFSICV80),DISP=OLD'
                                 2638+         PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(BLKSIZE=605),'
                                 2639+         PUNCH '// SPACE=(605,(100,50),RLSE,,ROUND)'
                                 2640+         PUNCH '//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),'
                                 2641+         PUNCH '// SPACE=(CYL,(10,5))'
                                 2642+         PUNCH '//SYSIN DD *'
                                 2643+         PUNCH '           CVBG  /ASSIGN,DFSICL70,CTM=0'
                                 2644+         PUNCH '           CVBG  /BROADCAST,DFSICLIO,FLAGS=98'
                                 2645+         PUNCH '           CVBG  /CANCEL,DFSCLP20,FLAGS=C6'
                                 2646+         PUNCH '           CVBG  /CHANGE,DFSICL60,CTM=0'
                                 2647+         PUNCH '           CVBG  /CHECKPOINT,DFSICL20,CTM=0,FLAGS=80'
                                 2648+         PUNCH '           CVBG  /DBDUMP,DFSICL20,CTM=C,FLAGS=80'
                                 2649+         PUNCH '          CVBG  /DBRECOVERY,DFSICL20,CTM=0,FLAGS=80'
                                 2650+         PUNCH '           CVBG  /DELETE,DFSICL80,CTM=0'
                                 2651+         PUNCH '           CVBG  /DEQUEUE,DFSICLJ0,FLAGS=80,CTM=0'
                                 2652+         PUNCH '           CVBG  /DISPLAY,DFSICLD0,CTM=0,FLAGS=80'
                                 2653+         PUNCH '           CVBG  /END,DFSICL50'
                                 2654+         PUNCH '           CVBG  /ERESTART,DFSICL20,CTM=0,FLAGS=B4'
                                 2655+         PUNCH '           CVBG  /EXCLUSIVE,DFSICL50'
                                 2656+         PUNCH '           CVBG  /EXIT,DFSICLH0'
                                 2657+         PUNCH '           CVBG  /FORMAT,DFSICLK0,FLAGS=80'
                                 2658+         PUNCH '           CVBG  /HOLD,DFSICLH0'
                                 2659+         PUNCH '           CVBG  /IAM,DFSICLA0,FLAGS=80'
                                 2660+         PUNCH '          CVBG  /IDLE,DFSICLG0,CTM=0,FLAGS=80'
                                 2661+         PUNCH '           CVBG  /LOCK,DFSICL90'
                                 2662+         PUNCH '           CVBG  /LOG,DFSCLP10,FLAGS=02'
                                 2663+         PUNCH '           CVBG  /LOOPTEST,DFSICL50,FLAGS=98' 2605
                                 2664+         PUNCH '           CVBG  /NRESTART,DFSICL20,CTM=0,FLAGS=A4'
                                 2665+         PUNCH '           CVBG  /PSTOP,DFSICL40,CTM=0'
                                 2666+         PUNCH '           CVBG  /PURGE,DFSICL40,CTM=0'
                                 2667+         PUNCH '           CVBG  /RDISPLAY,DFSICLD0,FLAGS=80'
                                 2668+         PUNCH '           CVBG  /RELEASE,DFSICLH0'
                                 2669+         PUNCH '           CVBG  /RESET,DFSICLE0'
                                 2670+         PUNCH '          CVBG  /RSTART,DFSICL40,CTM=0'
                                 2671+         PUNCH '           CVBG  /SET,DFSICLE0'
                                 2672+         PUNCH '           CVBG  /START,DFSICL40,CTM=0'
                                 2673+         PUNCH '           CVBG  /STOP,DFSICL40,CTM=0'
                                 2674+         PUNCH '           CVBG  /TEST,DFSICL50,FLAGS=80' 2605
                                 2675+         PUNCH '           CVBG  /TRACE,DFSICLN0,FLAGS=80,CTM=0'
                                 2676+         PUNCH '           CVBG  /UNLOCK,DFSICL90'
                                 2677+         PUNCH '          END'
                                 2678+         PUNCH '/*'
```

```
LOC  OBJECT CODE      ADDR1 ADDR2  STMT    SOURCE STATEMENT                                                F010CT71  10/02/72

                                   2679+          PUNCH '//STEP4 EXEC PGM=IEV90,REGION=200K,'
                                   2680+          PUNCH '// PARM=''LOAD,NODECK'''
                                   2681+          PUNCH '//SYSLIB DD DSN=IMS2.GENLIB,DISP=SHR'
                                   2682+          PUNCH '//          DD DSN=SYS1.MACLIB,DISP=SHR'
                                   2683+          PUNCH '//SYSLIN DD DSN=IMS2.BLKS50(DFSISDBB),DISP=OLD'
                                   2684+          PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(BLKSIZE=605),'
                                   2685+          PUNCH '// SPACE=(605,(100,50),RLSE,,ROUND)'
                                   2686+          PUNCH '//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),'
                                   2687+          PUNCH '// SPACE=(CYL,(10,5))'
                                   2688+          PUNCH '//SYSIN DD *'
                                   2689+          PUNCH 'SDB0       TITLE ''DFSISDBB - SECURITY DIRECTORY BLOCKS (X
                                       +                SDB).'''
                                   2690+          PUNCH 'CFSISCB0 CSECT'
                                   2691+          PUNCH '           ENTRY DFSISDB'
                                   2692+          PUNCH 'DFSISDB    DS     0D'
                                   2693+          PUNCH '           DC     A(SMBL-DFSISDB)      OFFSET TO SMB LIST.'
                                   2694+          PUNCH '           DC     A(CNTL-DFSISCB)      OFFSET TO CNT LIST. '
                                   2695+          PUNCH '           DC     A(DMDL-DFSISDB)      OFFSET TO DMD LIST.'
                                   2696+          PUNCH '           DC     A(PSBL-DFSISDB)      OFFSET TO PSB LIST.'
                                   2697+          PUNCH '           DC     A(CVBL-DFSISDB)      OFFSET TO CVB LIST.'
                                   2698+          PUNCH '           DC     A(CTBL-DFSISDB)      OFFSET TO CTB LIST.'
                                   2699+          PUNCH '           DC     A(MASTER-DFSISDB)  OFFSET TO MASTER CNT.X
                                       +          '
                                   2700+          PUNCH '           DC     A((CTBL-DFSISDB)+((1-1)*5))      CFFSET TOX
                                       +                MASTER CTB.'
                                   2701+          PUNCH 'SMBL       DC     AL2((SMBLE-SMBLL)/L''SMBLL,L''SMBLL)'
                                   2702+          PUNCH 'SMBLL      DS     0CL8'
                                   2703+          PUNCH '           DC     CL8''ADDINV'''
                                   2704+          PUNCH '           DC     CL8''ADDPART'''
                                   2705+          PUNCH '           DC     CL8''CLOSE'''
                                   2706+          PUNCH '           DC     CL8''CONV1'''
                                   2707+          PUNCH '           DC     CL8''CONV2'''
                                   2708+          PUNCH '           CC     CL8''DFS'''
                                   2709+          PUNCH '           DC     CL8''DISBURSE'''
                                   2710+          PUNCH '           DC     CL8''DLETINV'''
                                   2711+          PUNCH '           DC     CL8''DLETPART'''
                                   2712+          PUNCH '           DC     CL8''DLI'''
                                   2713+          PUNCH '           DC     CL8''DLN'''
                                   2714+          PUNCH '           DC     CL8''DSPALLI'''
                                   2715+          PUNCH '           DC     CL8''DSPINV'''
                                   2716+          PUNCH '           DC     CL8''ENQ'''
                                   2717+          PUNCH '           DC     CL8''ICS'''
                                   2718+          PUNCH '           DC     CL8''IMS'''
                                   2719+          PUNCH '           DC     CL8''LKM'''
                                   2720+          PUNCH '           DC     CL8''MR1'''
                                   2721+          PUNCH '           DC     CL8''MR2'''
                                   2722+          PUNCH '           DC     CL8''MR3'''
                                   2723+          PUNCH '           DC     CL8''MR4'''
                                   2724+          PUNCH '           DC     CL8''MR5'''
                                   2725+          PUNCH '           CC     CL8''MR6'''
                                   2726+          PUNCH '           DC     CL8''MR7'''
                                   2727+          PUNCH '           DC     CL8''MR8'''
                                   2728+          PUNCH '           DC     CL8''MR9'''
                                   2729+          PUNCH '           DC     CL8''PART'''
                                   2730+          PUNCH '           DC     CL8''SKD1'''
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71  10/02/72

                                 2731+        PUNCH '          DC     CL8''SKH1'''
                                 2732+        PUNCH '          CC     CL8''SKH2'''
                                 2733+        PUNCH '          DC     CL8''SKH3'''
                                 2734+        PUNCH '          DC     CL8''SKI1'''
                                 2735+        PUNCH '          DC     CL8''SKI2'''
                                 2736+        PUNCH '          DC     CL8''TPPL1'''
                                 2737+        PUNCH '          DC     CL8''TPPL2'''
                                 2738+        PUNCH '          DC     CL8''TUBE'''
                                 2739+        PUNCH 'SMBLE     EQU    *'
                                 2740+        PUNCH 'CNTL      DC     AL2((CNTLE-CNTLL)/L''CNTLL,L''CNTLL)'
                                 2741+        PUNCH 'CNTLL     DS     0CL8'
                                 2742+        PUNCH '          DC     CL8''AA'''
                                 2743+        PUNCH '          DC     CL8''AB'''
                                 2744+        PUNCH '          DC     CL8''ATELLER'''
                                 2745+        PUNCH '          DC     CL8''A29801'''
                                 2746+        PUNCH '          DC     CL8''A29802'''
                                 2747+        PUNCH '          DC     CL8''BTELLER'''
                                 2748+        PUNCH '          DC     CL8''CAROL'''
                                 2749+        PUNCH '          DC     CL8''COMMON'''
                                 2750+        PUNCH 'MASTER    DC     CL8''CTRL'''
                                 2751+        PUNCH '          DC     CL8''ELEANOR'''
                                 2752+        PUNCH '          DC     CL8''ERNE'''
                                 2753+        PUNCH '          DC     CL8''HOMER1'''
                                 2754+        PUNCH '          DC     CL8''HOMER2'''
                                 2755+        PUNCH '          DC     CL8''HOMER4'''
                                 2756+        PUNCH '          DC     CL8''HOWARD'''
                                 2757+        PUNCH '          DC     CL8''INQU0393'''
                                 2758+        PUNCH '          DC     CL8''INQU0401'''
                                 2759+        PUNCH '          DC     CL8''INQU0464'''
                                 2760+        PUNCH '          DC     CL8''INQU0469'''
                                 2761+        PUNCH '          CC     CL8''INQU0477'''
                                 2762+        PUNCH '          DC     CL8''INQU0485'''
                                 2763+        PUNCH '          DC     CL8''INQUC493'''
                                 2764+        PUNCH '          DC     CL8''JTELLER'''
                                 2765+        PUNCH '          DC     CL8''KTELLER'''
                                 2766+        PUNCH '          DC     CL8''LARRY'''
                                 2767+        PUNCH '          DC     CL8''MODEL2'''
                                 2768+        PUNCH '          DC     CL8''MODEL2K'''
                                 2769+        PUNCH '          DC     CL8''N29801'''
                                 2770+        PUNCH '          DC     CL8''N29802'''
                                 2771+        PUNCH '          DC     CL8''PRINTER1'''
                                 2772+        PUNCH '          DC     CL8''PRINTER2'''
                                 2773+        PUNCH '          DC     CL8''PRINTER3'''
                                 2774+        PUNCH '          DC     CL8''PRTSYS'''
                                 2775+        PUNCH '          DC     CL8''SHARRON'''
                                 2776+        PUNCH '          DC     CL8''SP1'''
                                 2777+        PUNCH '          DC     CL8''SP2'''
                                 2778+        PUNCH '          CC     CL8''SWCDPNCH'''
                                 2779+        PUNCH '          DC     CL8''SWPRNTR2'''
                                 2780+        PUNCH '          DC     CL8''SWTPPNCH'''
                                 2781+        PUNCH '          DC     CL8''SW1050'''
                                 2782+        PUNCH '          DC     CL8''TAPESYS'''
                                 2783+        PUNCH '          DC     CL8''T1033A'''
                                 2784+        PUNCH '          DC     CL8''T1033B'''
                                 2785+        PUNCH '          DC     CL8''T2265XCL'''
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                           F01OCT71  10/02/72
                                 2786+        PUNCH '          DC    CL8''T2740NSC'''
                                 2787+        PUNCH '          DC    CL8''T2741'''
                                 2788+        PUNCH '          DC    CL8''T2741N1'''
                                 2789+        PUNCH '          DC    CL8''T2741N2'''
                                 2790+        PUNCH '          DC    CL8''T2741N3'''
                                 2791+        PUNCH '          DC    CL8''T2741N4'''
                                 2792+        PUNCH '          DC    CL8''T2770C'''
                                 2793+        PUNCH '          DC    CL8''T2770P'''
                                 2794+        PUNCH '          DC    CL8''T2770V'''
                                 2795+        PUNCH '          DC    CL8''T3270A'''
                                 2796+        PUNCH '          DC    CL8''T3270B'''
                                 2797+        PUNCH '          DC    CL8''T3270C'''
                                 2798+        PUNCH '          DC    CL8''T3270L1'''
                                 2799+        PUNCH '          DC    CL8''T3270L2'''
                                 2800+        PUNCH '          DC    CL8''T3270P1'''
                                 2801+        PUNCH '          DC    CL8''T3270P2'''
                                 2802+        PUNCH '          DC    CL8''T3270P3'''
                                 2803+        PUNCH '          DC    CL8''T3275'''
                                 2804+        PUNCH '          DC    CL8''T3275P'''
                                 2805+        PUNCH '          DC    CL8''WTOR'''
                                 2806+        PUNCH 'CNTLE     EQU   *'
                                 2807+        PUNCH 'DMDL      DC    AL2((DMDLE-DMDLL)/L''DMDLL,L''DMDLL)'
                                 2808+        PUNCH 'DMDLL     DS    0CL8'
                                 2809+        PUNCH '          DC    CL8''DD41SK01'''
                                 2810+        PUNCH '          DC    CL8''DD41TS01'''
                                 2811+        PUNCH '          DC    CL8''DH41SK01'''
                                 2812+        PUNCH '          DC    CL8''DH41SK02'''
                                 2813+        PUNCH '          DC    CL8''DH41SK03'''
                                 2814+        PUNCH '          DC    CL8''DH41TS01'''
                                 2815+        PUNCH '          DC    CL8''DH41TS02'''
                                 2816+        PUNCH '          DC    CL8''DH41TS03'''
                                 2817+        PUNCH '          DC    CL8''DI21PART'''
                                 2818+        PUNCH '          DC    CL8''DI31LM01'''
                                 2819+        PUNCH '          DC    CL8''DI31PH02'''
                                 2820+        PUNCH '          DC    CL8''DI41SK01'''
                                 2821+        PUNCH '          DC    CL8''DI42SK01'''
                                 2822+        PUNCH '          DC    CL8''DS40JC01'''
                                 2823+        PUNCH '          DC    CL8''DX41SK01'''
                                 2824+        PUNCH '          DC    CL8''DX41TS01'''
                                 2825+        PUNCH 'DMDLE     EQU   *'
                                 2826+        PUNCH 'PSBL      DC    AL2((PSBLE-PSBLL)/L''PSBLL,L''PSBLL)'
                                 2827+        PUNCH 'PSBLL     DS    0CL8'
                                 2828+        PUNCH '          DC    CL8''CONTEST'''
                                 2829+        PUNCH '          DC    CL8''DFSSAM02'''
                                 2830+        PUNCH '          DC    CL8''DFSSAM03'''
                                 2831+        PUNCH '          DC    CL8''DFSSAM04'''
                                 2832+        PUNCH '          DC    CL8''DFSSAM05'''
                                 2833+        PUNCH '          DC    CL8''DFSSAM06'''
                                 2834+        PUNCH '          DC    CL8''DFSSAM07'''
                                 2835+        PUNCH '          DC    CL8''ENQ0SK41'''
                                 2836+        PUNCH '          DC    CL8''HDTASK01'''
                                 2837+        PUNCH '          DC    CL8''HHBLSK41'''
                                 2838+        PUNCH '          DC    CL8''HHBLSK42'''
                                 2839+        PUNCH '          DC    CL8''HHBLSK43'''
                                 2840+        PUNCH '          DC    CL8''HHTASK41'''
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                F01OCT71  10/02/72
                                 2841+           PUNCH '          DC      CL8''HHTASK42'''
                                 2842+           PUNCH '          DC      CL8''HHTASK43'''
                                 2843+           PUNCH '          DC      CL8''HIBASK41'''
                                 2844+           PUNCH '          DC      CL8''HIMAJC01'''
                                 2845+           PUNCH '          DC      CL8''HIMAJC02'''
                                 2846+           PUNCH '          DC      CL8''HIMAJC03'''
                                 2847+           PUNCH '          DC      CL8''HIMALM01'''
                                 2848+           PUNCH '          DC      CL8''HITASK41'''
                                 2849+           PUNCH '          DC      CL8''HITASK42'''
                                 2850+           PUNCH '          DC      CL8''HSBASK41'''
                                 2851+           PUNCH '          DC      CL8''LKMDFS00'''
                                 2852+           PUNCH '          DC      CL8''LKMDFS10'''
                                 2853+           PUNCH '          DC      CL8''MR1'''
                                 2854+           PUNCH '          DC      CL8''MR2'''
                                 2855+           PUNCH '          DC      CL8''MR3'''
                                 2856+           PUNCH '          DC      CL8''MR4'''
                                 2857+           PUNCH '          DC      CL8''MR5'''
                                 2858+           PUNCH '          DC      CL8''MR6'''
                                 2859+           PUNCH '          DC      CL8''MR7'''
                                 2860+           PUNCH '          DC      CL8''MR8'''
                                 2861+           PUNCH '          DC      CL8''MR9'''
                                 2862+           PUNCH 'PSBLE     EQU     *'
                                 2863+           PUNCH 'CVBL      DC      AL2((CVBLE-CVBLL)/L''CVBLL,L''CVBLL)'
                                 2864+           PUNCH 'CVBLL     DS      0CL10'
                                 2865+           PUNCH '          DC      CL10''ASSIGN'''
                                 2866+           PUNCH '          DC      CL10''BROADCAST'''
                                 2867+           PUNCH '          DC      CL10''CANCEL'''
                                 2868+           PUNCH '          DC      CL10''CHANGE'''
                                 2869+           PUNCH '          DC      CL10''CHECKPOINT'''
                                 2870+           PUNCH '          DC      CL10''DBDUMP'''
                                 2871+           PUNCH '          DC      CL10''DBRECOVERY'''
                                 2872+           PUNCH '          DC      CL10''DELETE'''
                                 2873+           PUNCH '          DC      CL10''DEQUEUE'''
                                 2874+           PUNCH '          DC      CL10''DISPLAY'''
                                 2875+           PUNCH '          DC      CL10''END'''
                                 2876+           PUNCH '          DC      CL10''ERESTART'''
                                 2877+           PUNCH '          DC      CL10''EXCLUSIVE'''
                                 2878+           PUNCH '          DC      CL10''EXIT'''
                                 2879+           PUNCH '          DC      CL10''FORMAT'''
                                 2880+           PUNCH '          DC      CL10''HOLD'''
                                 2881+           PUNCH '          DC      CL10''IAM'''
                                 2882+           PUNCH '          DC      CL10''IDLE'''
                                 2883+           PUNCH '          DC      CL10''LOCK'''
                                 2884+           PUNCH '          DC      CL10''LOG'''
                                 2885+           PUNCH '          DC      CL10''LOOPTEST'''
                                 2886+           PUNCH '          DC      CL10''NRESTART'''
                                 2887+           PUNCH '          DC      CL10''PSTOP'''
                                 2888+           PUNCH '          DC      CL10''PURGE'''
                                 2889+           PUNCH '          DC      CL10''RDISPLAY'''
                                 2890+           PUNCH '          DC      CL10''RELEASE'''
                                 2891+           PUNCH '          DC      CL10''RESET'''
                                 2892+           PUNCH '          DC      CL10''RSTART'''
                                 2893+           PUNCH '          DC      CL10''SET'''
                                 2894+           PUNCH '          DC      CL10''START'''
                                 2895+           PUNCH '          DC      CL10''STOP'''
```

```
LOC  OBJECT CODE      ADDR1 ADDR2  STMT    SOURCE STATEMENT                                                    F01OCT71  10/02/72
                                   2896+           PUNCH '            DC      CL10''TEST'''
                                   2897+           PUNCH '            DC      CL10''TRACE'''
                                   2898+           PUNCH '            DC      CL10''UNLOCK'''
                                   2899+           PUNCH 'CVBLE       EQU      *'
                                   2900+           PUNCH '            DS      0F'
                                   2901+           PUNCH 'CTBL        DC      AL2(42+1,5)'
                                   2902+           PUNCH '            DC      C''001E2''        S/360 OPERATOR''S CONSOLE.X
                                       +                 '
                                   2903+           PUNCH '            DC      C''002C6'''
                                   2904+           PUNCH '            DC      C''002E2'''
                                   2905+           PUNCH '            DC      C''00345'''
                                   2906+           PUNCH '            DC      C''00346'''
                                   2907+           PUNCH '            DC      C''004E2'''
                                   2908+           PUNCH '            DC      C''00540'''
                                   2909+           PUNCH '            DC      C''00540'''
                                   2910+           PUNCH '            DC      C''005C1'''
                                   2911+           PUNCH '            DC      C''005C2'''
                                   2912+           PUNCH '            DC      C''005C3'''
                                   2913+           PUNCH '            DC      C''005C4'''
                                   2914+           PUNCH '            DC      C''00610'''
                                   2915+           PUNCH '            DC      C''00612'''
                                   2916+           PUNCH '            DC      C''00711'''
                                   2917+           PUNCH '            DC      C''008E2'''
                                   2918+           PUNCH '            DC      C''008E4'''
                                   2919+           PUNCH '            DC      C''00951'''
                                   2920+           PUNCH '            DC      C''009A0'''
                                   2921+           PUNCH '            DC      C''009A1'''
                                   2922+           PUNCH '            DC      C''009A4'''
                                   2923+           PUNCH '            DC      C''010C1'''
                                   2924+           PUNCH '            DC      C''01140'''
                                   2925+           PUNCH '            DC      C''011F1'''
                                   2926+           PUNCH '            DC      C''011F2'''
                                   2927+           PUNCH '            DC      C''01201'''
                                   2928+           PUNCH '            DC      C''01350'''
                                   2929+           PUNCH '            DC      C''01453'''
                                   2930+           PUNCH '            DC      C''01564'''
                                   2931+           PUNCH '            DC      C''01564'''
                                   2932+           PUNCH '            DC      C''01600'''
                                   2933+           PUNCH '            DC      C''01700'''
                                   2934+           PUNCH '            DC      C''01800'''
                                   2935+           PUNCH '            DC      C''01900'''
                                   2936+           PUNCH '            DC      C''020C1'''
                                   2937+           PUNCH '            DC      C''021C4'''
                                   2938+           PUNCH '            DC      C''022E2'''
                                   2939+           PUNCH '            DC      C''023E2'''
                                   2940+           PUNCH '            DC      C''024E2'''
                                   2941+           PUNCH '            DC      C''02500'''
                                   2942+           PUNCH '            DC      C''02500'''
                                   2943+           PUNCH '            DC      C''02500'''
                                   2944+           PUNCH '            DC      C''02500'''
                                   2945+           PUNCH '            END'
                                   2946+           PUNCH '/*'
                                   2947+           PUNCH '//IMSGEN3 JOB (82C,6443),IMS,MSGLEVEL=1,MSGCLASS=A,CLASX
                                       +                 S=D,PRTY=8'
                                   2948+           PUNCH '//STEP1 EXEC PGM=IEV90,REGION=200K,'
```

```
LOC  OBJECT CODE      ADDR1 ADDR2  STMT    SOURCE STATEMENT                                            F01OCT71  10/02/72
                                   2949+          PUNCH '// PARM=''LOAD,NODECK'''
                                   2950+          PUNCH '//SYSLIB DD DSN=IMS2.GENLIB,DISP=SHR'
                                   2951+          PUNCH '//         DD DSN=SYS1.MACLIB,DISP=SHR'
                                   2952+          PUNCH '//SYSLIN DD DSN=IMS2.BLKS50(DFSISYS0),DISP=OLD'
                                   2953+          PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(BLKSIZE=605),'
                                   2954+          PUNCH '// SPACE=(605,(100,50),RLSE,,ROUND)'
                                   2955+          PUNCH '//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),'
                                   2956+          PUNCH '// SPACE=(CYL,(10,5))'
                                   2957+          PUNCH '//SYSIN DD *'
                                   2958+          PUNCH '          TITLE ''DFSISYS0 - IOB''''S, CHAN. PGM''''S, AX
                                       +                ND MSG QUEUE DCB''''S.'''
                                   2959+          PUNCH 'DFSFXC00 XC00  23,500,2000,TYPE=CSECT'
                                   2960+          PUNCH '          SPACE 3'
                                   2961+          PUNCH '          DFSICIOB NUMIOB=15'
                                   2962+          PUNCH '          MSGDCB QCRBUFN=31,MSGBUFN=13,                   X
                                       +                      CONTINUE'
                                   2963+          PUNCH '                SHMSG=192,LGMSG=576,                     X
                                       +                   CONTINUE'
                                   2964+          PUNCH '                DEVTYPE=(2314,2314,2314),                X
                                       +                          C'
                                   2965+          PUNCH '                MSGBLK=576 '
                                   2966+          PUNCH '          DFSAVARA 17,EVENTS=31,SECTYPE=CSECT'
                                   2967+          PUNCH '          TITLE ''DFSISYS0 - PARTITION SPECIFICATION TABX
                                       +                LES(PST''''S).'''
                                   2968+          PUNCH '          DFSIPST REGIONS=5'
                                   2969+          PUNCH '          TITLE ''DFSISYS0 - INTERNAL/EXTERNAL QUEUE BLOX
                                       +                CKS.'''
                                   2970+          PUNCH '          DFSQUEUE TASK=5,LINES=25,CLASS=5'
                                   2971+          PUNCH '          END'
                                   2972+          PUNCH '/*'
                                   2973+          PUNCH '//STEP2 EXEC PGM=IEV90,REGION=200K,'
                                   2974+          PUNCH '// PARM=''LOAD,NODECK'''
                                   2975+          PUNCH '//SYSLIB DD DSN=IMS2.GENLIB,DISP=SHR'
                                   2976+          PUNCH '//         DD DSN=SYS1.MACLIB,DISP=SHR'
                                   2977+          PUNCH '//SYSLIN DD DSN=IMS2.BLKS50(DFSVC000),DISP=OLD'
                                   2978+          PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(BLKSIZE=605),'
                                   2979+          PUNCH '// SPACE=(605,(100,50),RLSE,,ROUND)'
                                   2980+          PUNCH '//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),'
                                   2981+          PUNCH '// SPACE=(CYL,(10,5))'
                                   2982+          PUNCH '//SYSIN DD *'
                                   2983+          PUNCH 'DFSVC000 CSECT'
                                   2984+          PUNCH 'DFSVCS   SSCD  SECTYPE=CSECT,SVC=(234,248,,213),DESC=2,X
                                       +                    CONTINUE'
                                   2985+          PUNCH '               OSPS=10,TCT=5,                            X
                                       +                CONTINUE'
                                   2986+          PUNCH '                ROUTCDE=13,OCEP=Z5'
                                   2987+          PUNCH '          END'
                                   2988+          PUNCH '/*'
                                   2989+          PUNCH '//STEP3 EXEC PGM=IEV90,REGION=200K,'
                                   2990+          PUNCH '// PARM=''LOAD,NODECK'''
                                   2991+          PUNCH '//SYSLIB DD DSN=IMS2.GENLIB,DISP=SHR'
                                   2992+          PUNCH '//         DD DSN=SYS1.MACLIB,DISP=SHR'
                                   2993+          PUNCH '//SYSLIN DD DSN=IMS2.BLKS50(DFSISC00),DISP=OLD'
                                   2994+          PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(BLKSIZE=605),'
                                   2995+          PUNCH '// SPACE=(605,(100,50),RLSE,,ROUND)'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  F01OCT71  10/02/72
                                2996+         PUNCH '//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),'
                                2997+         PUNCH '// SPACE=(CYL,(10,5))'
                                2998+         PUNCH '//SYSIN DD *'
                                2999+         PUNCH '        TITLE ''DFSISCDO - SYSTEM CONTENTS DIRECTCRY (X
                                    +               SCD).'''
                                3000+         PUNCH '        CFSGLBS PSB=34,DMB=16,SMB=36,TCT=5,               X
                                    +                  CONTINUES' 7982
                                3001+         PUNCH '               CXB=1,CIB=9,CRB=0,                         X
                                    +                 CONTINUE'
                                3002+         PUNCH '               CLB=25,CTB=43,CNT=64,CDB=24,CTT=26,        X
                                    +                     CONTINUE'
                                3003+         PUNCH '               PST=5,SAV=17,WAT=31,RQE=14,QUE=(1,23),     X
                                    +                     CONTINUE'
                                3004+         PUNCH '               SVC=(234,248),OSAM=(213,251),CVB=34,CTM=1X
                                    +               ,CTML=8,   CONTINUE'
                                3005+         PUNCH '               FBP=13000,FRE=13,                          X
                                    +                      CONTINUE'
                                3006+         PUNCH '               COB=27,CCB=(9,0,192,150,100),EDIT=NO'
                                3007+         PUNCH '        ISCD   SECTYPE=CSECT,CPOPT=2500,PSYS=10,          X
                                    +                   CONTINUE'
                                3008+         PUNCH '               SINIT=(INIT,CLASSA,A),                     X
                                    +                         X'
                                3009+         PUNCH '               MCS=13,DESC=2'
                                3010+         PUNCH '        EJECT'
                                3011+         PUNCH '        DFSINT UCRS=13,MSGS=576,                          X
                                    +                  CONTINUE'
                                3012+         PUNCH '               PSBNO=8000,                                X
                                    +                   CCNTINUE'
                                3013+         PUNCH '               DMBNO=10000,                               X
                                    +                    CONTINUE'
                                3014+         PUNCH '               DBASE=11000,                               X
                                    +                    CONTINUE'
                                3015+         PUNCH '               FBPNO=13000,                               X
                                    +                    CCNTINUE'
                                3016+         PUNCH '               CCMM=10600,POOL=5000'
                                3017+         PUNCH '        END'
                                3018+         PUNCH '/*'
                                3019+         PUNCH '//IMSGEN4 JOB (82C,6443),IMS,MSGLEVEL=1,MSGCLASS=A,CLASX
                                    +               S=D,PRTY=8'
                                3020+         PUNCH '//STEP1 EXEC PGM=IEWL,'
                                3021+         PUNCH '// REGION=130K,'
                                3022+         PUNCH '// PARM=''RENT,REFR,NCAL,LET,XREF,LIST'''
                                3023+         PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZEX
                                    +               =605),'
                                3024+         PUNCH '// SPACE=(605,(10,10),RLSE,,RCUND)'
                                3025+         PUNCH '//SYSOBJ DD DSN=IMS2.BLKS50,DISP=SHR'
                                3026+         PUNCH '//LOAD CD DSN=IMS2.LOAD,DISP=SHR'
                                3027+         PUNCH '//USERLIB DD DSN=ICS.CLOD,DISP=SHR'
                                3028+         PUNCH '//SYSLMCC DD DSN=IMS2.RESLIB,DISP=OLD'
                                3029+         PUNCH '//SYSUT1 DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSOBJ)),SPACE=(CYX
                                    +               L,(10,1))'
                                3030+         PUNCH '//SYSLIN DD *'
                                3031+         PUNCH ' CHANGE DFSVC400(IGC0021C)'
                                3032+         PUNCH ' INCLUDE LOAD(DFSVC400)'
                                3033+         PUNCH ' NAME IGC0021C(R)            TYPE 4 SVC(LOAD 0)'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                      F01OCT71  10/02/72
                                 3034+          PUNCH '  CHANGE DFSVC410(IGC0121C)'
                                 3035+          PUNCH '  INCLUDE LOAD(DFSVC410)'
                                 3036+          PUNCH '  NAME IGC0121C(R)            TYPE 4 SVC(LOAD 1)'
                                 3037+          PUNCH '  CHANGE DFSVC420(IGC0221C)'
                                 3038+          PUNCH '  INCLUDE LOAD(DFSVC420)'
                                 3039+          PUNCH '  NAME IGC0221C(R)            TYPE 4 SVC(LOAD 2)'
                                 3040+          PUNCH '  CHANGE DFSVC430(IGC0321C)'
                                 3041+          PUNCH '  INCLUDE LOAD(DFSVC430)'
                                 3042+          PUNCH '  NAME IGC0321C(R)            TYPE 4 SVC(LOAD 3)'
                                 3043+          PUNCH '  CHANGE DFSAOSA0(IGC0421C)'
                                 3044+          PUNCH '  INCLUDE LOAD(DFSAOSA0)'
                                 3045+          PUNCH '  NAME IGC0421C(R)            TYPE 4 SVC(LOAD 4)'
                                 3046+          PUNCH '  CHANGE DFSAOSB0(IGC0521C)'
                                 3047+          PUNCH '  INCLUDE LOAD(DFSAOSB0)'
                                 3048+          PUNCH '  NAME IGC0521C(R)            TYPE 4 SVC(LOAD 5)'
                                 3049+          PUNCH '  CHANGE DFSAOSC0(IGC0621C)'
                                 3050+          PUNCH '  INCLUDE LOAD(DFSAOSC0)'
                                 3051+          PUNCH '  NAME IGC0621C(R)            TYPE 4 SVC(LOAD 6)'
                                 3052+          PUNCH '  CHANGE DFSAOSD0(IGC0721C)'
                                 3053+          PUNCH '  INCLUDE LOAD(DFSAOSD0)'
                                 3054+          PUNCH '  NAME IGC0721C(R)            TYPE 4 SVC(LOAD 7)'
                                 3055+          PUNCH '  CHANGE DFSICSC0(IGC1021C)'
                                 3056+          PUNCH '  INCLUDE LOAD(DFSICSC0)'
                                 3057+          PUNCH '  NAME IGC1021C(R)            TYPE 4 SVC(LOAD 10)'
                                 3058+          PUNCH '  CHANGE DFSVCIG0(IGC1121C)'
                                 3059+          PUNCH '  INCLUDE LOAD(DFSVCI00)'
                                 3060+          PUNCH '  NAME IGC1121C(R)            TYPE 4 SVC(LOAD 11)'
                                 3061+          PUNCH '  CHANGE DFSASV10(IGC1221C)'
                                 3062+          PUNCH '  INCLUDE LOAD(DFSASV10)'
                                 3063+          PUNCH '  NAME IGC1221C(R)            TYPE 4 SVC(LOAD 12)'
                                 3064+          PUNCH '  CHANGE DFSVC440(IGC1321C)'
                                 3065+          PUNCH '  INCLUDE LOAD(DFSVC440)'
                                 3066+          PUNCH '  NAME IGC1321C(R)            TYPE 4 SVC(LOAD 13)'
                                 3067+          PUNCH '  CHANGE DFSVC100(IGC234)'
                                 3068+          PUNCH '  INCLUDE LOAD(DFSVC100)'
                                 3069+          PUNCH '  NAME IGC234(R)               TYPE 1 SVC INTERFACE'
                                 3070+          PUNCH '  CHANGE DFSVC200(IGC248)'
                                 3071+          PUNCH '  INCLUDE LOAD(DFSVC200)'
                                 3072+          PUNCH '  NAME IGC248(R)               TYPE 2 SVC INTERFACE'
                                 3073+          PUNCH '  CHANGE DFSAOCE0(IGG019Z5)'
                                 3074+          PUNCH '  INCLUDE LOAD(DFSAOCE0)'
                                 3075+          PUNCH '  NAME IGG019Z5(R)          OSAM CHAN. END APPENDAGE'
                                 3076+          PUNCH '  CHANGE DFSAAP10(IGG019Z6)'
                                 3077+          PUNCH '  INCLUDE LOAD(DFSAAP10)'
                                 3078+          PUNCH '  NAME IGG019Z6(R)          7770 CHANNEL/ABNORMAL END APPX
                                     +                ENDAGE'
                                 3079+          PUNCH '  INCLUDE LOAD(DFSARW00)'
                                 3080+          PUNCH '  NAME DFSARW00(R)             7770 READ/WRITE MODULE'
                                 3081+          PUNCH '  INCLUDE LOAD(DFSAOS10)'
                                 3082+          PUNCH '  NAME DFSAOS10(R)             OSAM OPEN INTERFACE'
                                 3083+          PUNCH '  INCLUDE LOAD(DFSAOS20)'
                                 3084+          PUNCH '  NAME DFSAOS20(R)             OSAM READ/WRITE'
                                 3085+          PUNCH '  INCLUDE LOAD(DFSAOS30)'
                                 3086+          PUNCH '  NAME DFSAOS30(R)             OSAM CHECK'
                                 3087+          PUNCH '  INCLUDE LOAD(DFSAOS50)'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT    SOURCE STATEMENT                                              F01OCT71  10/02/72
                                3088+            PUNCH '  NAME DFSAOS50(R)              OSAM COMMON SUBROUTINES'
                                3089+            PUNCH '  INCLUDE LOAD(DFSDISM0)'
                                3090+            PUNCH '  NAME DFSDISM0(R)              DL/I ISAM SIMULATOR'
                                3091+            PUNCH '  INCLUDE LOAD(DFSDLA00)'
                                3092+            PUNCH '  INCLUDE LOAD(DFSDLA30)'
                                3093+            PUNCH '  ENTRY DFSSTART'
                                3094+            PUNCH '  NAME DFSDLA00(R)              DL/I ANALYZER'
                                3095+            PUNCH '  INCLUDE LOAD(DFSDLD00)'
                                3096+            PUNCH '  NAME DFSDLD00(R)              DL/I DELETE/REPLACE'
                                3097+            PUNCH '  INCLUDE LOAD(DFSDLOC0)'
                                3098+            PUNCH '  NAME DFSDLOC0(R)              DL/I OPEN/CLOSE'
                                3099+            PUNCH '  INCLUDE LOAD(DFSDLR00)'
                                3100+            PUNCH '  NAME DFSDLR00(R)              DL/I RETRIEVE'
                                3101+            PUNCH '  INCLUDE LOAD(DFSDDLE0)'
                                3102+            PUNCH '  ALIAS DFSDDLI0'
                                3103+            PUNCH '  NAME DFSDDLE0(R)              DL/I INSERT'
                                3104+            PUNCH '  INCLUDE LOAD(DFSDXMT0)'
                                3105+            PUNCH '  NAME DFSDXMT0(R)              DL/I INDEX MAINTENANCE'
                                3106+            PUNCH '  INCLUDE LOAD(DFSDDBH0)'
                                3107+            PUNCH '  INCLUDE LOAD(DFSDCSP0)'
                                3108+            PUNCH '  INCLUDE LOAD(DFSGNSR0)'
                                3109+            PUNCH '  INCLUDE LOAD(DFSFNDS0)'
                                3110+            PUNCH '  INCLUDE LOAD(DFSRCH00)'
                                3111+            PUNCH '  INCLUDE LOAD(DFSBFWR0)'
                                3112+            PUNCH '  INCLUDE LOAD(DFSDCVT0)'
                                3113+            PUNCH '  INCLUDE LOAD(DFSDBHR0)'
                                3114+            PUNCH '  NAME DFSDBH00(R)              DL/I BUFFER HANDLER'
                                3115+            PUNCH '  INCLUDE LOAD(DFSGGSP0) '
                                3116+            PUNCH '  INCLUDE LOAD(DFSLLCL0) '
                                3117+            PUNCH '  INCLUDE LOAD(DFSMMLC0) '
                                3118+            PUNCH '  INCLUDE LOAD(DFSMMUD0) '
                                3119+            PUNCH '  INCLUDE LOAD(DFSRRHM0) '
                                3120+            PUNCH '  INCLUDE LOAD(DFSRRHP0) '
                                3121+            PUNCH '  INCLUDE LOAD(DFSDHDG0)'
                                3122+            PUNCH '  INCLUDE LOAD(DFSRCHB0)'
                                3123+            PUNCH '  INCLUDE LOAD(DFSFRSP0)'
                                3124+            PUNCH '  ENTRY DFSDHDS0 '
                                3125+            PUNCH '  NAME DFSDHDS0(R)              DL/I SPACE MANAGEMENT'
                                3126+            PUNCH '  INCLUDE LOAD(DFSRRC00)'
                                3127+            PUNCH '  NAME DFSRRC00(R)              REGION CONTROLLOR'
                                3128+            PUNCH '  INCLUDE LOAD(DFSPCC30)'
                                3129+            PUNCH '  ENTRY PCSTART'
                                3130+            PUNCH '  NAME DFSPCC30(R)              BATCH PROGRAM CONTROLLER'
                                3131+            PUNCH '  INCLUDE LOAD(DFSPR000)'
                                3132+            PUNCH '  ALIAS DFSIPRXX'
                                3133+            PUNCH '  NAME DFSPR000(R)              PROG. REQUEST HANDLER'
                                3134+            PUNCH '  INCLUDE LOAD(DFSRRA00)'
                                3135+            PUNCH '  NAME DFSRRA00(R)              REGION PARM ANALYZER'
                                3136+            PUNCH '  INCLUDE LOAD(DFSRRA10)'
                                3137+            PUNCH '  NAME DFSRRA10(R)'
                                3138+            PUNCH '  INCLUDE LOAD(DFSRRA20)'
                                3139+            PUNCH '  NAME DFSRRA20(R)'
                                3140+            PUNCH '  INCLUDE LOAD(DFSRRA50)'
                                3141+            PUNCH '  NAME DFSRRA50(R)'
                                3142+            PUNCH '  INCLUDE LOAD(DFSRDBL0)'
```

```
LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71  10/02/72
                                   3143+          PUNCH '  NAME DFSRDBLO(R)             DATABASE LCGGER'
                                   3144+          PUNCH '  INCLUDE LOAD(DFSILNKO)'
                                   3145+          PUNCH '  NAME DFSILNKO(R)             LINKAGE EDITOR INTERFACE'
                                   3146+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                   3147+          PUNCH '  NAME DFSVC000(R)             SECCNDARY SCD MODULE'
                                   3148+          PUNCH '  INCLUDE LOAD(DFSRRC10)'
                                   3149+          PUNCH '  ENTRY RCSTART'
                                   3150+          PUNCH '  NAME CFSRRC10(R)'
                                   3151+          PUNCH '  INCLUDE LOAD(CFSPCC10)'
                                   3152+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                   3153+          PUNCH '  ENTRY PCSTART'
                                   3154+          PUNCH '  NAME CFSPCC10(R)'
                                   3155+          PUNCH '  INCLUDE LOAD(DFSFPTCO)'
                                   3156+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                   3157+          PUNCH '  NAME DFSFPTCO(R)             STOP REGION'
                                   3158+          PUNCH '/*'
                                   3159+          PUNCH '//STEP2 EXEC PGM=IEWL,'
                                   3160+          PUNCH '// REGICN=130K,'
                                   3161+          PUNCH '// PARM=''NCAL,LET,REUS,XREF,LIST'''
                                   3162+          PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZEX
                                       +                =605),'
                                   3163+          PUNCH '// SPACE=(605,(10,10),RLSE,,RCUND)'
                                   3164+          PUNCH '//SYSOBJ DD DSN=IMS2.BLKS50,DISP=SHR'
                                   3165+          PUNCH '//LOAD DC DSN=IMS2.LOAD,DISP=SHR'
                                   3166+          PUNCH '//USERLIB DD DSN=ICS.CLOD,DISP=SHR'
                                   3167+          PUNCH '//SYSLMOD DD DSN=IMS2.RESLIB,DISP=CLD'
                                   3168+          PUNCH '//SYSUT1 DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSOBJ)),SPACE=(CYX
                                       +                L,(10,1))'
                                   3169+          PUNCH '//SYSLIN DD *'
                                   3170+          PUNCH '  INCLUDE SYSOBJ(DFSBBLKB)          BATCH CONTROL BLOCKS.X
                                       +                '
                                   3171+          PUNCH '  INCLUDE LOAD(DFSBINTO)' BATCH INTERFACE SIMULATOR'
                                   3172+          PUNCH '  INCLUDE LOAD(DFSBLOIO)          BATCH LOG RECORDER'
                                   3173+          PUNCH '  INCLUDE LOAD(DFSFLOGC)           LOG WRITER'
                                   3174+          PUNCH '  ENTRY DFSBNUC'
                                   3175+          PUNCH '  NAME CFSBNUCO(R)             BATCH NUCLEUS'
                                   3176+          PUNCH '  INCLUCE LCAD(DFSISTOO)'
                                   3177+          PUNCH '  INCLUDE LOAD(DFSISTUO)'
                                   3178+          PUNCH '  ENTRY CFSISTOO'
                                   3179+          PUNCH '  NAME DFSISTOO(R)             STATISTICS O'
                                   3180+          PUNCH '  INCLUDE SYSOBJ(DFSIDLIO)     DL/I ON-LINE CONTROL BLOCKX
                                       +                S.'
                                   3181+          PUNCH '  INCLUDE SYSOBJ(DFSICLLO)     COMM. CONTROL BLOCKS(1).'
                                   3182+          PUNCH '  INCLUDE SYSOBJ(DFSICLL1)     COMM. CONTROL BLOCKS(2).'
                                   3183+          PUNCH '  INCLUDE SYSOBJ(DFSICVBO)     COMM. VERB BLOCKS.'
                                   3184+          PUNCH '  INCLUDE SYSCBJ(DFSISYSO)     SYSTEM CONTROL BLOCKS.'
                                   3185+          PUNCH '  INCLUDE SYSOBJ(DFSISCDO)     SYSTEM CONTENTS DIRECTORY.X
                                       +                '
                                   3186+          PUNCH '  INCLUDE USERLIB(DFS10300)  USER SUPPLIED RCUTINE.'
                                   3187+          PUNCH '  INCLUDE USERLIB(DFSS7770)  USER SUPPLIED MODULE'
                                   3188+          PUNCH '  INCLUDE USERLIB(DFSI7770)  USER SUPPLIED MODULE'
                                   3189+          PUNCH '  INCLUDE USERLIB(DFS07770)  USER SUPPLIED MODULE'
                                   3190+          PUNCH '  INCLUCE USERLIB(DFS29800) USER SUPPLIED MODULE'
                                   3191+          PUNCH '  INCLUDE USERLIB(OUT7770)  USER SUPPLIED MODULE.'
                                   3192+          PUNCH '  NAME CFSIBLKB(R)            COMPOSITE SYSTEM CONTROL BLX
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT    SOURCE STATEMENT                                              F01OCT71  10/02/72

                                    +            OCKS MODULE.'
                                3193+         PUNCH '  INCLUDE SYSOBJ(DFSISDBB)'
                                3194+         PUNCH '  NAME DFSISDBB(R)          SECURITY DIRECTORY BLOCKS MX
                                    +            ODULE.'
                                3195+         PUNCH '  INCLUDE LOAD(DFSLI000)'
                                3196+         PUNCH '  ALIAS PLITDLI'
                                3197+         PUNCH '  ALIAS DFSPLI'
                                3198+         PUNCH '  ALIAS CBLTDLI'
                                3199+         PUNCH '  ALIAS DFSCOBOL'
                                3200+         PUNCH '  ALIAS ASMTDLI'
                                3201+         PUNCH '  ALIAS DFSASM'
                                3202+         PUNCH '  ALIAS FORTDLI'
                                3203+         PUNCH '  ALIAS DFSFOR'
                                3204+         PUNCH '  NAME DFSLI000(R)          DL/I LANGUAGE INTERFACE'
                                3205+         PUNCH '  INCLUDE LOAD(DFSDLBL0)'
                                3206+         PUNCH '  INCLUDE LOAD(DFSLI000)'
                                3207+         PUNCH '  NAME DFSDLBL0(R)          DL/I BLOCK BUILDER'
                                3208+         PUNCH '  INCLUDE LOAD(DFSFSTC0)'
                                3209+         PUNCH '  NAME DFSFSTC0(R)          START REGION'
                                3210+         PUNCH '  INCLUDE LOAD(DFSLI000)'
                                3211+         PUNCH '  INCLUDE LOAD(DFSUTS00)'
                                3212+         PUNCH '  ENTRY DLITCBL'
                                3213+         PUNCH '  NAME DFSUTS00(R)'
                                3214+         PUNCH '  INCLUDE LOAD(DFSUTL00)'
                                3215+         PUNCH '  NAME DFSUTL00(R)      FORMAT UTILITY INFO TABLE '
                                3216+         PUNCH '  INCLUDE LOAD(DFSUTL10)'
                                3217+         PUNCH '  INCLUDE LOAD(DFSUTLX0)'
                                3218+         PUNCH '  INCLUDE LOAD(DFSUTLZ0)'
                                3219+         PUNCH '  INCLUDE LOAD(DFSINDX0)'
                                3220+         PUNCH '  ENTRY DFSUTL10'
                                3221+         PUNCH '  NAME DFSUNU10(R)      FORMAT UTILITY PHASE 1 PROC. '
                                3222+         PUNCH '  INCLUDE LOAD(DFSUTL20)'
                                3223+         PUNCH '  NAME DFSUTL20(R)       FORMAT UTILITY PHASE 2 PROC. '
                                3224+         PUNCH '  INCLUDE LOAD(DFSLI000)'
                                3225+         PUNCH '  INCLUDE LOAD(DFSUTL30)'
                                3226+         PUNCH '  INCLUDE LOAD(DFSUTL50)'
                                3227+         PUNCH '  INCLUDE LOAD(DFSUTL60)'
                                3228+         PUNCH '  INCLUDE LOAD(DFSUTL70)'
                                3229+         PUNCH '  INCLUDE LOAD(DFSUTL80)'
                                3230+         PUNCH '  INCLUDE LOAD(DFSUTL90)'
                                3231+         PUNCH '  INCLUDE LOAD(DFSUTLA0)'
                                3232+         PUNCH '  INCLUDE LOAD(DFSUTLB0)'
                                3233+         PUNCH '  INCLUDE LOAD(DFSUTLC0)'
                                3234+         PUNCH '  INCLUDE LOAD(DFSUTLD0)'
                                3235+         PUNCH '  INCLUDE LOAD(DFSUTLE0)'
                                3236+         PUNCH '  INCLUDE LOAD(DFSUTLF0)'
                                3237+         PUNCH '  INCLUDE LOAD(DFSUTLG0)'
                                3238+         PUNCH '  INCLUDE LOAD(DFSUTLH0)'
                                3239+         PUNCH '  INCLUDE LOAD(DFSUTLT0)'
                                3240+         PUNCH '  INCLUDE LOAD(DFSUTLU0)'
                                3241+         PUNCH '  INCLUDE LOAD(DFSUTLV0)'
                                3242+         PUNCH '  INCLUDE LOAD(DFSUTLW0)'
                                3243+         PUNCH '  INCLUDE LOAD(DFSUTLX0)'
                                3244+         PUNCH '  INCLUDE LOAD(DFSUTLY0)'
                                3245+         PUNCH '  INCLUDE LOAD(DFSUTLZ0)'
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT                                              F01OCT71  10/02/72
                                  3246+          PUNCH '   INCLUDE LOAD(DFSUT010)'
                                  3247+          PUNCH '   INCLUDE LOAD(DFSUT020)'
                                  3248+          PUNCH '   INCLUDE LOAD(DFSUT030)'
                                  3249+          PUNCH '   INCLUDE LOAD(DFSUT040)'
                                  3250+          PUNCH '   INCLUDE LOAD(DFSUT050)'
                                  3251+          PUNCH '   INCLUDE LOAD(DFSUT060)'
                                  3252+          PUNCH '   INCLUDE LOAD(DFSUT070)'
                                  3253+          PUNCH '   INCLUDE LOAD(DFSUT080)'
                                  3254+          PUNCH '   INCLUDE LOAD(DFSUT090)'
                                  3255+          PUNCH '   INCLUDE LOAD(DFSUT0A0)'
                                  3256+          PUNCH '   INCLUDE LOAD(DFSINDX0)'
                                  3257+          PUNCH '   INCLUDE LOAD(DFSICATC)'
                                  3258+          PUNCH '   INCLUDE LOAD(DFSISUB0)'
                                  3259+          PUNCH '   ENTRY DLITCBL'
                                  3260+          PUNCH '   NAME DFSUNU20(R)          FORMAT UTILITY PHASE 3 PROC. '
                                  3261+          PUNCH '   INCLUDE LOAD(DFSUTL40)'
                                  3262+          PUNCH '   NAME DFSUNU30(R)          FORMAT UTILITY PHASE 4 PROC. '
                                  3263+          PUNCH '   INCLUDE LOAD(DFSPRRC0)'
                                  3264+          PUNCH '   NAME DFSPRRC0(R)'
                                  3265+          PUNCH '   INCLUDE LOAD(DFSPRPX0)'
                                  3266+          PUNCH '   NAME DFSPRPX0(R)'
                                  3267+          PUNCH '   INCLUDE LOAD(DFSPRPL0)'
                                  3268+          PUNCH '   NAME DFSPRPL0(R)'
                                  3269+          PUNCH '   INCLUDE LOAD(DFSPRRG0)'
                                  3270+          PUNCH '   NAME DFSPRRG0(R)'
                                  3271+          PUNCH '   INCLUDE LOAD(DFSFLOS0)'
                                  3272+          PUNCH '   NAME DFSFLOS0(R)               STAE EXIT ROUTINE'
                                  3273+          PUNCH '   INCLUDE LOAD(DFSDSEH0)'
                                  3274+          PUNCH '   NAME DFSDSEH0(R)            PREFIX RESOLUTION ROUTINE'
                                  3275+          PUNCH '/*'
                                  3276+          PUNCH '//IMSGEN5 JOB (82C,6443),IMS,MSGLEVEL=1,MSGCLASS=A,CLASX
                                      +                S=D,PRTY=8'
                                  3277+          PUNCH '//STEP1 EXEC PGM=IEWL,'
                                  3278+          PUNCH '// REGION=130K,'
                                  3279+          PUNCH '// PARM=''OVLY,NCAL,LET,XREF,LIST'''
                                  3280+          PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZEX
                                      +                =605),'
                                  3281+          PUNCH '// SPACE=(605,(10,10),RLSE,,ROUND)'
                                  3282+          PUNCH '//SYSOBJ DD DSN=IMS2.BLKS50,DISP=SHR'
                                  3283+          PUNCH '//LOAD DD DSN=IMS2.LOAD,DISP=SHR'
                                  3284+          PUNCH '//USERLIB DD DSN=ICS.CLOD,DISP=SHR'
                                  3285+          PUNCH '//TELLIB DD DSN=SYS1.TELCMLIB,DISP=SHR'
                                  3286+          PUNCH '//SYSLMOD DD DSN=IMS2.RESLIB,DISP=OLD'
                                  3287+          PUNCH '//SYSUT1 DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSOBJ)),SPACE=(CYX
                                      +                L,(10,1))'
                                  3288+          PUNCH '//SYSLIN DD *'
                                  3289+          PUNCH '   INCLUDE LOAD(DFSFSWA0)' 2589
                                  3290+          PUNCH '   NAME DFSFSWA0(R)               STAE WKAREA MODULE' 2589
                                  3291+          PUNCH '   INCLUDE LOAD(DFSDLN00)'
                                  3292+          PUNCH '   INCLUDE LOAD(DFSIINL0)'
                                  3293+          PUNCH '   INCLUDE LOAD(DFSIIN10)'
                                  3294+          PUNCH '   INCLUDE LOAD(DFSIIN20)'
                                  3295+          PUNCH '   INCLUDE LOAD(DFSDBPI0)'
                                  3296+          PUNCH '   NAME DFSDLBN0(R)               DL/I INITIALIZATION'
                                  3297+          PUNCH '   INCLUDE LOAD(DFSDPSB0)'
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                             F01OCT71 10/02/72
                                  3298+          PUNCH '  NAME DFSDPSB0(R)           DL/I UTILITY INTERFACE,  PX
                                     +                 SB GENERATOR'
                                  3299+          PUNCH '  INCLUDE LOAD(DFSDAPL0)'
                                  3300+          PUNCH '  NAME DFSDAPL0(R)           DL/I UTILITY INTERFACE,  AX
                                     +                 UTH PGM LIST'
                                  3301+          PUNCH '  INCLUDE LOAD(DFSUCMN0)'
                                  3302+          PUNCH '  INCLUDE LOAD(DFSUCCT0)'
                                  3303+          PUNCH '  INCLUDE LOAD(DFSUC150)'
                                  3304+          PUNCH '  INCLUDE LOAD(DFSUC350)'
                                  3305+          PUNCH '  INCLUDE LOAD(DFSUCER0)'
                                  3306+          PUNCH '  INCLUDE LOAD(DFSCUMM0)'
                                  3307+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                  3308+          PUNCH '  NAME DFSUCUM0(R)'
                                  3309+          PUNCH '  INCLUDE LOAD(DFSUDMP0)'
                                  3310+          PUNCH '  INCLUDE LOAD(DFSDMPM0)'
                                  3311+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                  3312+          PUNCH '  NAME DFSUDMP0(R)           IMAGE DUMP'
                                  3313+          PUNCH '  INCLUDE LOAD(DFSUACB0)'
                                  3314+          PUNCH '  INCLUDE LOAD(DFSUSCH0)'
                                  3315+          PUNCH '  INCLUDE LOAD(DFSUMGT0)'
                                  3316+          PUNCH '  INCLUDE LOAD(DFSUMSG0)'
                                  3317+          PUNCH '  INCLUDE LOAD(DFSUSRC0)'
                                  3318+          PUNCH '  INCLUDE LOAD(DFSUAMB0)'
                                  3319+          PUNCH '  ENTRY DFSUACB0'
                                  3320+          PUNCH '  NAME DFSUACB0(R)             ACBLIB UTILITY '
                                  3321+          PUNCH '  INCLUDE LOAD(DFSLBLM0) '
                                  3322+          PUNCH '  NAME DFSLBLM0(R)       BLOCK BUILDER MSG ROUTER '
                                  3323+          PUNCH '  INCLUDE LOAD(DFSURUL0)'
                                  3324+          PUNCH '  INCLUDE LOAD(DFSAOS10)'
                                  3325+          PUNCH '  INCLUDE LOAD(DFSRULM0)'
                                  3326+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                  3327+          PUNCH '  NAME DFSURUL0(R)           HISAM REORG UNLOAD'
                                  3328+          PUNCH '  INCLUDE LOAD(DFSURRL0)'
                                  3329+          PUNCH '  INCLUDE LOAD(DFSRRLM0)'
                                  3330+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                  3331+          PUNCH '  NAME DFSURRL0(R)           HISAM REORG RE-LOAD'
                                  3332+          PUNCH '  INCLUDE LOAD(DFSURDB0)'
                                  3333+          PUNCH '  INCLUDE LOAD(DFSRDBM0)'
                                  3334+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                  3335+          PUNCH '  NAME DFSURDB0(R)           DATABASE RECOVERY'
                                  3336+          PUNCH '  INCLUDE LOAD(DFSBIND0)'
                                  3337+          PUNCH '  INCLUDE LOAD(DFSDBLM0)'
                                  3338+          PUNCH '  INCLUDE LOAD(DFSDBLD0)'
                                  3339+          PUNCH '  INCLUDE LOAD(DFSDBLP0)'
                                  3340+          PUNCH '  INCLUDE LOAD(DFSDBLR0)'
                                  3341+          PUNCH '  ENTRY DFSBIND0'
                                  3342+          PUNCH '  NAME  DFSBBLD0(R)          BATCH ACB BLOCKS LOADER'
                                  3343+          PUNCH '  INCLUDE LOAD(DFSURGU0)'
                                  3344+          PUNCH '  INCLUDE LOAD(DFSLI000)'
                                  3345+          PUNCH '  INCLUDE LOAD(DFSRGUM0)'
                                  3346+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                  3347+          PUNCH '  NAME DFSURGU0(R)           HDAM REORG UNLOAD'
                                  3348+          PUNCH '  INCLUDE LOAD(DFSURGL0)'
                                  3349+          PUNCH '  INCLUDE LOAD(DFSUEX10)'
                                  3350+          PUNCH '  INCLUDE LOAD(DFSLI000)'
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71  10/02/72
                                  3351+           PUNCH '  INCLUDE LOAD(DFSRGLMC)'
                                  3352+           PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                  3353+           PUNCH '  NAME DFSURGL0(R)              HDAM REORG RE-LOAD'
                                  3354+           PUNCH '  INCLUDE LOAD(DFSURGP0)'
                                  3355+           PUNCH '  INCLUDE LOAD(DFSLI000)'
                                  3356+           PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                  3357+           PUNCH '  INCLUDE LOAD(DFSURGM0)'
                                  3358+           PUNCH '  NAME DFSURGP0(R)              REORG PREFIX UPDATE'
                                  3359+           PUNCH '  INCLUDE LOAD(DFSURG10)'
                                  3360+           PUNCH '  INCLUDE LOAD(DFSLI000)'
                                  3361+           PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                  3362+           PUNCH '  INCLUDE LOAD(DFSURGMC)'
                                  3363+           PUNCH '  NAME DFSURG10(R)              REORG SORT 1'
                                  3364+           PUNCH '  INCLUDE LOAD(DFSURGS0)'
                                  3365+           PUNCH '  INCLUDE LOAD(DFSDSEH0)'
                                  3366+           PUNCH '  INCLUDE LOAD(DFSLI000)'
                                  3367+           PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                  3368+           PUNCH '  INCLUDE LOAD(DFSURGM0)'
                                  3369+           PUNCH '  NAME DFSURGS0(R)              REORG SCAN'
                                  3370+           PUNCH '  INCLUDE LOAD(DFSURPR0)'
                                  3371+           PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                  3372+           PUNCH '  INCLUDE LOAD(DFSURGM0)'
                                  3373+           PUNCH '  NAME DFSURPR0(R)              PRE-REORG/LOAD'
                                  3374+           PUNCH '  INCLUDE LOAD(DFSERA10)'
                                  3375+           PUNCH '  NAME DFSERA10(R)              LOG PRINT SERVICE AID. '
                                  3376+           PUNCH '  INCLUDE LOAD(DFSACP10)'
                                  3377+           PUNCH '  NAME DFSACP10(R)              7770 CHANNEL PROGRAM'
                                  3378+           PUNCH '  INCLUDE LOAD(DFSUPRT0)        LOCAL SYSOUT UTILITY. '
                                  3379+           PUNCH '  NAME DFSUPRT0(R)'
                                  3380+           PUNCH '  INCLUDE LOAD(DFSIMP00)'
                                  3381+           PUNCH '  INCLUDE LOAD(DFSIMP10)'
                                  3382+           PUNCH '  INCLUDE LOAD(DFSIMP20)'
                                  3383+           PUNCH '  ENTRY DFSISMP0'
                                  3384+           PUNCH '  NAME DFSISMP0(R)              SECURITY MAINTENANCE'
                                  3385+           PUNCH '  INCLUDE LOAD(DFSFLBD0)' V954
                                  3386+           PUNCH '  ENTRY DFSFLBD1'
                                  3387+           PUNCH '  NAME DFSFLBD0(R)              STAND-ALONE LOG TERMINATORX
                                      +                '
                                  3388+           PUNCH '  INCLUDE LOAD(DFSFLOT0)'
                                  3389+           PUNCH '  INCLUDE LOAD(DFSFLMT0)'
                                  3390+           PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                  3391+           PUNCH '  ENTRY DFSFLOT0'
                                  3392+           PUNCH '  NAME DFSFLOT0(R)              SYSTEM LOG TERMINATOR '
                                  3393+           PUNCH '  INCLUDE LOAD(DFSIST10)'
                                  3394+           PUNCH '  NAME DFSIST10(R)              STATISTICS 1'
                                  3395+           PUNCH '  INCLUDE LOAD(DFSIST20)'
                                  3396+           PUNCH '  NAME DFSIST20(R)              STATISTICS 2'
                                  3397+           PUNCH '  INCLUDE LOAD(DFSIST30)'
                                  3398+           PUNCH '  NAME DFSIST30(R)              STATISTICS 3'
                                  3399+           PUNCH '  INCLUDE LOAD(DFSIST40)'
                                  3400+           PUNCH '  NAME DFSIST40(R)              STATISTICS 4'
                                  3401+           PUNCH '  INCLUDE LOAD(DFSDDLS0)'
                                  3402+           PUNCH '  INCLUDE LOAD(DFSLI000)'
                                  3403+           PUNCH '  ENTRY DLITCBL'
                                  3404+           PUNCH '  NAME DFSDDLT0(R)                DL/I TEST PROGRAM'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT    SOURCE STATEMENT                                              F01OCT71  10/02/72
                                 3405+            PUNCH '   INCLUDE LOAD(DFSBACKO)'
                                 3406+            PUNCH '   INCLUDE LCAD(DFSRDBCO)'
                                 3407+            PUNCH '   INCLUDE LOAD(DFSLIOOG)'
                                 3408+            PUNCH '   INCLUDE LOAD(DFSBACMO)'
                                 3409+            PUNCH '   ENTRY CLITCBL'
                                 3410+            PUNCH '   NAME DFSBBCCO(R)           BATCH DATABASE BACKOUT'
                                 3411+            PUNCH '   INCLUDE LOAD(DFSIXXXO)     SYSTEM DIRECTORY.'
                                 3412+            PUNCH '   INCLUDE SYSLMOD(DFSIBLKB)  SYSTEM CTRL BLOCKS.'
                                 3413+            PUNCH '   INCLUDE LOAD(DFSISVVO)     SVC MODULES.'
                                 3414+            PUNCH '   INCLUDE LOAD(DFSCFEOO)     FORMAT EDITOR INPUT SCAN.'
                                 3415+            PUNCH '   INCLUDE LOAD(DFSCFE80)     FORMAT ED. OUTPUT BUILD. '
                                 3416+            PUNCH '   INCLUDE LOAD(DFSCFEIO)     FORMAT ED. INPUT PROCESS.'
                                 3417+            PUNCH '   INCLUDE LOAD(DFSCFEPC)     FORMAT EDITOR PAGING.'
                                 3418+            PUNCH '   INCLUDE LOAD(DFSCFEGO)     FORMAT ED. OUTPUT BUFFER.'
                                 3419+            PUNCH '   INCLUDE LOAD(DFSCFEOO)     FORMAT ED. OUTPUT PROCESS'
                                 3420+            PUNCH '   INCLUDE LOAD(DFSCFESO)     FORMAT EDITOR SORT.'
                                 3421+            PUNCH '   INCLUDE LCAD(DFSCFEQC)     FORMAT EDITOR CLEAN-UP.'
                                 3422+            PUNCH '   INCLUDE LOAD(DFSFPMMO)     FORMAT POOL MANAGER. '
                                 3423+            PUNCH '   INCLUDE LCAD(DFSFPRFO)     PRE-FETCH MODULE. '
                                 3424+            PUNCH '   INCLUDE LOAD(DFSFFETO)     IMMEDIATE FETCH MODULE. '
                                 3425+            PUNCH '   INCLUDE LOAD(DFSFFRHO)     FETCH REQUEST HANDLER. '
                                 3426+            PUNCH '   INCLUDE LOAD(DFSCFEZO)     FORMAT EDITOR TRACE. '
                                 3427+            PUNCH '   INCLUDE LOAD(DFSCLMOO)     MESSAGE GENERATOR OVLY.'
                                 3428+            PUNCH '   INCLUDE LOAD(DFSCLMRO)     MESSAGE GENERATOR ROOT.'
                                 3429+            PUNCH '   INCLUDE LOAD(DFSCMT10)     MSG GENERATOR TABLE 1.'
                                 3430+            PUNCH '   INCLUDE LOAD(DFSCMT20)     MSG GENERATOR TABLE 2.'
                                 3431+            PUNCH '   INCLUDE LOAD(DFSCMT30)     MSG GENERATOR TABLE 3.'
                                 3432+            PUNCH '   INCLUDE LCAD(DFSDBLMO)     DL/I BLOCK MOVER.'
                                 3433+            PUNCH '   INCLUDE LOAD(DFSDBLDO)     DMB LOADER/RELOCATION'
                                 3434+            PUNCH '   INCLUDE LOAD(DFSDBLIC)     INTENT LIST LOAD/RELOCATE'
                                 3435+            PUNCH '   INCLUDE LOAD(DFSDBLPO)     PSB LOADER/RELOCATION'
                                 3436+            PUNCH '   INCLUDE LOAD(DFSDSSTO)     DATABASE RESOURCE MNGMT'
                                 3437+            PUNCH '   INCLUDE LCAD(DFSDBLRO)     ACBLIB READER '
                                 3438+            PUNCH '   INCLUDE LOAD(DFSDPDMO)     PSB/DMB POOL MANAGER '
                                 3439+            PUNCH '   INCLUDE LCAD(DFSDBPIC)     BUFFER POOL INIT'
                                 3440+            PUNCH '   INCLUDE LOAD(DFSFXC1C)     IMS/360 ENQUEUE'
                                 3441+            PUNCH '   INCLUDE LOAD(DFSFLOSO)     IMS/360 STAE EXIT ROUTINE'
                                 3442+            PUNCH '   INCLUDE LOAD(DFSFBGNO)     LOG INTERFACE INITIALIZATIX
                                     +                  ON.'
                                 3443+            PUNCH '   INCLUDE LOAD(DFSFLOIO)     LOG RECORDER ROUTINE.'
                                 3444+            PUNCH '   INCLUDE LOAD(DFSFLOGO)     LOG WRITER MODULE'
                                 3445+            PUNCH '   INCLUDE LOAD(DFSIASEC)     SIM REGION TERMINATION.'
                                 3446+            PUNCH '   INCLUDE LOAD(DFSTERMO)     TERMINATION CLEANUP'
                                 3447+            PUNCH '   INCLUDE LOAD(DFSIASIC)     REGION INITIATOR.'
                                 3448+            PUNCH '   INCLUDE LOAD(DFSIASTO)     REGION TERMINATOR.'
                                 3449+            PUNCH '   INCLUDE LOAD(DFSICIOC)     COMM. I/O ANALYZER.'
                                 3450+            PUNCH '   INCLUDE LOAC(DFSICLBO)     COMM. INPUT EDITOR.'
                                 3451+            PUNCH '   INCLUDE LOAD(DFSICLDO)     COMM. DISPLAY PROCESSOR.'
                                 3452+            PUNCH '   INCLUDE LOAD(DFSICLEO)     SET/RESET COMMAND.'
                                 3453+            PUNCH '   INCLUDE LCAD(DFSICLFO)     FIND DESTINATION MODULE.'
                                 3454+            PUNCH '   INCLUDE LOAD(DFSICLKC)     FORMAT COMMAND. '
                                 3455+            PUNCH '   INCLUDE LOAD(DFSICLNO)     TRACE COMMAND. '
                                 3456+            PUNCH '   INCLUDE LOAD(DFSICLPO)     COMMAND PROCESSOR.'
                                 3457+            PUNCH '   INCLUDE LOAD(DFSICLRO)     MESSAGE ROUTER.'
                                 3458+            PUNCH '   INCLUDE LOAD(DFSICLSO)     SECURITY CHECKER.'
```

```
LOC  OBJECT CODE   ADDR1 ADDR2  STMT   SOURCE STATEMENT                                     F01OCT71  10/02/72
                                3459+         PUNCH '   INCLUDE LOAD(DFSICLTO)      MESSAGE TRANSLATCR.'
                                3460+         PUNCH '   INCLUDE LOAD(DFSICLXO)      RESET POLL.'
                                3461+         PUNCH '   INCLUDE LOAD(DFSICL10)      BROADCAST COMMAND.'
                                3462+         PUNCH '   INCLUDE LOAD(DFSICL20)      CHE/RES COMMAND.'
                                3463+         PUNCH '   INCLUDE LOAD(DFSICL30)      COMMAND EDITOR.'
                                3464+         PUNCH '   INCLUDE LOAD(DFSICL40)      START COMMAND.'
                                3465+         PUNCH '   INCLUDE LOAD(DFSICL50)      TEST/EXC/END COMMAND.'
                                3466+         PUNCH '   INCLUDE LOAD(DFSICL60)      CHANGE COMMAND.'
                                3467+         PUNCH '   INCLUDE LOAD(DFSICL70)      ASSIGN COMMAND.'
                                3468+         PUNCH '   INCLUDE LOAD(DFSICL80)      DELETE COMMAND.'
                                3469+         PUNCH '   INCLUDE LOAD(DFSICL90)      LOCK/UNLOCK COMMANDS.'
                                3470+         PUNCH '   INCLUDE LOAD(DFSICLGO)      IDLE COMMAND.'
                                3471+         PUNCH '   INCLUDE LOAD(DFSICLJO)      DEQUEUE COMMAND  '
                                3472+         PUNCH '   INCLUDE LOAD(DFSICURO)      OUTPUT EDITOR.'
                                3473+         PUNCH '   INCLUDE LOAD(DFSIDP00)      DISPLAY CONTROL.'
                                3474+         PUNCH '   INCLUDE LOAD(DFSIDP10)      DISPLAY STATUS.'
                                3475+         PUNCH '   INCLUDE LOAD(DFSIDP20)      DISPLAY ACTIVE.'
                                3476+         PUNCH '   INCLUDE LOAD(DFSIDP30)      DISPLAY QUEUES.'
                                3477+         PUNCH '   INCLUDE LOAD(DFSIDP40)      DISPLAY TRAN && LTERM.'
                                3478+         PUNCH '   INCLUDE LOAD(DFSIDP50)      DISPLAY PGM && DATABASE.'
                                3479+         PUNCH '   INCLUDE LOAD(DFSIDP60)      DISPLAY LINE && PTERM.'
                                3480+         PUNCH '   INCLUDE LOAD(DFSIDP70)      DISPLAY ASSIGNMENT.'
                                3481+         PUNCH '   INCLUDE LOAD(DFSIDP80)      DISPLAY CONVERSATIONS.'
                                3482+         PUNCH '   INCLUDE LOAD(DFSIDP90)      DISPLAY SHUTDOWN STATUS.'
                                3483+         PUNCH '   INCLUDE LOAD(DFSIDPA0)      DISPLAY POOL. '
                                3484+         PUNCH '   INCLUDE LOAD(DFSIDSPO)      DISPATCHER.'
                                3485+         PUNCH '   INCLUDE LOAD(DFSIIDEO)      BLOCK DEQUEUE.'
                                3486+         PUNCH '   INCLUDE LOAD(DFSIIENO)      BLOCK ENQUEUE.'
                                3487+         PUNCH '   INCLUDE LOAD(DFSIINBO)      INIT   - COMMUNICATIONS.'
                                3488+         PUNCH '   INCLUDE LOAD(DFSIINDO)             - DMB DIRECTORY.'
                                3489+         PUNCH '   INCLUDE LOAD(DFSIINFO)             - FORMAT BUFPOOL. '
                                3490+         PUNCH '   INCLUDE LOAD(DFSIINLO)             - MODULE LOADER.'
                                3491+         PUNCH '   INCLUDE LOAD(DFSIINQC)             - QUEUE MANAGEMENT.'
                                3492+         PUNCH '   INCLUDE LOAD(DFSIINSO)             - POOL MANAGEMENT.'
                                3493+         PUNCH '   INCLUDE LOAD(DFSIINTO)             - CONTROL && MISC.'
                                3494+         PUNCH '   INCLUDE LOAD(DFSIINXO)             - RESIDENT XFR CTRL'
                                3495+         PUNCH '   INCLUDE LOAD(DFSIIN10)             - JOBLIB MODULE TBL'
                                3496+         PUNCH '   INCLUDE LOAD(DFSIIN20)             - SVCLIB MODULE TBL'
                                3497+         PUNCH '   INCLUDE LOAD(DFSIMBDO)      SMB DEQUEUE.'
                                3498+         PUNCH '   INCLUDE LOAD(DFSIMBEO)      SMB ENQUEUE.'
                                3499+         PUNCH '   INCLUDE LOAD(DFSIPCPO)      CHECKPOINT POST ROUTINE.'
                                3500+         PUNCH '   INCLUDE LOAD(DFSIPOLO)      POLLING LIST MANIPULATOR.'
                                3501+         PUNCH '   INCLUDE LOAD(DFSFTRMO)      TERMINAL LOOKUP MODULE. '
                                3502+         PUNCH '   INCLUDE LOAD(DFSIPREO)      LOG PREFIX BUILDER.'
                                3503+         PUNCH '   INCLUDE LOAD(DFSIRD10)      DISPLAY MASTER.'
                                3504+         PUNCH '   INCLUDE LOAD(DFSQMGRO)      QUEUE MANAGEMENT.'
                                3505+         PUNCH '   INCLUDE LOAD(DFSQLOGO)      QUEUE LOG BUILDER. '
                                3506+         PUNCH '   INCLUDE LOAD(DFSQRSTO)      QUEUE RESTART INTERFACE. '
                                3507+         PUNCH '   INCLUDE LOAD(DFSISMIO)      SECURITY MAINT. INIT.'
                                3508+         PUNCH '   INCLUDE LOAD(DFSISMNO)      STORAGE MANAGEMENT.'
                                3509+         PUNCH '   INCLUDE LOAD(DFSRCPOO)      CHECKPOINT MODULE.'
                                3510+         PUNCH '   INCLUDE LOAD(DFSRBOIO)      BACK OUT INTERFACE MODULE'
                                3511+         PUNCH '   INCLUDE LOAD(DFSRDBCO)      BACK OUT MODULE'
                                3512+         PUNCH '   INCLUDE LOAD(DFSREREO)      EMER RESTART OVLY MODULE.'
                                3513+         PUNCH '   INCLUDE LOAD(DFSRNREO)      NORM RESTART OVLY MODULE.'
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT                                          F01OCT71  10/02/72
                                  3514+          PUNCH '  INCLUDE LOAD(DFSRSTOO)       RESTART ROOT MODULE.'
                                  3515+          PUNCH '  INCLUDE LOAD(DFSWRAPC)       POLL OPEN/WRAP.'
                                  3516+          PUNCH '  INCLUDE TELLIB(IECTLOPN)     BTAM LINE OPEN'
                                  3517+          PUNCH '  INCLUDE TELLIB(IECTCHGN)     BTAM POLL CHANGER.'
                                  3518+          PUNCH '  INCLUDE LOAD(DFSDNSCO)       SYSTEM CONSOLE DEVICE MODUX
                                     +                LE.'
                                  3519+          PUNCH '  INCLUDE LOAD(DFSPAGEO)       TERMINAL PAGING MODULE'
                                  3520+          PUNCH '  INCLUDE LOAD(DFSDN010)       2740 DEVICE MODULE(NS).'
                                  3521+          PUNCH '  INCLUDE LOAD(DFSDN120)       2740 DEVICE MODULE(NSC). '
                                  3522+          PUNCH '  INCLUDE LOAD(DFSDS010)       2740 DEVICE MODULE(S).'
                                  3523+          PUNCH '  INCLUDE LOAD(DFSDN080)        2741 DEVICE MODULE(NS)'
                                  3524+          PUNCH '  INCLUDE LOAD(DFSDS050)        2741 DEVICE MODULE(S). '
                                  3525+          PUNCH '  INCLUDE LOAD(DFSDN110)       2980 DEVICE MODULE(NS). '
                                  3526+          PUNCH '  INCLUDE LOAD(DFSDN020)       1050 DEVICE MODULE(NS).'
                                  3527+          PUNCH '  INCLUDE LOAD(DFSDS020)       1050 DEVICE MODULE(S).'
                                  3528+          PUNCH '  INCLUDE LOAD(DFSDN030)       2260 DEVICE MODULE(REMOTE)X
                                     +                .'
                                  3529+          PUNCH '  INCLUDE LOAD(DFSDN130)       3270 REMOTE DEVICE MODULE'
                                  3530+          PUNCH '  INCLUDE LOAD(DFSDN140)       3270 LOCAL DEVICE MODULE.'
                                  3531+          PUNCH '  INCLUDE LOAD(DFSDN040)       2740-II DEVICE MODULE.'
                                  3532+          PUNCH '  INCLUDE LOAD(DFSDN060)       1030 DEVICE MODULE.'
                                  3533+          PUNCH '  INCLUDE LOAD(DFSDN090)       2770 DEVICE MODULE(NS). '
                                  3534+          PUNCH '  INCLUDE LOAD(DFSBSCKC)       BSC CHECK MODULE. '
                                  3535+          PUNCH '  INCLUDE LOAD(DFSDNS30)       2770 RWA SERVICE MODULE. '
                                  3536+          PUNCH '  INCLUDE LOAD(DFSDNS20)       2770 DEBLOCK MODULE. '
                                  3537+          PUNCH '  INCLUDE LOAD(DFSDS040)       TWX DEVICE MODULE. '
                                  3538+          PUNCH '  INCLUDE LOAD(DFSDN150)       LOCAL SYSOUT MODULE. '
                                  3539+          PUNCH '  INCLUDE LOAD(DFSII150)       LOCAL SYSOUT INIT '
                                  3540+          PUNCH '  INCLUDE LOAD(DFSICONO)       CONVERSATION PROCESSOR.'
                                  3541+          PUNCH '  INCLUDE LOAD(DFSICLHO)       HOLD/EXIT/RELEASE COMMANDSX
                                     +                .'
                                  3542+          PUNCH '  INCLUDE LOAD(DFSICLAO)       IAM ROOT(CONNECT-DISCONNECX
                                     +                T).'
                                  3543+          PUNCH '  INCLUDE LOAD(DFSICA10)       SIGN ON OVERLAY.'
                                  3544+          PUNCH '  INCLUDE LOAD(DFSII030)       7770 OPEN INITIALIZATION'
                                  3545+          PUNCH '  INCLUDE LOAD(DFSDS030)       7770 DEVICE MODULE'
                                  3546+          PUNCH '  INCLUDE LOAD(DFSIDMYO)       INSURE NO UNRESOLVES.'
                                  3547+          PUNCH '    ENTRY DFSSTART'
                                  3548+          PUNCH '  OVERLAY CHEA'
                                  3549+          PUNCH '    INSERT DFSRSTOO'
                                  3550+          PUNCH '  OVERLAY CHEA1'
                                  3551+          PUNCH '    INSERT DFSDBPIO'
                                  3552+          PUNCH '  OVERLAY CHEA'
                                  3553+          PUNCH '    INSERT DFSRCPOO'
                                  3554+          PUNCH '  OVERLAY IMSA(REGION)'
                                  3555+          PUNCH '    INSERT DFSCLMOO'
                                  3556+          PUNCH '  OVERLAY IMSA'
                                  3557+          PUNCH '    INSERT DFSICL10'
                                  3558+          PUNCH '  OVERLAY IMSA'
                                  3559+          PUNCH '    INSERT DFSICL20'
                                  3560+          PUNCH '  OVERLAY IMSA'
                                  3561+          PUNCH '    INSERT DFSICL30'
                                  3562+          PUNCH '  OVERLAY IMSA'
                                  3563+          PUNCH '    INSERT DFSICL40'
                                  3564+          PUNCH '  OVERLAY IMSA'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                   F01OCT71  10/02/72

                                3565+          PUNCH '     INSERT DFSICL50'
                                3566+          PUNCH '  OVERLAY IMSA'
                                3567+          PUNCH '     INSERT DFSICL60'
                                3568+          PUNCH '  OVERLAY IMSA'
                                3569+          PUNCH '     INSERT DFSICL70'
                                3570+          PUNCH '  OVERLAY IMSA'
                                3571+          PUNCH '     INSERT DFSICL80'
                                3572+          PUNCH '  OVERLAY IMSA'
                                3573+          PUNCH '     INSERT DFSICL90'
                                3574+          PUNCH '  OVERLAY IMSA'
                                3575+          PUNCH '     INSERT DFSICLE0'
                                3576+          PUNCH '  OVERLAY IMSA'
                                3577+          PUNCH '     INSERT DFSICLH0'
                                3578+          PUNCH '  OVERLAY IMSA'
                                3579+          PUNCH '     INSERT DFSICLG0'
                                3580+          PUNCH '  OVERLAY IMSA'
                                3581+          PUNCH '     INSERT DFSICLJ0'
                                3582+          PUNCH '  OVERLAY IMSA'
                                3583+          PUNCH '     INSERT DFSICLN0'
                                3584+          PUNCH '  OVERLAY IMSA'
                                3585+          PUNCH '     INSERT DFSIDP00'
                                3586+          PUNCH '  OVERLAY IMSA'
                                3587+          PUNCH '     INSERT DFSIDP10'
                                3588+          PUNCH '  OVERLAY IMSA'
                                3589+          PUNCH '     INSERT DFSIDP20'
                                3590+          PUNCH '  OVERLAY IMSA'
                                3591+          PUNCH '     INSERT DFSIDP30'
                                3592+          PUNCH '  OVERLAY IMSA'
                                3593+          PUNCH '     INSERT DFSIDP40'
                                3594+          PUNCH '  OVERLAY IMSA'
                                3595+          PUNCH '     INSERT DFSIDP50'
                                3596+          PUNCH '  OVERLAY IMSA'
                                3597+          PUNCH '     INSERT DFSIDP60'
                                3598+          PUNCH '  OVERLAY IMSA'
                                3599+          PUNCH '     INSERT DFSIDP70'
                                3600+          PUNCH '  OVERLAY IMSA'
                                3601+          PUNCH '     INSERT DFSIDP80'
                                3602+          PUNCH '  OVERLAY IMSA'
                                3603+          PUNCH '     INSERT DFSIDP90'
                                3604+          PUNCH '  OVERLAY IMSA'
                                3605+          PUNCH '     INSERT DFSIDPA0'
                                3606+          PUNCH '  OVERLAY IMSA'
                                3607+          PUNCH '     INSERT DFSCFEZ0'
                                3608+          PUNCH '     INSERT DFSCFEZ1'
                                3609+          PUNCH '  OVERLAY IMSA'
                                3610+          PUNCH '     INSERT DFSIRD10'
                                3611+          PUNCH '  OVERLAY IMSA'
                                3612+          PUNCH '     INSERT DFSIPCP0'
                                3613+          PUNCH '  OVERLAY IMSA'
                                3614+          PUNCH '     INSERT DFSCMT10'
                                3615+          PUNCH '  OVERLAY IMSA'
                                3616+          PUNCH '     INSERT DFSCMT20'
                                3617+          PUNCH '  OVERLAY IMSA'
                                3618+          PUNCH '     INSERT DFSCMT30'
                                3619+          PUNCH '  OVERLAY IMSA'
```



```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                   F01OCT71  10/02/72

                                3620+          PUNCH '     INSERT DFSICA10'
                                3621+          PUNCH '  OVERLAY IMSA'
                                3622+          PUNCH '     INSERT DFSIINX0'
                                3623+          PUNCH '  OVERLAY IMSB'
                                3624+          PUNCH '     INSERT DFSIINT0'
                                3625+          PUNCH '     INSERT DFSFBGN0'
                                3626+          PUNCH '  OVERLAY IMSB'
                                3627+          PUNCH '     INSERT DFSIINL0'
                                3628+          PUNCH '     INSERT DFSIIN10'
                                3629+          PUNCH '     INSERT DFSIIN20'
                                3630+          PUNCH '  OVERLAY IMSB'
                                3631+          PUNCH '     INSERT DFSIIND0'
                                3632+          PUNCH '  OVERLAY IMSB'
                                3633+          PUNCH '     INSERT DFSIINS0'
                                3634+          PUNCH '     INSERT DFSINTB0'
                                3635+          PUNCH '  OVERLAY IMSB'
                                3636+          PUNCH '     INSERT DFSIINQ0'
                                3637+          PUNCH '  OVERLAY IMSB'
                                3638+          PUNCH '     INSERT DFSIINF0'
                                3639+          PUNCH '  OVERLAY IMSB'
                                3640+          PUNCH '     INSERT DFSIINB0'
                                3641+          PUNCH '     INSERT DFSII030'
                                3642+          PUNCH '     INSERT DFSII150'
                                3643+          PUNCH '  OVERLAY IMSA '
                                3644+          PUNCH '     INSERT DFSIASE0'
                                3645+          PUNCH '  OVERLAY IMSA'
                                3646+          PUNCH '     INSERT DFSTERM0'
                                3647+          PUNCH '  OVERLAY IMSA'
                                3648+          PUNCH '     INSERT DFSRNRE0'
                                3649+          PUNCH '  OVERLAY IMSD'
                                3650+          PUNCH '     INSERT DFSISMI0'
                                3651+          PUNCH '  OVERLAY IMSA'
                                3652+          PUNCH '     INSERT DFSRERE0'
                                3653+          PUNCH '  OVERLAY IMSA'
                                3654+          PUNCH '           INSERT DFSRBOIC'
                                3655+          PUNCH '           INSERT DFSRDBC0'
                                3656+          PUNCH '  OVERLAY IMSA'
                                3657+          PUNCH '      INSERT IECTLOPN'
                                3658+          PUNCH '  OVERLAY IMSA'
                                3659+          PUNCH '      INSERT IECTCHGN'
                                3660+          PUNCH '  NAME DFSCNUCB(R)'
                                3661+          PUNCH '/*'
                                3662+          PUNCH '//'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                           F01OCT71  10/02/72

                                 3664                 *,*******************************************************
                                 3665                 *,*******************************************************
                                 3666                 *,**                                                    **
                                 3667                 *,**        SUCCESSFUL IMS SYSTEM DEFINITION            **
                                 3668                 *,**                FOR ALL FUNCTIONS.                  **
                                 3669                 *,
                                 3670                 *,     ---  BEFORE STAGE II EXECUTION  ---
                                 3671                 *,
                                 3672                 *,        THE FOLLOWING IMS DATA SETS MUST BE ALLO-
                                 3673                 *,   CATED AND CATALOGED BEFORE ATTEMPTING TO EXECUTE
                                 3674                 *,   THE GENERATED STAGE II JOB STREAM:
                                 3675                 *,        IMS2.LOAD     IMS2.GENLIB    IMS2.RESLIB
                                 3676                 *,        IMS2.MACLIB   IMS2.PROCLIB
                                 3677                 *,        IMS2.BLKS50   ICS.CLOD
                                 3678                 *,
                                 3679                 *,        OTHER REQUIRED DATA SETS MUST BE ALLOCATED
                                 3680                 *,   AND CATALOGED PRIOR TO EXECUTION OF THE IMS
                                 3681                 *,   CONTROL PROGRAM.
                                 3682                 *,
                                 3683                 *,        SPECIFIED USER SUPPLIED MODULES MUST RESIDE
                                 3684                 *,   IN ICS.CLOD BEFORE EXECUTION OF THE
                                 3685                 *,   CONTROL BLOCK LINK EDIT STEP OF STAGE II.
                                 3686                 *,
                                 3687                 *,        1030 LINE GROUPS REQUIRE LOAD MEMBER NAMED
                                 3688                 *,   (DFS10300) AND MUST RESIDE IN ICS.CLOD
                                 3689                 *,   BEFORE EXECUTION OF THE CONTROL BLOCK LINK EDIT
                                 3690                 *,   STEP OF STAGE II.
                                 3691                 *,
                                 3692                 *,        2980 LINE GROUPS REQUIRE LOAD MEMBER NAMED
                                 3693                 *,   (DFS29800) AND MUST RESIDE IN ICS.CLOD
                                 3694                 *,   BEFORE EXECUTION OF THE CONTROL BLOCK LINK EDIT
                                 3695                 *,   STEP OF STAGE II.
                                 3696                 *,
                                 3697                 *,
                                 3698                 *,        ----  BEFORE SYSTEM EXECUTION  ----
                                 3699                 *,
                                 3700                 *,        THE FOLLOWING FUNCTIONS MUST BE COMPLETED
                                 3701                 *,   BEFORE SUCCESSFUL EXECUTION OF THE IMS CONTROL
                                 3702                 *,   PROGRAM CAN BE ACHIEVED:
                                 3703                 *,
                                 3704                 *,      THE REQUIRED TYPE 4 SVC AND APPENDAGE MODULES
                                 3705                 *,      MUST BE PLACED IN SYS1.SVCLIB.
                                 3706                 *,
                                 3707                 *,      THE RESIDENT SVC INTERFACE MODULES MUST BE
                                 3708                 *,     LINK EDITED WITH THE OS NUCLEUS.  THESE
                                 3709                 *,      MODULES ARE PLACED IN IMS2.RESLIB BY THE
                                 3710                 *,      STAGE II JOB STREAM AND ARE NAMED IGC234
                                 3711                 *,      AND IGC248.
                                 3712                 *,
                                 3713                 *,      THE LOAD MEMBER DFSRRC00 MUST BE ADDED TO THE
                                 3714                 *,      'PRIVILIGED PROGRAM NAME LIST' MEMBER (IEEVLNKT)
                                 3715                 *,      IN SYS1.LINKLIB.
                                 3716                 *,
                                 3717                 *,      THE IMS2.RESLIB DATASET MUST BE CONCATENATED WITH
                                 3718                 *,      SYS1.LINKLIB BY ADDING THE NAME 'IMS2.RESLIB' TO
```



```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                           F01OCT71  10/02/72

                                 3719                 *,      THE LNKLST00 MEMBER OF SYS1.PARMLIB.
                                 3720                 *,
                                 3721                 *,      THE TERMINAL CONVERSATION FEATURE REQUIRES THE
                                 3722                 *,      IMS2.SPA DATA SET BE ALLOCATED AND CATALOGED,
                                 3723                 *,      WITH SPACE SPECIFICATION - SPACE=(0,8).
                                 3724                 *,
                                 3725                 *,      THE CATALOGED PROCEDURE 'IMS' MUST BE UPDATED
                                 3726                 *,      TO INCLUDE DD STATEMENTS FOR ON-LINE DATABASES
                                 3727                 *,      AND PLACED IN SYS1.PROCLIB BEFORE THIS PROCEDURE
                                 3728                 *,      CAN BE SUCCESSFULLY EXECUTED.
                                 3729                 *,
                                 3730                 *,      THE CATALOGED PROCEDURE 'IMSRDR' MUST BE PLACED
                                 3731                 *,      IN SYS1.PROCLIB BEFORE ANY OTHER IMS GENERATED
                                 3732                 *,      PROCEDURE CAN BE SUCCESSFULLY EXECUTED.
                                 3733                 *,
                                 3734                 *,          REFER TO THE IMS SYSTEM PROGRAMMING
                                 3735                 *,      REFERENCE MANUAL FOR INSTRUCTIONS AND ASSIST-
                                 3736                 *,      ANCE IN PERFORMING THESE AND OTHER REQUIRED
                                 3737                 *,      FUNCTIONS BEFORE SYSTEM EXECUTION.
                                 3738                 *,**                                                    **
                                 3739                 *,**                                                    **
                                 3740                 *,*******************************************************
                                 3741                 *,*******************************************************
                                 3742         END

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
*STATISTICS*    SOURCE RECORDS (SYSIN) =   360      SOURCE RECORDS (SYSLIB) =  8589
*OPTIONS IN EFFECT*  LIST, NODECK, NOLOAC, NCRENT, XREF, NOTEST, ALGN, OS, NOTERM, LINECNT =  55
 4104 PRINTED LINES
```

BATCH DATA BASE EXAMPLE

This example illustrates the output from Stage 1 of IMS/360 system definition. The input to Stage 1 (that is, the control cards) is provided in the output listing as is a summary of the Data Set and Data Base Specifications, followed by the punch statements and warning comments at the end.

```
                                                                                    PAGE    1

LOC  OBJECT CODE      ADDR1 ADDR2  STMT    SOURCE STATEMENT                    F01OCT71  10/10/72

                                    1          IMSCTRL    SYSTEM=(MVT,BATCH)

                                    3                *,        BATCH IMS SYSTEM FUNCTIONS ARE SELECTED
                                    4                *,        FOR A MVT OS PROGRAMMING SYSTEM.

                                    6          IMSGEN    ,
                                    7+         TITLE 'IMS SYSTEM DEFINITION.                                 X
    IMS SYSTEM DEFINITION.                                        V2  R3  CL00   14NOV72         PAGE    2

LOC  OBJECT CODE      ADDR1 ADDR2  STMT    SOURCE STATEMENT                    F01OCT71  10/10/72

                                    8                *,  OS SYSTEM OPTIONS:

                                   10                *,        MVT SYSTEM
                                   11                *,        RESV'D SVC'S (253,254,255)
                                   12                *,        RESV'D APNDG IGG019Z9
                                   13                *,        RESV'D SQS   (500,1000)

                                   15                *,  IMS/360 SYSTEM OPTIONS:

                                   17                *,        1 REGIONS
                                   18                *,        1 SUBTASKS
                                   19                *,        18 CONCURRENT EXCTL REQUESTS
                                   20                *,        1000 CHECKPOINT LOG FREQUENCY
                                   21                *,        GENERATE FOR BATCH
    IMS SYSTEM DEFINITION.                                        V2  F3  CL00   14NOV72         PAGE    3

LOC  OBJECT CODE      ADDR1 ADDR2  STMT    SOURCE STATEMENT                    F01OCT71  10/10/72

                                   23+           PUNCH '//IMSGEN1 JOB MSGLEVEL=1,MSGCLASS=A,CLASS=A'
                                   24+           PUNCH '//STEP1 EXEC PGM=IEBCOPY'
                                   25+           PUNCH '//SYSPRINT DD SYSOUT=A'
                                   26+           PUNCH '//SYSUT1 DD DSN=IMS2.GENLIB,DISP=SHR'
                                   27+           PUNCH '//SYSUT2 DD DSN=IMS2.MACLIB,DISP=OLD'
                                   28+           PUNCH '//SYSUT3 DD UNIT=SYSDA,SPACE=(80,(10,15))  '
                                   29+           PUNCH '//SYSUT4 DD UNIT=SYSDA,SPACE=(256,(5,1))   '
                                   30+           PUNCH '//SYSIN DD *'
                                   31+           PUNCH '  COPY OUTDD=SYSUT2,INDD=SYSUT1  '
                                   32+           PUNCH '  SELECT MEMBER=ALPHA '
                                   33+           PUNCH '  SELECT MEMBER=CKDDN '
                                   34+           PUNCH '  SELECT MEMBER=CKOPT '
                                   35+           PUNCH '  SELECT MEMBER=CONVERT '
                                   36+           PUNCH '  SELECT MEMBER=DATASET '
                                   37+           PUNCH '  SELECT MEMBER=DBD '
                                   38+           PUNCH '  SELECT MEMBER=DBDGEN '
                                   39+           PUNCH '  SELECT MEMBER=DBDLRECL '
                                   40+           PUNCH '  SELECT MEMBER=DEVSIZE '
                                   41+           PUNCH '  SELECT MEMBER=DMAN '
                                   42+           PUNCH '  SELECT MEMBER=EXPARMS '
                                   43+           PUNCH '  SELECT MEMBER=EXTDBD '
                                   44+           PUNCH '  SELECT MEMBER=FIELD '
                                   45+           PUNCH '  SELECT MEMBER=FINISH '
                                   46+           PUNCH '  SELECT MEMBER=FLD '
                                   47+           PUNCH '  SELECT MEMBER=FLDK '
                                   48+           PUNCH '  SELECT MEMBER=GLOBALS '
                                   49+           PUNCH '  SELECT MEMBER=HIERSEQ '
                                   50+           PUNCH '  SELECT MEMBER=LCHILD '
                                   51+           PUNCH '  SELECT MEMBER=PCB '
                                   52+           PUNCH '  SELECT MEMBER=PCBPDV '
                                   53+           PUNCH '  SELECT MEMBER=PSBGEN '
                                   54+           PUNCH '  SELECT MEMBER=SEGM '
                                   55+           PUNCH '  SELECT MEMBER=SEGPTRS '
                                   56+           PUNCH '  SELECT MEMBER=SENSEG '
                                   57+           PUNCH '  SELECT MEMBER=SETFLGS '
                                   58+           PUNCH '  SELECT MEMBER=SETFREQ '
                                   59+           PUNCH '  SELECT MEMBER=SOURSEG '
                                   60+           PUNCH '  SELECT MEMBER=XDFLD '
                                   61+           PUNCH '//STEP2 EXEC PGM=IEBUPDTE,'
                                   62+           PUNCH '// PARM=NEW'
                                   63+           PUNCH '//SYSPRINT DD  SYSOUT=A'
                                   64+           PUNCH '//SYSUT2 DD DSN=IMS2.PROCLIB,DISP=OLD'
                                   65+           PUNCH '//SYSIN  DD    DATA'
                                   66+           PUNCH './       ADD   NAME=CBLTDLI'
                                   67+           PUNCH '  LIBRARY RESLIB(CBLTDLI)    DL/I LANGUAGE INTERFACE'
                                   68+           PUNCH '  ENTRY DLITCBL'
                                   69+           PUNCH './       ADD   NAME=PLITDLI'
                                   70+           PUNCH '  LIBRARY RESLIB(PLITDLI)    DL/I LANGUAGE INTERFACE'
                                   71+           PUNCH '  ENTRY IHESAPD'
                                   72+           PUNCH './       ADD   NAME=DLIBATCH'
                                   73+           PUNCH './       NUMBER NEW1=10,INCR=10'
                                   74+           PUNCH '//       PROC   MBR=TEMPNAME,SOUT=A,PSB=,BUF=,SPIE=0,TEX
                                    +                ST=0'
                                   75+           PUNCH '//G      EXEC  PGM=DFSRRC00,REGION=130K,'
                                   76+           PUNCH '//             PARM=''DLI,&&MBR,&&PSB,&&BUF,&&SPIE&&TESX
```

```
LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                           F01OCT71  10/10/72
                                     +              T'''
                                   77+            PUNCH '//STEPLIB DD DSN=IMS2.RESLIB,DISP=SHR'
                                   78+            PUNCH '//        DD DSN=IMS2.PGMLIB,DISP=SHR'
                                   79+            PUNCH '//IMS     DD DSN=IMS2.PSBLIB,DISP=SHR'
                                   80+            PUNCH '//        DD DSN=IMS2.DBDLIB,DISP=SHR'
                                   81+            PUNCH '//DFSLOGTT DD DSN=IMS2.LOGT,DISP=SHR'
                                   82+            PUNCH '//IEFRDER DD DSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),UNIT=(X
                                     +              2400,,DEFER),'
                                   83+            PUNCH '// DCB=(RECFM=VBS,BLKSIZE=1408,LRECL=1400,BUFNO=1)'
                                   84+            PUNCH '//SYSUDUMP DD SYSOUT=&&SOUT,DCB=(RECFM=FBA,LRECL=121,BLX
                                     +              KSIZE=605),'
                                   85+            PUNCH '// SPACE=(605,(500,500),RLSE,,ROUND)'
                                   86+            PUNCH './         ADD    NAME=DBBBATCH'
                                   87+            PUNCH './         NUMBER NEW1=10,INCR=10'
                                   88+            PUNCH '//         PROC   MBR=TEMPNAME,SOUT=A,PSB=,BUF=8,SPIE=0,TEX
                                     +              ST=0'
                                   89+            PUNCH '//G        EXEC   PGM=DFSRRC00,REGION=130K,'
                                   90+            PUNCH '//                PARM=''DBB,&&MBR,&&PSB,&&BUF,&&SPIE&&TESX
                                     +              T'''
                                   91+            PUNCH '//STEPLIB DD DSN=IMS2.RESLIB,DISP=SHR'
                                   92+            PUNCH '//        DD DSN=IMS2.PGMLIB,DISP=SHR'
                                   93+            PUNCH '//IMSACB  DD  DSN=IMS2.ACBLIB,DISP=SHR'
                                   94+            PUNCH '//DFSLOGTT DD DSN=IMS2.LOGT,DISP=SHR'
                                   95+            PUNCH '//IEFRDER DD DSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),UNIT=(X
                                     +              2400,,DEFER),'
                                   96+            PUNCH '// DCB=(RECFM=VBS,BLKSIZE=1408,LRECL=1400,BUFNO=1)'
                                   97+            PUNCH '//SYSUDUMP DD SYSOUT=&&SOUT,DCB=(RECFM=FBA,LRECL=121,BLX
                                     +              KSIZE=605),'
                                   98+            PUNCH '// SPACE=(605,(500,500),RLSE,,ROUND)'
                                   99+            PUNCH './         ADD  NAME=ACBGEN'
                                  100+            PUNCH './         NUMBER NEW1=10,INCR=10'
                                  101+            PUNCH '//         PROC  SOUT=A,COMP=,RGN=100K'
                                  102+            PUNCH '//G        EXEC  PGM=DFSRRC00,PARM=''UPB,&&COMP'',REGION=X
                                     +              &&RGN'
                                  103+            PUNCH '//SYSPRINT DD SYSOUT=&&SOUT'
                                  104+            PUNCH '//IMS          DD DSN=IMS2.PSBLIB,DISP=SHR'
                                  105+            PUNCH '//             DD DSN=IMS2.DBDLIB,DISP=SHR'
                                  106+            PUNCH '//IMSACB       DD DSN=IMS2.ACBLIB,DISP=OLD'
                                  107+            PUNCH '//SYSUT3       DD UNIT=SYSDA,SPACE=(80,(100,100))'
                                  108+            PUNCH '//SYSUT4       DD UNIT=SYSDA,SPACE=(256,(100,100)),DCB=KEYLX
                                     +              EN=8'
                                  109+            PUNCH '//COMPCTL  DD DSN=IMS2.PROCLIB(DFSACBCP),DISP=SHR'
                                  110+            PUNCH './         ADD  NAME=DFSACBCP'
                                  111+            PUNCH './         NUMBER NEW1=10,INCR=10'
                                  112+            PUNCH '            COPY  INDD=IMSACB,OUTDD=IMSACB'
                                  113+            PUNCH './         ADD    NAME=PSBGEN'
                                  114+            PUNCH './         NUMBER NEW1=10,INCR=10'
                                  115+            PUNCH '//         PROC   MBR=TEMPNAME,SOUT=A'
                                  116+            PUNCH '//C        EXEC   PGM=IEUASM,REGION=120K,PARM=''LOAD,NODECX
                                     +              K'''
                                  117+            PUNCH '//SYSLIB DD      DSN=IMS2.MACLIB,DISP=SHR'
                                  118+            PUNCH '//SYSGO  DD      UNIT=SYSDA,DISP=(,PASS),SPACE=(80,(100,1X
                                     +              00),RLSE),'
                                  119+            PUNCH '//               DCB=(BLKSIZE=400,RECFM=FB,LRECL=80)'
                                  120+            PUNCH '//SYSPRINT DD    SYSOUT=&&SOUT,DCB=(LRECL=121,RECFM=FBM,BX
```

```
LOC  OBJECT CODE      ADDR1 ADDR2  STMT    SOURCE STATEMENT                                              F01OCT71  10/10/72
                                      +              LKSIZE=605),'
                                   121+        PUNCH '//                SPACE=(121,(500,500),RLSE,,ROUND)'
                                   122+        PUNCH '//SYSUT1 DD       UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(1X
                                      +              00,50))'
                                   123+        PUNCH '//SYSUT2 DD       UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(1X
                                      +              00,50))'
                                   124+        PUNCH '//SYSUT3 DD       UNIT=(SYSDA,SEP=(SYSLIB,SYSUT1,SYSUT2)),X
                                      +              '
                                   125+        PUNCH '//                SPACE=(1700,(100,50))'
                                   126+        PUNCH '//L        EXEC   PGM=DFSILNK0,PARM=''XREF,LIST'',COND=(0,X
                                      +              LT,C),REGION=120K'
                                   127+        PUNCH '//STEPLIB DD      DSN=IMS2.RESLIB,DISP=SHR'
                                   128+        PUNCH '//SYSLIN DD       DSN=*.C.SYSGO,DISP=(OLD,DELETE)'
                                   129+        PUNCH '//SYSPRINT DD     SYSOUT=&&SOUT,DCB=(LRECL=121,RECFM=FBA,BX
                                      +              LKSIZE=605),'
                                   130+        PUNCH '//                SPACE=(121,(100,100),RLSE)'
                                   131+        PUNCH '//SYSLMOD DD      DSN=IMS2.PSBLIB(&&MBR),DISP=SHR'
                                   132+        PUNCH '//SYSUT1 DD       UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),DISP=(X
                                      +              ,DELETE),'
                                   133+        PUNCH '//                SPACE=(1024,(100,10),RLSE)'
                                   134+        PUNCH './        ADD     NAME=DBDGEN'
                                   135+        PUNCH './        NUMBER  NEW1=10,INCR=10'
                                   136+        PUNCH '//        PROC    MBR=TEMPNAME,SOUT=A'
                                   137+        PUNCH '//C       EXEC    PGM=IEUASM,REGION=120K,PARM=''LOAD,NODECX
                                      +              K'''
                                   138+        PUNCH '//SYSLIB DD       DSN=IMS2.MACLIB,DISP=SHR'
                                   139+        PUNCH '//SYSGO  DD       UNIT=SYSDA,DISP=(,PASS),SPACE=(80,(100,1X
                                      +              00),RLSE),'
                                   140+        PUNCH '//                DCB=(BLKSIZE=400,RECFM=FB,LRECL=80)'
                                   141+        PUNCH '//SYSPRINT DD     SYSOUT=&&SOUT,DCB=(LRECL=121,RECFM=FBM,BX
                                      +              LKSIZE=605),'
                                   142+        PUNCH '//                SPACE=(121,(500,500),RLSE,,ROUND)'
                                   143+        PUNCH '//SYSUT1 DD       UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(1X
                                      +              00,50))'
                                   144+        PUNCH '//SYSUT2 DD       UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(1X
                                      +              00,50))'
                                   145+        PUNCH '//SYSUT3 DD       UNIT=(SYSDA,SEP=(SYSLIB,SYSUT1,SYSUT2)),X
                                      +              '
                                   146+        PUNCH '//                SPACE=(1700,(100,50))'
                                   147+        PUNCH '//L        EXEC   PGM=DFSILNK0,PARM=''XREF,LIST'',COND=(0,X
                                      +              LT,C),REGION=120K'
                                   148+        PUNCH '//STEPLIB DD      DSN=IMS2.RESLIB,DISP=SHR'
                                   149+        PUNCH '//SYSLIN DD       DSN=*.C.SYSGO,DISP=(OLD,DELETE)'
                                   150+        PUNCH '//SYSPRINT DD     SYSOUT=&&SOUT,DCB=(LRECL=121,RECFM=FBA,BX
                                      +              LKSIZE=605),'
                                   151+        PUNCH '//                SPACE=(121,(100,100),RLSE)'
                                   152+        PUNCH '//SYSLMOD DD      DSN=IMS2.DBDLIB(&&MBR),DISP=SHR'
                                   153+        PUNCH '//SYSUT1 DD       UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),DISP=(X
                                      +              ,DELETE),'
                                   154+        PUNCH '//                SPACE=(1024,(100,10),RLSE)'
                                   155+        PUNCH './        ADD     NAME=IMSCOBOL'
                                   156+        PUNCH './        NUMBER  NEW1=10,INCR=10'
                                   157+        PUNCH '//        PROC    MBR=,PAGES=60,SOUT=A'
                                   158+        PUNCH '//C       EXEC    PGM=IKFCBL00,REGION=150K,'
                                   159+        PUNCH '//                PARM=''SIZE=130K,BUF=10K,LINECNT=50'''
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                            F01OCT71  10/10/72

                                 160+          PUNCH '//SYSLIN DD      DSN=&&&&LIN,DISP=(MOD,PASS),UNIT=SYSDA,'
                                 161+          PUNCH '//               DCB=(LRECL=80,RECFM=FB,BLKSIZE=400),'
                                 162+          PUNCH '//               SPACE=(3520,(40,10),RLSE,,ROUND)'
                                 163+          PUNCH '//SYSPRINT DD    SYSOUT=&&SOUT,DCB=(LRECL=121,BLKSIZE=605X
                                   +                ,RECFM=FBA),'
                                 164+          PUNCH '//               SPACE=(605,(&&PAGES.0,&&PAGES),RLSE,,ROUX
                                   +                ND)'
                                 165+          PUNCH '//SYSUT1 DD      UNIT=SYSDA,DISP=(,DELETE),'
                                 166+          PUNCH '//               SPACE=(3520,(100,10),RLSE,,ROUND)'
                                 167+          PUNCH '//SYSUT2 DD      UNIT=SYSDA,DISP=(,DELETE),'
                                 168+          PUNCH '//               SPACE=(3520,(100,10),RLSE,,ROUND)'
                                 169+          PUNCH '//SYSUT3 DD      UNIT=SYSDA,DISP=(,DELETE),'
                                 170+          PUNCH '//               SPACE=(3520,(100,10),RLSE,,ROUND)'
                                 171+          PUNCH '//SYSUT4 DD      UNIT=SYSDA,DISP=(,DELETE),'
                                 172+          PUNCH '//               SPACE=(3520,(100,10),RLSE,,ROUND)'
                                 173+          PUNCH '//L        EXEC  PGM=DFSILNK0,REGION=120K,PARM=''XREF,LETX
                                   +                ,LIST'','
                                 174+          PUNCH '//               COND=(4,LT,C)'
                                 175+          PUNCH '//STEPLIB DD     DSN=IMS2.RESLIB,DISP=SHR'
                                 176+          PUNCH '//SYSLIB DD      DSN=SYS1.COBLIB,DISP=SHR'
                                 177+          PUNCH '//         DD    DSN=SYS1.PL1LIB,DISP=SHR'
                                 178+          PUNCH '//RESLIB DD      DSN=IMS2.RESLIB,DISP=SHR'
                                 179+          PUNCH '//SYSLIN DD      DSN=&&&&LIN,DISP=(OLD,DELETE),VOL=REF=*.X
                                   +                C.SYSLIN'
                                 180+          PUNCH '//         DD    DSN=IMS2.PROCLIB(CBLTDLI),DISP=SHR'
                                 181+          PUNCH '//         DD    DDNAME=SYSIN'
                                 182+          PUNCH '//SYSLMOD DD     DSN=IMS2.PGMLIB(&&MBR),DISP=SHR'
                                 183+          PUNCH '//SYSPRINT DD    SYSOUT=&&SOUT,DCB=(RECFM=FBA,LRECL=121,BX
                                   +                LKSIZE=605),'
                                 184+          PUNCH '//               SPACE=(605,(&&PAGES.0,&&PAGES),RLSE,,ROUX
                                   +                ND)'
                                 185+          PUNCH '//SYSUT1 DD      UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),DISP=(X
                                   +                ,DELETE),'
                                 186+          PUNCH '//               SPACE=(3520,(100,10),RLSE,,ROUND)'
                                 187+          PUNCH './         ADD   NAME=IMSCOBGO'
                                 188+          PUNCH './         NUMBER NEW1=10,INCR=10'
                                 189+          PUNCH '//         PROC  MBR=,PAGES=60,SOUT=A,PSB=,SPIE=0,TEST=0,X
                                   +                BUF='
                                 190+          PUNCH '//C        EXEC  PGM=IKFCBL00,REGION=150K,'
                                 191+          PUNCH '//               PARM=''SIZE=130K,BUF=10K,LINECNT=50'''
                                 192+          PUNCH '//SYSLIN DD      DSN=&&&&LIN,DISP=(MOD,PASS),UNIT=SYSDA,'
                                 193+          PUNCH '//               DCB=(LRECL=80,RECFM=FB,BLKSIZE=400),'
                                 194+          PUNCH '//               SPACE=(3520,(40,10),RLSE,,ROUND)'
                                 195+          PUNCH '//SYSPRINT DD    SYSOUT=&&SOUT,DCB=(LRECL=121,BLKSIZE=605X
                                   +                ,RECFM=FBA),'
                                 196+          PUNCH '//               SPACE=(605,(&&PAGES.0,&&PAGES),RLSE,,ROUX
                                   +                ND)'
                                 197+          PUNCH '//SYSUT1 DD      UNIT=SYSDA,DISP=(,DELETE),'
                                 198+          PUNCH '//               SPACE=(3520,(100,10),RLSE,,ROUND)'
                                 199+          PUNCH '//SYSUT2 DD      UNIT=SYSDA,DISP=(,DELETE),'
                                 200+          PUNCH '//               SPACE=(3520,(100,10),RLSE,,ROUND)'
                                 201+          PUNCH '//SYSUT3 DD      UNIT=SYSDA,DISP=(,DELETE),'
                                 202+          PUNCH '//               SPACE=(3520,(100,10),RLSE,,ROUND)'
                                 203+          PUNCH '//SYSUT4 DD      UNIT=SYSDA,DISP=(,DELETE),'
                                 204+          PUNCH '//               SPACE=(3520,(100,10),RLSE,,ROUND)'
```

```
LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                         F01OCT71  10/10/72
                                   205+         PUNCH '//L        EXEC  PGM=DFSILNK0,REGION=120K,PARM=''XREF,LETX
                                     +                ,LIST'','
                                   206+         PUNCH '//                COND=(4,LT,C)'
                                   207+         PUNCH '//STEPLIB DD      DSN=IMS2.RESLIB,DISP=SHR'
                                   208+         PUNCH '//SYSLIB DD       DSN=SYS1.COBLIB,DISP=SHR'
                                   209+         PUNCH '//         DD     DSN=SYS1.PL1LIB,DISP=SHR'
                                   210+         PUNCH '//RESLIB DD       DSN=IMS2.RESLIB,DISP=SHR'
                                   211+         PUNCH '//SYSLIN DD       DSN=&&&&LIN,DISP=(OLD,DELETE),VOL=REF=*.X
                                     +                C.SYSLIN'
                                   212+         PUNCH '//         DD     DSN=IMS2.PROCLIB(CBLTDLI),DISP=SHR'
                                   213+         PUNCH '//         DD     DDNAME=SYSIN'
                                   214+         PUNCH '//SYSLMOD DD      DSN=IMS2.PGMLIB(&&MBR),DISP=SHR'
                                   215+         PUNCH '//SYSPRINT DD     SYSOUT=&&SOUT,DCB=(RECFM=FBA,LRECL=121,BX
                                     +                LKSIZE=605),'
                                   216+         PUNCH '//                SPACE=(605,(&&PAGES.0,&&PAGES),RLSE,,ROUX
                                     +                ND)'
                                   217+         PUNCH '//SYSUT1 DD       UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),DISP=(X
                                     +                ,DELETE),'
                                   218+         PUNCH '//                SPACE=(3520,(100,10),RLSE,,ROUND)'
                                   219+         PUNCH '//G        EXEC  PGM=DFSRRC00,REGION=150K,TIME=2,COND=(4,X
                                     +                LT),'
                                   220+         PUNCH '//                PARM=''DLI,&&MBR,&&PSB,&&BUF,&&SPIE&&TESX
                                     +                T'''
                                   221+         PUNCH '//STEPLIB DD      DSN=IMS2.RESLIB,DISP=SHR'
                                   222+         PUNCH '//         DD     DSN=IMS2.PGMLIB,DISP=SHR'
                                   223+         PUNCH '//IMS      DD     DSN=IMS2.PSBLIB,DISP=SHR'
                                   224+         PUNCH '//         DD     DSN=IMS2.DBDLIB,DISP=SHR'
                                   225+         PUNCH '//IEFRDER DD DSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),UNIT=(X
                                     +                2400,,DEFER),'
                                   226+         PUNCH '//               DCB=(RECFM=VBS,BLKSIZE=1408,LRECL=1400,BUX
                                     +                FNO=1)'
                                   227+         PUNCH '//SYSOUT DD       SYSOUT=&&SOUT,SPACE=(CYL,(1,1)),DCB=(LREX
                                     +                CL=133,RECFM=FA)'
                                   228+         PUNCH '//SYSUDUMP DD     SYSOUT=&&SOUT,DCB=(LRECL=121,RECFM=FBA,BX
                                     +                LKSIZE=3025),'
                                   229+         PUNCH '//                SPACE=(3025,(200,100),RLSE,,ROUND)'
                                   230+         PUNCH './         ADD    NAME=IMSPLI'
                                   231+         PUNCH './         NUMBER NEW1=10,INCR=10'
                                   232+         PUNCH '//         PROC   MBR=,PAGES=50,SOUT=A'
                                   233+         PUNCH '//C        EXEC   PGM=IEMAA,REGION=114K,'
                                   234+         PUNCH '//                PARM=''XREF,ATR,LOAD,NODECK,NOMACRO,,OPTX
                                     +                =1'''
                                   235+         PUNCH '//SYSUT1 DD       UNIT=SYSDA,SPACE=(1024,(60,60),RLSE,,ROUX
                                     +                ND),'
                                   236+         PUNCH '//                DCB=BLKSIZE=1024,DISP=(,DELETE)'
                                   237+         PUNCH '//SYSUT3 DD       UNIT=SYSDA,SPACE=(1024,(60,60),RLSE,,ROUX
                                     +                ND),'
                                   238+         PUNCH '//                DCB=BLKSIZE=1024,DISP=(,DELETE)'
                                   239+         PUNCH '//SYSPRINT DD     SYSOUT=&&SOUT,DCB=(LRECL=125,BLKSIZE=629X
                                     +                ,RECFM=VBA),'
                                   240+         PUNCH '//                SPACE=(605,(&&PAGES.0,&&PAGES),RLSE)'
                                   241+         PUNCH '//SYSLIN DD       UNIT=SYSDA,SPACE=(80,(250,80),RLSE),DCB=X
                                     +                BLKSIZE=80,'
                                   242+         PUNCH '//                DISP=(,PASS)'
                                   243+         PUNCH '//L        EXEC   PGM=DFSILNK0,PARM=''XREF,LIST,LET'',CONDX
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                               F01OCT71  10/10/72

                                   +             =(4,LT,C),'
                                244+          PUNCH '//                 REGION=120K'
                                245+          PUNCH '//STEPLIB DD       DSN=IMS2.RESLIB,DISP=SHR'
                                246+          PUNCH '//SYSLIB DD        DSN=SYS1.PL1LIB,DISP=SHR'
                                247+          PUNCH '//         DD      DSN=SYS1.COBLIB,DISP=SHR'
                                248+          PUNCH '//RESLIB DD        DSN=IMS2.RESLIB,DISP=SHR'
                                249+          PUNCH '//SYSLIN DD        DSN=*.C.SYSLIN,DISP=(OLD,DELETE)'
                                250+          PUNCH '//         DD      DSN=IMS2.PROCLIB(PLITDLI),DISP=SHR'
                                251+          PUNCH '//         DD      DDNAME=SYSIN'
                                252+          PUNCH '//SYSLMOD DD       DSN=IMS2.PGMLIB(&&MBR),DISP=SHR'
                                253+          PUNCH '//SYSPRINT DD      SYSOUT=&&SOUT,DCB=(LRECL=121,RECFM=FBA,BX
                                   +             LKSIZE=605),'
                                254+          PUNCH '//                 SPACE=(605,(&&PAGES.0,&&PAGES),RLSE)'
                                255+          PUNCH '//SYSUT1 DD        UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(5,X
                                   +             1),RLSE)'
                                256+          PUNCH './         ADD     NAME=IMSPLIGO'
                                257+          PUNCH './         NUMBER  NEW1=10,INCR=10'
                                258+          PUNCH '//         PROC    MBR=,PAGES=50,SOUT=A,PSB=,SPIE=0,TEST=0,X
                                   +             BUF='
                                259+          PUNCH '//C        EXEC    PGM=IEMAA,REGION=114K,'
                                260+          PUNCH '//                 PARM=''XREF,ATR,LOAD,NODECK,NOMACRO,,OPTX
                                   +             =1'''
                                261+          PUNCH '//SYSUT1 DD        UNIT=SYSDA,SPACE=(1024,(60,60),RLSE,,ROUX
                                   +             ND),'
                                262+          PUNCH '//                 DCB=BLKSIZE=1024,DISP=(,DELETE)'
                                263+          PUNCH '//SYSUT3 DD        UNIT=SYSDA,SPACE=(1024,(60,60),RLSE,,ROUX
                                   +             ND),'
                                264+          PUNCH '//                 DCB=BLKSIZE=1024,DISP=(,DELETE)'
                                265+          PUNCH '//SYSPRINT DD      SYSOUT=&&SOUT,DCB=(LRECL=125,BLKSIZE=629X
                                   +             ,RECFM=VBA),'
                                266+          PUNCH '//                 SPACE=(605,(&&PAGES.0,&&PAGES),RLSE)'
                                267+          PUNCH '//SYSLIN DD        UNIT=SYSDA,SPACE=(80,(250,80),RLSE),DCB=X
                                   +             BLKSIZE=80,'
                                268+          PUNCH '//                 DISP=(,PASS)'
                                269+          PUNCH '//L        EXEC    PGM=DFSILNK0,PARM=''XREF,LIST,LET'',CONDX
                                   +             =(4,LT,C),'
                                270+          PUNCH '//                 REGION=120K'
                                271+          PUNCH '//STEPLIB DD       DSN=IMS2.RESLIB,DISP=SHR'
                                272+          PUNCH '//SYSLIB DD        DSN=SYS1.PL1LIB,DISP=SHR'
                                273+          PUNCH '//         DD      DSN=SYS1.COBLIB,DISP=SHR'
                                274+          PUNCH '//RESLIB DD        DSN=IMS2.RESLIB,DISP=SHR'
                                275+          PUNCH '//SYSLIN DD        DSN=*.C.SYSLIN,DISP=(OLD,DELETE)'
                                276+          PUNCH '//         DD      DSN=IMS2.PROCLIB(PLITDLI),DISP=SHR'
                                277+          PUNCH '//         DD      DDNAME=SYSIN'
                                278+          PUNCH '//SYSLMOD DD       DSN=IMS2.PGMLIB(&&MBR),DISP=SHR'
                                279+          PUNCH '//SYSPRINT DD      SYSOUT=&&SOUT,DCB=(LRECL=121,RECFM=FBA,BX
                                   +             LKSIZE=605),'
                                280+          PUNCH '//                 SPACE=(605,(&&PAGES.0,&&PAGES),RLSE)'
                                281+          PUNCH '//SYSUT1 DD        UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(5,X
                                   +             1),RLSE)'
                                282+          PUNCH '//G        EXEC    PGM=DFSRRC00,REGION=150K,TIME=5,COND=(4,X
                                   +             LT),'
                                283+          PUNCH '//                 PARM=''DLI,&&MBR,&&PSB,&&BUF,&&SPIE&&TESX
                                   +             T'''
                                284+          PUNCH '//STEPLIB DD       DSN=IMS2.RESLIB,DISP=SHR'
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                F01OCT71  10/10/72
                                  285+            PUNCH '//          DD     DSN=IMS2.PGMLIB,DISP=SHR'
                                  286+            PUNCH '//IMS       DD     DSN=IMS2.PSBLIB,DISP=SHR'
                                  287+            PUNCH '//          DD     DSN=IMS2.DBDLIB,DISP=SHR'
                                  288+            PUNCH '//IEFRDER DD DSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),UNIT=(X
                                     +                  2400,,DEFER),'
                                  289+            PUNCH '//                 DCB=(RECFM=VBS,BLKSIZE=1408,LRECL=1400,BUX
                                     +                  FNO=1)'
                                  290+            PUNCH '//SYSPRINT DD  SYSOUT=&&SOUT,DCB=(LRECL=121,BLKSIZE=605X
                                     +                  ,RECFM=FBA),'
                                  291+            PUNCH '//                 SPACE=(605,(500,500),RLSE,,ROUND)'
                                  292+            PUNCH '//SYSUDUMP DD  SYSOUT=&&SOUT,DCB=(LRECL=121,BLKSIZE=605X
                                     +                  ,RECFM=FBA),'
                                  293+            PUNCH '//                 SPACE=(605,(500,500),RLSE,,ROUND)'
                                  294+            PUNCH './         ADD    NAME=MFDBDUMP'
                                  295+            PUNCH './         NUMBER NEW1=10,INCR=10 '
                                  296+            PUNCH '//         PROC SOUT=A'
                                  297+            PUNCH '//DUMP     EXEC  PGM=DFSRRC00,PARM=''DLI,DFSSAM08'',REGIOX
                                     +                  N=130K'
                                  298+            PUNCH '//STEPLIB DD DSN=IMS2.RESLIB,DISP=SHR'
                                  299+            PUNCH '//          DD DSN=IMS2.PGMLIB,DISP=SHR'
                                  300+            PUNCH '//IMS       DD DSN=IMS2.PSBLIB,DISP=SHR'
                                  301+            PUNCH '//          DD DSN=IMS2.DBDLIB,DISP=SHR'
                                  302+            PUNCH '//SYSUDUMP DD SYSOUT=&&SOUT'
                                  303+            PUNCH '//DI21PART DD DSN=IMS2.DI21PART,DISP=SHR'
                                  304+            PUNCH '//DI21PARO DD DSN=IMS2.DI21PARO,DISP=SHR'
                                  305+            PUNCH '//OUTPUT DD    SYSOUT=&&SOUT'
                                  306+            PUNCH './         ADD  NAME=MFDBLOAD'
                                  307+            PUNCH './         NUMBER NEW1=10,INCR=10'
                                  308+            PUNCH '//         PROC SOUT=A'
                                  309+            PUNCH '//LOAD     EXEC  PGM=DFSRRC00,PARM=''DLI,DFSSAM01'',REGIOX
                                     +                  N=130K'
                                  310+            PUNCH '//STEPLIB DD DSN=IMS2.RESLIB,DISP=SHR'
                                  311+            PUNCH '//          DD DSN=IMS2.PGMLIB,DISP=SHR'
                                  312+            PUNCH '//IMS       DD DSN=IMS2.PSBLIB,DISP=SHR'
                                  313+            PUNCH '//          DD DSN=IMS2.DBDLIB,DISP=SHR'
                                  314+            PUNCH '//SYSUDUMP DD SYSOUT=&&SOUT'
                                  315+            PUNCH '//DI21PART DD DSN=IMS2.DI21PART(PRIME),DISP=(,KEEP),DCBX
                                     +                  =DSORG=IS,        X'
                                  316+            PUNCH '//                 SPACE=(CYL,3,,CONTIG),VOL=SER=&&PSER,UNIX
                                     +                  T=&&PUNIT'
                                  317+            PUNCH '//DI21PARO DD DSN=IMS2.DI21PARO,DISP=(,KEEP),SPACE=(CYLX
                                     +                  ,3,,CONTIG),      X'
                                  318+            PUNCH '//                 VOL=SER=&&OSER,UNIT=&&OUNIT'
                                  319+            PUNCH '//SYSOUT DD    SYSOUT=&&SOUT'
                                  320+            PUNCH '//INPUT DD DSN=IMS2.MACLIB(MFDFSYSN),DISP=SHR'
                                  321+            PUNCH './         ADD    NAME=IMSRDR'
                                  322+            PUNCH './         NUMBER NEW1=10,INCR=10'
                                  323+            PUNCH '//          PROC  MBR=IMSMSG'
                                  324+            PUNCH '//IEFPROC EXEC PGM=IEFIRC,           READER FIRST LOAD'
                                  325+            PUNCH '//                 REGION=48K,         READER BASIC REGION'
                                  326+            PUNCH '//                 PARM=''00103005001024905210SYSDA    ''   X
                                     +                  DEFAULT PARM FLD'
                                  327+            PUNCH '//*                       BPPTTTOOOMMMIIICCCRLSSSSSSSS'
                                  328+            PUNCH '//*'
                                  329+            PUNCH '//*                B          DEFINED PROGRAMMER NAME && ACCX
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                               F01OCT71  10/10/72
                                    +              T NBR NOT NEEDED'
                                 330+          PUNCH '//*              PP          PRIORITY=01'
                                 331+          PUNCH '//*              TTT         JOB STEP INTERVAL=30 MINUTES'
                                 332+          PUNCH '//*              OOO         PRIMARY SYSOUT SPACE=50 TRACKSX
                                    +                '
                                 333+          PUNCH '//*              MMM         SECONDARY SYSOUT SPACE=10 TRACX
                                    +                KS'
                                 334+          PUNCH '//*              III         READER/INTERPRETER DISPATCHINGX
                                    +                 PRIORITY=249'
                                 335+          PUNCH '//*              CCC         JOB STEP DEFAULT REGION=52K'
                                 336+          PUNCH '//*              R           DISPLAY && EXECUTE COMMANDS=1'
                                 337+          PUNCH '//*              L           BYPASS LABEL=0'
                                 338+          PUNCH '//*              SSSSSSSS  SYSOUT UNIT NAME=SYSDA'
                                 339+          PUNCH '//*'
                                 340+          PUNCH '//IEFRDER DD DSN=IMS2.PROCLIB(&&MBR),DISP=SHR,DCB=BUFNOX
                                    +                =1'
                                 341+          PUNCH '//IEFPDSI DD    DSN=IMS2.PROCLIB,DISP=SHR      PROCEDURE X
                                    +                LIBRARY'
                                 342+          PUNCH '//          DD    DSN=SYS1.PROCLIB,DISP=SHR'
                                 343+          PUNCH '//IEFDATA DD    UNIT=SYSDA,                            SPOOLX
                                    +                 DEVICE'
                                 344+          PUNCH '//                SPACE=(80,(500,500),RLSE,CONTIG),  AMOUNX
                                    +                T'
                                 345+          PUNCH '//                DCB=(BUFNO=2,LRECL=80,BLKSIZE=80,RECFM=FX
                                    +                B,BUFL=80)'
                                 346+          PUNCH './        ENDUP'
                                 347+          PUNCH '/*'
                                 348+          PUNCH '//STEP3 EXEC PGM=IEUASM,REGION=96K,'
                                 349+          PUNCH '// PARM=''LOAD,NODECK'''
                                 350+          PUNCH '//SYSLIB DD DSN=IMS2.GENLIB,DISP=SHR'
                                 351+          PUNCH '//         DD DSN=SYS1.MACLIB,DISP=SHR'
                                 352+          PUNCH '//SYSGO DD DSN=IMS2.OBJDSET(DFSBBLK0),DISP=OLD'
                                 353+          PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBM,LRECL=121,BLKSIZEX
                                    +                =605), '
                                 354+          PUNCH '// SPACE=(605,(100,50),RLSE,,ROUND)'
                                 355+          PUNCH '//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(400,X
                                    +                100))' 3401+4860
                                 356+          PUNCH '//SYSUT2 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(400,X
                                    +                100))' 3401+4860
                                 357+          PUNCH '//SYSUT3 DD UNIT=(SYSDA,SEP=(SYSLIB,SYSUT1,SYSUT2)),DISX
                                    +                P=(,DELETE),'
                                 358+          PUNCH '// SPACE=(1700,(400,100)) ' 3401+4860
                                 359+          PUNCH '//SYSIN DD *'
                                 360+          PUNCH '         PRINT OFF'
                                 361+          PUNCH '         DFSPSBD DUMMY,2018,0'
                                 362+          PUNCH '         SPACE 3'
                                 363+          PUNCH '         SPACE 3'
                                 364+          PUNCH '       IMSBATCH CENDA=Z9Z8,SVCNO=(253,254,255)'
                                 365+          PUNCH '       DFSIPST REGIONS=1'
                                 366+          PUNCH '         TITLE ''DFSIOS40 - OSAM IOB.'''
                                 367+          PUNCH '       DFSIOIOB NUMIOB=1'
                                 368+          PUNCH '       DFSAVARA 1,SECTYPE=CSECT'
                                 369+          PUNCH '         TITLE ''DFSIWAIT - BATCH IWAIT ROUTINE.'''
                                 370+          PUNCH 'DFSIWAIT CSECT'
                                 371+          PUNCH '****'
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  F01OCT71  10/10/72
                                  372+         PUNCH '*'
                                  373+         PUNCH '*          IWAIT ROUTINE'
                                  374+         PUNCH '*'
                                  375+         PUNCH '*          THIS CODE PUTS THE BATCH TASK'
                                  376+         PUNCH '*          INTO OS WAIT FOR THE DURATION'
                                  377+         PUNCH '*          OF AN I/O OPERATION.'
                                  378+         PUNCH '*'
                                  379+         PUNCH '*          ASSUMES UPON ENTRY:'
                                  380+         PUNCH '*          '
                                  381+         PUNCH '*                R1   PST ADDR'
                                  382+         PUNCH '*                R13  SAVE AREA ADDR'
                                  383+         PUNCH '*'
                                  384+         PUNCH '****'
                                  385+         PUNCH '           SPACE 3'
                                  386+         PUNCH '           SAVE (14,12),,DFSIWAIT'
                                  387+         PUNCH '           L     13,8(,13)'
                                  388+         PUNCH '           WAIT  ECB=(1)'
                                  389+         PUNCH '           L     13,4(,13)'
                                  390+         PUNCH '           RETURN (14,12)'
                                  391+         PUNCH '           TITLE ''DFSISCD - SYSTEM CONTENTS DIRECTORY (SX
                                     +               CD).'''
                                  392+         PUNCH '       ISCD      SECTYPE=CSECT'
                                  393+         PUNCH '           ORG   SCDDBFPL'
                                  394+         PUNCH '           DC    F''7168''        DEFAULT BUFFER POOL SIZEX
                                     +               .'
                                  395+         PUNCH '           ORG'
                                  396+         PUNCH '        END'
                                  397+         PUNCH '/*'
                                  398+         PUNCH '//STEP4 EXEC PGM=IEUASM,REGION=96K,'
                                  399+         PUNCH '// PARM=''LOAD,NODECK'''
                                  400+         PUNCH '//SYSLIB DD DSN=IMS2.GENLIB,DISP=SHR'
                                  401+         PUNCH '//        DD DSN=SYS1.MACLIB,DISP=SHR'
                                  402+         PUNCH '//SYSGO DD DSN=IMS2.OBJDSET(DFSVC000),DISP=OLD'
                                  403+         PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBM,LRECL=121,BLKSIZEX
                                     +               =605), '
                                  404+         PUNCH '// SPACE=(605,(100,50),RLSE,,ROUND)'
                                  405+         PUNCH '//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(400,X
                                     +               100))' 3401+4860
                                  406+         PUNCH '//SYSUT2 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(1700,(400,X
                                     +               100))' 3401+4860
                                  407+         PUNCH '//SYSUT3 DD UNIT=(SYSDA,SEP=(SYSLIB,SYSUT1,SYSUT2)),DISX
                                     +               P=(,DELETE),'
                                  408+         PUNCH '// SPACE=(1700,(400,100)) ' 3401+4860
                                  409+         PUNCH '//SYSIN DD *'
                                  410+         PUNCH '           PRINT OFF'
                                  411+         PUNCH 'DFSVC000 CSECT'
                                  412+         PUNCH 'DFSVCS   SSCD  SECTYPE=CSECT,SVC=(253,254,,255),OCEP=Z9X
                                     +               Z8,OSPS=10, CONTINUE'
                                  413+         PUNCH '                  ROUTCDE=,DESC='
                                  414+         PUNCH '          END'
                                  415+         PUNCH '/*'
                                  416+         PUNCH '//IMSGEN2 JOB MSGLEVEL=1,MSGCLASS=A,CLASS=A'
                                  417+         PUNCH '//STEP1 EXEC PGM=IEWL,'
                                  418+         PUNCH '// REGION=130K,'
                                  419+         PUNCH '// PARM=''RENT,REFR,NCAL,LET,,'''
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                      F01OCT71  10/10/72

                                  420+          PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZEX
                                    +                 =605),'
                                  421+          PUNCH '// SPACE=(605,(10,10),RLSE,,ROUND)'
                                  422+          PUNCH '//SYSOBJ DD DSN=IMS2.OBJDSET,DISP=SHR'
                                  423+          PUNCH '//LOAD DD DSN=IMS2.LOAD,DISP=SHR'
                                  424+          PUNCH '//USERLIB DD DSN=IMS2.RESLIB,DISP=SHR'
                                  425+          PUNCH '//SYSLMOD DD DSN=IMS2.RESLIB,DISP=OLD'
                                  426+          PUNCH '//SYSUT1 DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSOBJ)),SPACE=(CYX
                                    +                 L,(10,1))'
                                  427+          PUNCH '//SYSLIN DD *'
                                  428+          PUNCH '  CHANGE DFSVC400(IGC0025E)'
                                  429+          PUNCH '  INCLUDE LOAD(DFSVC400)'
                                  430+          PUNCH '  NAME IGC0025E(R)            TYPE 4 SVC(LOAD 0)'
                                  431+          PUNCH '  CHANGE DFSVC410(IGC0125E)'
                                  432+          PUNCH '  INCLUDE LOAD(DFSVC410)'
                                  433+          PUNCH '  NAME IGC0125E(R)            TYPE 4 SVC(LOAD 1)'
                                  434+          PUNCH '  CHANGE DFSVC420(IGC0225E)'
                                  435+          PUNCH '  INCLUDE LOAD(DFSVC420)'
                                  436+          PUNCH '  NAME IGC0225E(R)            TYPE 4 SVC(LOAD 2)'
                                  437+          PUNCH '  CHANGE DFSVC430(IGC0325E)'
                                  438+          PUNCH '  INCLUDE LOAD(DFSVC430)'
                                  439+          PUNCH '  NAME IGC0325E(R)            TYPE 4 SVC(LOAD 3)'
                                  440+          PUNCH '  CHANGE DFSAOSA0(IGC0425E)'
                                  441+          PUNCH '  INCLUDE LOAD(DFSAOSA0)'
                                  442+          PUNCH '  NAME IGC0425E(R)            TYPE 4 SVC(LOAD 4)'
                                  443+          PUNCH '  CHANGE DFSAOSB0(IGC0525E)'
                                  444+          PUNCH '  INCLUDE LOAD(DFSAOSB0)'
                                  445+          PUNCH '  NAME IGC0525E(R)            TYPE 4 SVC(LOAD 5)'
                                  446+          PUNCH '  CHANGE DFSAOSC0(IGC0625E)'
                                  447+          PUNCH '  INCLUDE LOAD(DFSAOSC0)'
                                  448+          PUNCH '  NAME IGC0625E(R)            TYPE 4 SVC(LOAD 6)'
                                  449+          PUNCH '  CHANGE DFSAOSD0(IGC0725E)'
                                  450+          PUNCH '  INCLUDE LOAD(DFSAOSD0)'
                                  451+          PUNCH '  NAME IGC0725E(R)            TYPE 4 SVC(LOAD 7)'
                                  452+          PUNCH '  CHANGE DFSICSC0(IGC1025E)'
                                  453+          PUNCH '  INCLUDE LOAD(DFSICSC0)'
                                  454+          PUNCH '  NAME IGC1025E(R)            TYPE 4 SVC(LOAD 10)'
                                  455+          PUNCH '  CHANGE DFSVCI00(IGC1125E)'
                                  456+          PUNCH '  INCLUDE LOAD(DFSVCI00)'
                                  457+          PUNCH '  NAME IGC1125E(R)            TYPE 4 SVC(LOAD 11)'
                                  458+          PUNCH '  CHANGE DFSVC440(IGC1325E)'
                                  459+          PUNCH '  INCLUDE LOAD(DFSVC440)'
                                  460+          PUNCH '  NAME IGC1325E(R)            TYPE 4 SVC(LOAD 13)'
                                  461+          PUNCH '  CHANGE DFSAOCE0(IGG019Z9)'
                                  462+          PUNCH '  INCLUDE LOAD(DFSAOCE0)'
                                  463+          PUNCH '  NAME IGG019Z9(R)          OSAM CHAN. END APPENDAGE'
                                  464+          PUNCH '  INCLUDE LOAD(DFSAOS10)'
                                  465+          PUNCH '  NAME DFSAOS10(R)             OSAM OPEN INTERFACE'
                                  466+          PUNCH '  INCLUDE LOAD(DFSAOS20)'
                                  467+          PUNCH '  NAME DFSAOS20(R)             OSAM READ/WRITE'
                                  468+          PUNCH '  INCLUDE LOAD(DFSAOS30)'
                                  469+          PUNCH '  NAME DFSAOS30(R)             OSAM CHECK'
                                  470+          PUNCH '  INCLUDE LOAD(DFSAOS50)'
                                  471+          PUNCH '  NAME DFSAOS50(R)             OSAM COMMON SUBROUTINES'
                                  472+          PUNCH '  INCLUDE LOAD(DFSDISM0)'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F01OCT71  10/10/72
                                 473+          PUNCH '  NAME DFSDISM0(R)            DL/I ISAM SIMULATOR'
                                 474+          PUNCH '  INCLUDE LOAD(DFSDLA00)'
                                 475+          PUNCH '  INCLUDE LOAD(DFSDLA30)'
                                 476+          PUNCH '  ENTRY DFSSTART'
                                 477+          PUNCH '  NAME DFSDLA00(R)            DL/I ANALYZER'
                                 478+          PUNCH '  INCLUDE LOAD(DFSDLD00)'
                                 479+          PUNCH '  NAME DFSDLD00(R)            DL/I DELETE/REPLACE'
                                 480+          PUNCH '  INCLUDE LOAD(DFSDLOC0)'
                                 481+          PUNCH '  NAME DFSDLOC0(R)            DL/I OPEN/CLOSE'
                                 482+          PUNCH '  INCLUDE LOAD(DFSDLR00)'
                                 483+          PUNCH '  NAME DFSDLR00(R)            DL/I RETRIEVE'
                                 484+          PUNCH '  INCLUDE LOAD(DFSDDLE0)'
                                 485+          PUNCH '  ALIAS DFSDDLI0'
                                 486+          PUNCH '  NAME DFSDDLE0(R)            DL/I INSERT'
                                 487+          PUNCH '  INCLUDE LOAD(DFSDXMT0)'
                                 488+          PUNCH '  NAME DFSDXMT0(R)            DL/I INDEX MAINTENANCE'
                                 489+          PUNCH '  INCLUDE LOAD(DFSDDBH0)'
                                 490+          PUNCH '  INCLUDE LOAD(DFSDCSP0)'
                                 491+          PUNCH '  INCLUDE LOAD(DFSGNSR0)'
                                 492+          PUNCH '  INCLUDE LOAD(DFSFNDS0)'
                                 493+          PUNCH '  INCLUDE LOAD(DFSRCH00)'
                                 494+          PUNCH '  INCLUDE LOAD(DFSBFWR0)'
                                 495+          PUNCH '  INCLUDE LOAD(DFSDCVT0)'
                                 496+          PUNCH '  INCLUDE LOAD(DFSDBHR0)'
                                 497+          PUNCH '  NAME DFSDBH00(R)            DL/I BUFFER HANDLER'
                                 498+          PUNCH '  INCLUDE LOAD(DFSGGSP0) '
                                 499+          PUNCH '  INCLUDE LOAD(DFSLLCL0) '
                                 500+          PUNCH '  INCLUDE LOAD(DFSMMLC0) '
                                 501+          PUNCH '  INCLUDE LOAD(DFSMMUD0) '
                                 502+          PUNCH '  INCLUDE LOAD(DFSRRHM0) '
                                 503+          PUNCH '  INCLUDE LOAD(DFSRRHP0) '
                                 504+          PUNCH '  INCLUDE LOAD(DFSDHD00)'
                                 505+          PUNCH '  INCLUDE LOAD(DFSRCHB0)'
                                 506+          PUNCH '  INCLUDE LOAD(DFSFRSP0)'
                                 507+          PUNCH '  ENTRY DFSDHDS0 '
                                 508+          PUNCH '  NAME DFSDHDS0(R)            DL/I SPACE MANAGEMENT'
                                 509+          PUNCH '  INCLUDE LOAD(DFSRRC00)'
                                 510+          PUNCH '  NAME DFSRRC00(R)            REGION CONTROLLOR'
                                 511+          PUNCH '  INCLUDE LOAD(DFSPCC30)'
                                 512+          PUNCH '  ENTRY PCSTART'
                                 513+          PUNCH '  NAME DFSPCC30(R)            BATCH PROGRAM CONTROLLER'
                                 514+          PUNCH '  INCLUDE LOAD(DFSPR000)'
                                 515+          PUNCH '  ALIAS DFSIPRXX'
                                 516+          PUNCH '  NAME DFSPR000(R)            PROG. REQUEST HANDLER'
                                 517+          PUNCH '  INCLUDE LOAD(DFSRRA00)'
                                 518+          PUNCH '  NAME DFSRRA00(R)            REGION PARM ANALYZER'
                                 519+          PUNCH '  INCLUDE LOAD(DFSRRA10)'
                                 520+          PUNCH '  NAME DFSRRA10(R)'
                                 521+          PUNCH '  INCLUDE LOAD(DFSRRA20)'
                                 522+          PUNCH '  NAME DFSRRA20(R)'
                                 523+          PUNCH '  INCLUDE LOAD(DFSRRA50)'
                                 524+          PUNCH '  NAME DFSRRA50(R)'
                                 525+          PUNCH '  INCLUDE LOAD(DFSRDBL0)'
                                 526+          PUNCH '  NAME DFSRDBL0(R)            DATABASE LOGGER'
                                 527+          PUNCH '  INCLUDE LOAD(DFSILNK0)'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                         F01OCT71  10/10/72
                                 528+         PUNCH '  NAME DFSILNK0(R)            LINKAGE EDITOR INTERFACE'
                                 529+         PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                 530+         PUNCH '  NAME DFSVC000(R)            SECONDARY SCD MODULE'
                                 531+         PUNCH '/*'
                                 532+         PUNCH '//STEP2 EXEC PGM=IEWL,'
                                 533+         PUNCH '// REGION=130K,'
                                 534+         PUNCH '// PARM=''NCAL,LET,REUS,,'''
                                 535+         PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZEX
                                    +               =605),'
                                 536+         PUNCH '// SPACE=(605,(10,10),RLSE,,ROUND)'
                                 537+         PUNCH '//SYSOBJ DD DSN=IMS2.OBJDSET,DISP=SHR'
                                 538+         PUNCH '//LOAD DD DSN=IMS2.LOAD,DISP=SHR'
                                 539+         PUNCH '//USERLIB DD DSN=IMS2.RESLIB,DISP=SHR'
                                 540+         PUNCH '//SYSLMOD DD DSN=IMS2.RESLIB,DISP=OLD'
                                 541+         PUNCH '//SYSUT1 DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSOBJ)),SPACE=(CYX
                                    +               L,(10,1))'
                                 542+         PUNCH '//SYSLIN DD *'
                                 543+         PUNCH '  INCLUDE SYSOBJ(DFSBBLK0)          BATCH CONTROL BLOCKS.X
                                    +               '
                                 544+         PUNCH '  INCLUDE LOAD(DFSBINT0)' BATCH INTERFACE SIMULATOR'
                                 545+         PUNCH '  INCLUDE LOAD(DFSBLOI0)            BATCH LOG RECORDER'
                                 546+         PUNCH '  INCLUDE LOAD(DFSFLOG0)            LOG WRITER'
                                 547+         PUNCH '  ENTRY DFSBNUC'
                                 548+         PUNCH '  NAME DFSBNUC0(R)            BATCH NUCLEUS'
                                 549+         PUNCH '  INCLUDE LOAD(DFSIST00)'
                                 550+         PUNCH '  INCLUDE LOAD(DFSISTU0)'
                                 551+         PUNCH '  ENTRY DFSIST00'
                                 552+         PUNCH '  NAME DFSIST00(R)            STATISTICS 0'
                                 553+         PUNCH '  INCLUDE LOAD(DFSLI000)'
                                 554+         PUNCH '  ALIAS PLITDLI'
                                 555+         PUNCH '  ALIAS DFSPLI'
                                 556+         PUNCH '  ALIAS CBLTDLI'
                                 557+         PUNCH '  ALIAS DFSCOBOL'
                                 558+         PUNCH '  ALIAS ASMTDLI'
                                 559+         PUNCH '  ALIAS DFSASM'
                                 560+         PUNCH '  ALIAS FORTDLI'
                                 561+         PUNCH '  ALIAS DFSFOR'
                                 562+         PUNCH '  NAME DFSLI000(R)            DL/I LANGUAGE INTERFACE'
                                 563+         PUNCH '  INCLUDE LOAD(DFSDLBL0)'
                                 564+         PUNCH '  INCLUDE LOAD(DFSLI000)'
                                 565+         PUNCH '  NAME DFSDLBL0(R)            DL/I BLOCK BUILDER'
                                 566+         PUNCH '  INCLUDE LOAD(DFSPRRC0)'
                                 567+         PUNCH '  NAME DFSPRRC0(R)'
                                 568+         PUNCH '  INCLUDE LOAD(DFSPRPX0)'
                                 569+         PUNCH '  NAME DFSPRPX0(R)'
                                 570+         PUNCH '  INCLUDE LOAD(DFSPRPL0)'
                                 571+         PUNCH '  NAME DFSPRPL0(R)'
                                 572+         PUNCH '  INCLUDE LOAD(DFSPRRG0)'
                                 573+         PUNCH '  NAME DFSPRRG0(R)'
                                 574+         PUNCH '  INCLUDE LOAD(DFSFLOS0)'
                                 575+         PUNCH '  NAME DFSFLOS0(R)            STAE EXIT ROUTINE'
                                 576+         PUNCH '  INCLUDE LOAD(DFSDSEH0)'
                                 577+         PUNCH '  NAME DFSDSEH0(R)           PREFIX RESOLUTION ROUTINE'
                                 578+         PUNCH '/*'
                                 579+         PUNCH '//IMSGEN3 JOB MSGLEVEL=1,MSGCLASS=A,CLASS=A'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                     F01OCT71  10/10/72
                                 580+          PUNCH '//STEP1 EXEC PGM=IEWL,'
                                 581+          PUNCH '// REGION=130K,'
                                 582+          PUNCH '// PARM=''OVLY,NCAL,LET,,'''
                                 583+          PUNCH '//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZEX
                                   +                =605),'
                                 584+          PUNCH '// SPACE=(605,(10,10),RLSE,,ROUND)'
                                 585+          PUNCH '//SYSOBJ DD DSN=IMS2.OBJDSET,DISP=SHR'
                                 586+          PUNCH '//LOAD DD DSN=IMS2.LOAD,DISP=SHR'
                                 587+          PUNCH '//USERLIB DD DSN=IMS2.RESLIB,DISP=SHR'
                                 588+          PUNCH '//SYSLMOD DD DSN=IMS2.RESLIB,DISP=OLD'
                                 589+          PUNCH '//SYSUT1 DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSOBJ)),SPACE=(CYX
                                   +                L,(10,1))'
                                 590+          PUNCH '//SYSLIN DD *'
                                 591+          PUNCH '  INCLUDE LOAD(DFSFSWA0)' 2589
                                 592+          PUNCH '  NAME DFSFSWA0(R)              STAE WKAREA MODULE' 2589
                                 593+          PUNCH '  INCLUDE LOAD(DFSDLN00)'
                                 594+          PUNCH '  INCLUDE LOAD(DFSIINL0)'
                                 595+          PUNCH '  INCLUDE LOAD(DFSIIN10)'
                                 596+          PUNCH '  INCLUDE LOAD(DFSIIN20)'
                                 597+          PUNCH '  INCLUDE LOAD(DFSDBPI0)'
                                 598+          PUNCH '  NAME DFSDLBN0(R)              DL/I INITIALIZATION'
                                 599+          PUNCH '  INCLUDE LOAD(DFSDPSB0)'
                                 600+          PUNCH '  NAME DFSDPSB0(R)              DL/I UTILITY INTERFACE,  PX
                                   +                SB GENERATOR'
                                 601+          PUNCH '  INCLUDE LOAD(DFSDAPL0)'
                                 602+          PUNCH '  NAME DFSDAPL0(R)              DL/I UTILITY INTERFACE,  AX
                                   +                UTH PGM LIST'
                                 603+          PUNCH '  INCLUDE LOAD(DFSUCMN0)'
                                 604+          PUNCH '  INCLUDE LOAD(DFSUCCT0)'
                                 605+          PUNCH '  INCLUDE LOAD(DFSUC150)'
                                 606+          PUNCH '  INCLUDE LOAD(DFSUC350)'
                                 607+          PUNCH '  INCLUDE LOAD(DFSUCER0)'
                                 608+          PUNCH '  INCLUDE LOAD(DFSCUMM0)'
                                 609+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                 610+          PUNCH '  NAME DFSUCUM0(R)'
                                 611+          PUNCH '  INCLUDE LOAD(DFSUDMP0)'
                                 612+          PUNCH '  INCLUDE LOAD(DFSDMPM0)'
                                 613+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                 614+          PUNCH '  NAME DFSUDMP0(R)              IMAGE DUMP'
                                 615+          PUNCH '  INCLUDE LOAD(DFSUACB0)'
                                 616+          PUNCH '  INCLUDE LOAD(DFSUSCH0)'
                                 617+          PUNCH '  INCLUDE LOAD(DFSUMGT0)'
                                 618+          PUNCH '  INCLUDE LOAD(DFSUMSG0)'
                                 619+          PUNCH '  INCLUDE LOAD(DFSUSRC0)'
                                 620+          PUNCH '  INCLUDE LOAD(DFSUAMB0)'
                                 621+          PUNCH '  ENTRY DFSUACB0'
                                 622+          PUNCH '  NAME DFSUACB0(R)              ACBLIB UTILITY '
                                 623+          PUNCH '  INCLUDE LOAD(DFSLBLM0) '
                                 624+          PUNCH '  NAME DFSLBLM0(R)        BLOCK BUILDER MSG ROUTER '
                                 625+          PUNCH '  INCLUDE LOAD(DFSURUL0)'
                                 626+          PUNCH '  INCLUDE LOAD(DFSAOS10)'
                                 627+          PUNCH '  INCLUDE LOAD(DFSRULM0)'
                                 628+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                 629+          PUNCH '  NAME DFSURUL0(R)              HISAM REORG UNLOAD'
                                 630+          PUNCH '  INCLUDE LOAD(DFSURRL0)'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71  10/10/72

                                 631+          PUNCH '  INCLUDE LOAD(DFSRRLM0)'
                                 632+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                 633+          PUNCH '  NAME DFSURRL0(R)          HISAM REORG RE-LOAD'
                                 634+          PUNCH '  INCLUDE LOAD(DFSURDB0)'
                                 635+          PUNCH '  INCLUDE LOAD(DFSRDBM0)'
                                 636+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                 637+          PUNCH '  NAME DFSURDB0(R)          DATABASE RECOVERY'
                                 638+          PUNCH '  INCLUDE LOAD(DFSBIND0)'
                                 639+          PUNCH '  INCLUDE LOAD(DFSDBLM0)'
                                 640+          PUNCH '  INCLUDE LOAD(DFSDBLD0)'
                                 641+          PUNCH '  INCLUDE LOAD(DFSDBLP0)'
                                 642+          PUNCH '  INCLUDE LOAD(DFSDBLR0)'
                                 643+          PUNCH '  ENTRY DFSBIND0'
                                 644+          PUNCH '  NAME  DFSBBLD0(R)         BATCH ACB BLOCKS LOADER'
                                 645+          PUNCH '  INCLUDE LOAD(DFSURGU0)'
                                 646+          PUNCH '  INCLUDE LOAD(DFSLI000)'
                                 647+          PUNCH '  INCLUDE LOAD(DFSRGUM0)'
                                 648+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                 649+          PUNCH '  NAME DFSURGU0(R)          HDAM REORG UNLOAD'
                                 650+          PUNCH '  INCLUDE LOAD(DFSURGL0)'
                                 651+          PUNCH '  INCLUDE LOAD(DFSUEX10)'
                                 652+          PUNCH '  INCLUDE LOAD(DFSLI000)'
                                 653+          PUNCH '  INCLUDE LOAD(DFSRGLM0)'
                                 654+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                 655+          PUNCH '  NAME DFSURGL0(R)          HDAM REORG RE-LOAD'
                                 656+          PUNCH '  INCLUDE LOAD(DFSURGP0)'
                                 657+          PUNCH '  INCLUDE LOAD(DFSLI000)'
                                 658+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                 659+          PUNCH '  INCLUDE LOAD(DFSURGM0)'
                                 660+          PUNCH '  NAME DFSURGP0(R)          REORG PREFIX UPDATE'
                                 661+          PUNCH '  INCLUDE LOAD(DFSURG10)'
                                 662+          PUNCH '  INCLUDE LOAD(DFSLI000)'
                                 663+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                 664+          PUNCH '  INCLUDE LOAD(DFSURGM0)'
                                 665+          PUNCH '  NAME DFSURG10(R)          REORG SORT 1'
                                 666+          PUNCH '  INCLUDE LOAD(DFSURGS0)'
                                 667+          PUNCH '  INCLUDE LOAD(DFSDSEH0)'
                                 668+          PUNCH '  INCLUDE LOAD(DFSLI000)'
                                 669+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                 670+          PUNCH '  INCLUDE LOAD(DFSURGM0)'
                                 671+          PUNCH '  NAME DFSURGS0(R)          REORG SCAN'
                                 672+          PUNCH '  INCLUDE LOAD(DFSURPR0)'
                                 673+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                 674+          PUNCH '  INCLUDE LOAD(DFSURGM0)'
                                 675+          PUNCH '  NAME DFSURPR0(R)          PRE-REORG/LOAD'
                                 676+          PUNCH '  INCLUDE LOAD(DFSERA10)'
                                 677+          PUNCH '  NAME DFSERA10(R)          LOG PRINT SERVICE AID.
                                 678+          PUNCH '  INCLUDE LOAD(DFSFLBD0)' V954
                                 679+          PUNCH '  ENTRY DFSFLBD1'
                                 680+          PUNCH '  NAME DFSFLBD0(R)          STAND-ALONE LOG TERMINATORX
                                     +                '
                                 681+          PUNCH '  INCLUDE LOAD(DFSFLOT0)'
                                 682+          PUNCH '  INCLUDE LOAD(DFSFLMT0)'
                                 683+          PUNCH '  INCLUDE SYSOBJ(DFSVC000)'
                                 684+          PUNCH '  ENTRY DFSFLOT0'
```



```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71  10/10/72

                                 685+          PUNCH '  NAME DFSFLOT0(R)          SYSTEM LOG TERMINATOR '
                                 686+          PUNCH '  INCLUDE LOAD(DFSIST10)'
                                 687+          PUNCH '  NAME DFSIST10(R)          STATISTICS 1'
                                 688+          PUNCH '  INCLUDE LOAD(DFSIST20)'
                                 689+          PUNCH '  NAME DFSIST20(R)          STATISTICS 2'
                                 690+          PUNCH '  INCLUDE LOAD(DFSIST30)'
                                 691+          PUNCH '  NAME DFSIST30(R)          STATISTICS 3'
                                 692+          PUNCH '  INCLUDE LOAD(DFSIST40)'
                                 693+          PUNCH '  NAME DFSIST40(R)          STATISTICS 4'
                                 694+          PUNCH '  INCLUDE LOAD(DFSDDLS0)'
                                 695+          PUNCH '  INCLUDE LOAD(DFSLI000)'
                                 696+          PUNCH '  ENTRY CLITCBL'
                                 697+          PUNCH '  NAME DFSDDLT0(R)           DL/I TEST PROGRAM'
                                 698+          PUNCH '  INCLUDE LOAD(DFSBACK0)'
                                 699+          PUNCH '  INCLUDE LOAD(DFSRDBC0)'
                                 700+          PUNCH '  INCLUDE LOAD(DFSLI000)'
                                 701+          PUNCH '  INCLUDE LOAD(DFSBACM0)'
                                 702+          PUNCH '  ENTRY DLITCBL'
                                 703+          PUNCH '  NAME DFSBBO00(R)          BATCH DATABASE BACKOUT'
                                 704+          PUNCH '//'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                              F01OCT71  10/10/72
                                              *,*****************************************************
                                              *,*****************************************************
                                              *,**                                                 **
                                              *,**        SUCCESSFUL IMS SYSTEM DEFINITION         **
                                              *,**                FOR BATCH FUNCTION.              **
                                              *,
                                              *,     ---  BEFORE STAGE II EXECUTION  ---
                                              *,
                                              *,       THE FOLLOWING IMS DATA SETS MUST BE ALLO-
                                              *,   CATED AND CATALOGED BEFORE ATTEMPTING TO EXECUTE
                                              *,   THE GENERATED STAGE II JOB STREAM:
                                              *,       IMS2.LOAD      IMS2.GENLIB   IMS2.RESLIB
                                              *,       IMS2.MACLIB    IMS2.PROCLIB
                                              *,       IMS2.OBJDSET   IMS2.RESLIB
                                              *,
                                              *,       OTHER REQUIRED DATA SETS MUST BE ALLOCATED
                                              *,   AND CATALOGED PRIOR TO EXECUTION OF THE IMS
                                              *,   CONTROL PROGRAM.
                                              *,
                                              *,
                                              *,      ----  BEFORE SYSTEM EXECUTION  ----
                                              *,
                                              *,       THE FOLLOWING FUNCTIONS MUST BE COMPLETED
                                              *,   BEFORE SUCCESSFUL EXECUTION OF THE IMS CONTROL
                                              *,   PROGRAM CAN BE ACHIEVED:
                                              *,
                                              *,     THE REQUIRED TYPE 4 SVC AND APPENDAGE MODULES
                                              *,     MUST BE PLACED IN SYS1.SVCLIB.
                                              *,
                                              *,     THE LOAD MEMBER DFSRRC00 MUST BE ADDED TO THE
                                              *,     'PRIVILIGED PROGRAM NAME LIST' MEMBER (IEEVLNKT)
                                              *,     IN SYS1.LINKLIB.
                                              *,
                                              *,     THE IMS2.RESLIB DATASET MUST BE CONCATENATED WITH
                                              *,     SYS1.LINKLIB BY ADDING THE NAME 'IMS2.RESLIB' TO
                                              *,     THE LNKLST00 MEMBER OF SYS1.PARMLIB.
                                              *,
                                              *,     THE CATALOGED PROCEDURE 'IMSRDR' MUST BE PLACED
                                              *,     IN SYS1.PROCLIB BEFORE ANY OTHER IMS GENERATED
                                              *,     PROCEDURE CAN BE SUCCESSFULLY EXECUTED.
                                              *,
                                              *,         REFER TO THE IMS SYSTEM PROGRAMMING
                                              *,     REFERENCE MANUAL FOR INSTRUCTIONS AND ASSIST-
                                              *,     ANCE IN PERFORMING THESE AND OTHER REQUIRED
                                              *,     FUNCTIONS BEFORE SYSTEM EXECUTION.
                                              *,**                                                 **
                                              *,**                                                 **
                                              *,*****************************************************
                                              *,*****************************************************
                                         END

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
*STATISTICS*    SOURCE RECORDS (SYSIN) =      3       SOURCE RECORDS (SYSLIB) =  5679
*OPTIONS IN EFFECT*   LIST, NODECK, NOLOAD, NORENT, XREF, NOTEST, ALGN, OS, NOTERM, LINECNT =  55
```

CHAPTER 4. IMS/360 SECURITY MAINTENANCE PROGRAM

SECURITY MAINTENANCE

Although IMS/360 system definition creates the majority of resident control blocks for the IMS/360 control program, it supplies only a minimum set of terminal security to protect basic master terminal commands against unauthorized entry. Full security capabilities are supplied through a security maintenance program which allows the IMS/360 user the flexibility of changing security information without redefining his entire system. Security is provided by terminal and by password.

The reader should be familiar with IMS/360 System Definition to obtain the best use of the following information.

The function of the security maintenance program (SMP) is to create or alter password or terminal protection of an online IMS/360 system. The generated IMS/360 system has only a minimum subset of terminal security to protect DISPLAY, NRESTART, RSTART, CHECKPOINT, ERESTART, START, CHANGE, STOP, PURGE, DBRECOVERY, IDLE, DBDUMP, ASSIGN, DELETE, and PSTOP commands. The security maintenance program creates password and terminal security for transactions and additional commands entered from terminals. It also creates password security on data bases and programs. The control of the security maintenance program is such that the user may view his system in terms of resources to which passwords may have access, or he may view the system as a security profile system, that is, by defining a password which has access to a set of resources. The detailed explanation covers the use of the various control cards to describe either a "profile-oriented" system or a "resource-oriented" system of security maintenance. There is no restriction on the use of both types of description within the same security maintenance program execution.

PASSWORD MAINTENANCE

If password maintenance control cards are presented in the input stream for the SMP, the password maintenance function is performed. Using the SMP password control cards, the following functions are available:

- Add passwords to or delete passwords from the IMS/360 communication password table (CPT)

- Change the password security requirements for transaction codes, terminal command verbs, program status changes, data base status changes, logical or physical terminal status changes

IMS/360 password table and password matrix changes may become effective the next time IMS/360 is restarted. If the next restart is a "cold start", the master terminal operator may specify that no password security be used or that the new password table and matrix be used. If the next restart is a "warm start", the master terminal operator may specify that the current status of the password table and matrix is to be restored using the system checkpoint records, or that the new password table and matrix are to be used.

TERMINAL SECURITY MAINTENANCE

If terminal security maintenance control cards are presented in the input stream for the SMP, maintenance functions are performed upon the IMS/360 communications terminal matrix. Using the SMP terminal security control cards, the following function is available:

- Add to or delete from terminal security requirements for command verbs and application program transaction codes

Terminal security changes become effective the next time IMS/360 is restarted. If the next restart is a "cold start", the master terminal operator may specify that no terminal security be used or that the new terminal matrix is to be used. If the next start is a "warm start", the master terminal operator may specify that the current status of terminal security be restored using system checkpoint records, or that the new terminal security matrix is to be used.

The security maintenance program can not be executed until an IMS/360 system definition has been performed. Input requirements for the SMP include an IMS/360 system description block (SDB) which is created at system definition time and which must reside in the IMS2.RESLIB library with the IMS/360 control program nucleus. If multiple IMS/360 systems exist, the SMP maintains as many as nine sets of security control blocks in the same library. If errors are encountered in processing SMP control cards, no security block update functions are performed. Diagnostic error messages are produced for the entire input stream.

At user option, the SMP performs a no-update run, producing a printed analysis of IMS/360 security requirements. In addition, each execution of the SMP produces a printed analysis of the IMS/360 configuration being maintained.

CONTROL AND DATA STATEMENTS

The security maintenance program control and data statements available are; PASSWORD, TERMINAL, TRANSACT, COMMAND, DATABASE, PROGRAM, and PTERM. In general, each of these cards may be used as required. The specifications which one considers in designing a password security system must be tailored to the particular environment in which IMS/360 is to run. The control cards above are used to describe the security environment which the IMS/360 system is to use in processing messages and commands.

Control statements are identified by)(characters (close and open parentheses in combination) in card columns 1 and 2, followed by a blank in column 3. Data statements are identified by a blank in column 1. A control statement remains in effect until another control statement or end of input data is encountered. Each statement, control or data, has only one allowable operand. Valid combinations of control and data statements are shown in Figure 9.

NAME	OPERATION	OPERAND
)(	PASSWORD	password
	TERMINAL	logical terminal name
	TRANSACT	transaction code

	COMMAND	command language verb
	DATABASE	name
	PROGRAM	name
	PTERM	number
) (	TERMINAL	logical terminal name
	PASSWORD	password
	TRANSACT	transaction code
	COMMAND	command language verb
) (	TRANSACT	transaction code
	PASSWORD	password
	TERMINAL	logical terminal name
) (	COMMAND	command language verb
	PASSWORD	password
	TERMINAL	logical terminal name
) (	DATABASE or	name
) (	PROGRAM or	name
) (	PTERM	name
	PASSWORD	password

where:

password

A password must contain only alphameric characters and may be one through eight characters in length. The longest password statement encountered in the input stream governs the maximum length of input passwords which will be accepted by the user's IMS/360 system. Data statements: TERMINAL, TRANSACT, COMMAND, DATABASE, PROGRAM, and PTERM.

logical terminal name

A valid logical terminal name may be one through eight characters in length. Terminal names which are not defined in the IMS/360 system being maintained are invalid and will be rejected by the security maintenance program.

transaction code

A valid transaction code may be one through eight characters in length and must be defined in the IMS/360 system being maintained. If it is not defined, it is treated as invalid by the security maintenance program.

name

A valid data base name, program name, or physical terminal number is available from Stage 2 output of IMS/360 system definition.

command language verb

Valid command language verbs may be obtained from the Stage 2 output of IMS/360 system definition. The command verb, less leading slash, may be abbreviated to the first three characters.

General Notes: Only the first three characters of the operation code are required to identify control or data statements. Physical terminal numbers may be found in the terminal map printed in the assembly of DFSISDB0 in Stage 2 of IMS/360 system definition.

To define additional passwords, a PASSWORD control statement may be used with no following data statements:

```
)( PASSWORD    ABCD
)( PASSWORD    EFGH
```

DATA CARD TYPE	CONTROL CARD TYPE						
	PASSWORD	TERMINAL	TRANSACT	COMMAND	DATABASE	PROGRAM	PTERM
PASSWORD	NO	YES	YES	YES	YES	YES	YES
TERMINAL	YES	NO	YES	YES	NO	NO	NO
TRANSACT	YES	YES	NO	NO	NO	NO	NO
COMMAND	YES	YES	NO	NO	NO	NO	NO
DATABASE	YES	NO	NO	NO	NO	NO	NO
PROGRAM	YES	NO	NO	NO	NO	NO	NO
PTERM	YES	NO	NO	NO	NO	NO	NO

Figure 9. Security maintenance control and data card types

CONTROL AND DATA STATEMENT COMBINATIONS

The following outlines the use of various control and data statement combinations:

Control <u>Statement</u>	Data <u>Statement</u>	<u>Explanation</u>
1. PASSWORD TERMINAL	TERMINAL PASSWORD	To require a password to be used with the logical terminal name when modifying the status of a logical terminal through a /LOCK, /UNLOCK, or /IAM command
2. PASSWORD TRANSACT	TRANSACT PASSWORD	To require a password to be entered from the input terminal following the transaction code for each message

	Control Statement	Data Statement	Explanation
3.	PASSWORD COMMAND	COMMAND PASSWORD	To require a password to be entered following the command verb when using the terminal command language
4.	PASSWORD DATABASE	DATABASE PASSWORD	To require a password to be entered following the data base name when modifying the status of a data base through a /LOCK or /UNLOCK command
5.	PASSWORD PROGRAM	PROGRAM PASSWORD	To require a password to be entered following the program name when modifying the status of a program (PSB) through a /LOCK or /UNLOCK command
6.	PASSWORD PTERM	PTERM PASSWORD	To require a password to be entered following the key word PTERM when modifying the status of a physical terminal through a /LOCK, /UNLOCK, or /IAM command
7.	TERMINAL TRANSACT	TRANSACT TERMINAL	To restrict use of a transaction code to a specific logical terminal. name. Note: Entry of the named transaction codes will only be permitted from the terminals specified.
8.	COMMAND TERMINAL	TERMINAL COMMAND	To restrict use of a command verb to specific logical terminals

Input statements may be used as control cards or data cards. Using the input statements, security requirements may be expressed as either profile oriented or resource oriented. A profile security system would describe the resources to be secured in terms of the securing element. For example, the following describes a profile for password SAMSMITH.

```
)( PASSWORD SAMSMITH
        TRANSACT    PAYROLL
        TRANSACT    PERS
        COMMAND     LOCK
        COMMAND     UNLOCK
        DATABASE    PAYREC
        PROGRAM     PAYPROG
```

To describe these same security requirements by resource, the following statements would be required.

```
)(      TRANSACT    PAYROLL
        PASSWORD    SAMSMITH
)(      TRANSACT    PERS
        PASSWORD    SAMSMITH
)(      COMMAND     LOCK
        PASSWORD    SAMSMITH
)(      COMMAND     UNLOCK
        PASSWORD    SAMSMITH
)(      DATABASE    PAYREC
        PASSWORD    SAMSMITH
)(      PROGRAM     PAYPROG
        PASSWORD    SAMSMITH
```

As the preceding example illustrates, passwords may be more easily described by using the securing elements as data. Terminal security, however, is more easily described by using the secured element, the transaction, as a control statement, followed by the security elements, the terminals, as data.

```
) ( TRANSACT PAYROLL
      TERMINAL    DEPT40
      TERMINAL    DEPT65
      TERMINAL    VPPERS
) ( TRANSACT PERS
      TERMINAL    DEPT40
```

The reverse or profile example would be:

```
) ( TERMINAL        DEPT40
      TRANSACT    PAYROLL
      TRANSACT    PERS
) ( TERMINAL        DEPT65
      TRANSACT    PAYROLL
) ( TERMINAL        VPPERS
      TRANSACT    PAYROLL
```

The basic online system only provides terminal security for a subset of the command language. The following example would secure a more typical set of commands against entry from any terminal except the Master Terminal:

```
) ( TERMINAL  master terminal name
    COMMAND     START
    COMMAND     STOP
    COMMAND     NRESTART
    COMMAND     CHECKPOINT
    COMMAND     PSTOP
    COMMAND     ERESTART
    COMMAND     DBRECOVERY
    COMMAND     ASSIGN
    COMMAND     BROADCAST
    COMMAND     CHANGE
    COMMAND     DBDUMP
    COMMAND     PURGE
    COMMAND     LOG
    COMMAND     DISPLAY
    COMMAND     DELETE
```

DESCRIPTION OF SMP OUTPUT

The security maintenance program produces three printed reports. The first report is the logical configuration of IMS/360 system being maintained, the second is the password table generated, and the third is the matrix for the security of a particular IMS/360 nucleus.

SECURITY MAINTENANCE PROGRAM EXECUTION

The security maintenance run is a three-step job. The first step accepts the input control and data cards for the security maintenance program and edits them for correct format and validity against the IMS/360 system being maintained. If there are no errors in the first step, the second step, an operating system assembly, is performed. Step three is a link edit which takes the assembly output from step two and creates the communication password table, communications password matrix, and communication terminal matrix load modules used by the IMS/360 control program. Depending upon the input presented, there will be a variable number of output load modules created.

The maximum bounds of the generated matrices, terminal or password, are expressed as:

$$\frac{I * R}{8} = M = < 32767$$

where:

I

is number of securing resources (passwords or logical terminals)

R

is number of unique combinations of secured resources

M

is the total main storage requirement in bytes

The maximum number of entries in the password table is expressed as:

$$I = < 32767$$

where:

I

is the total number of passwords

To perform a security maintenance run, the user must have previously defined an IMS/360 control program using the value ALL, ON-LINE, NUCLEUS, or CTLBLKS as the second sublist entry of the SYSTEM operand of the IMSCTRL macro-instruction. One of the modules created during Stage 2 of IMS/360 system definition is a directory of resources of the defined system which is placed in the IMS2.RESLIB data set. This directory and the security maintenance control cards comprise the input requirements for the security maintenance program (SMP). Output from the SMP consists of four sequential members which are placed in IMS2.RESLIB. These members may not be reprocessed using the linkage editor. The four members contain:

1) Communication Password Table (CPT)
2) Communication Terminal Matrix (CTM)
3) Terminal Offset List (CTL)
4) Password Offset List (CPL)

In addition, the SMP provides a listing of the created maintenance tables. Each run of the SMP replaces previously created members. Of course, ten different sets of security maintenance tables can be concurrently maintained. Each set is associated with one IMS/360 online control program nucleus and gains identity by the last character of the IMS/360 nucleus name. Figure 10 depicts the security maintenance flow.

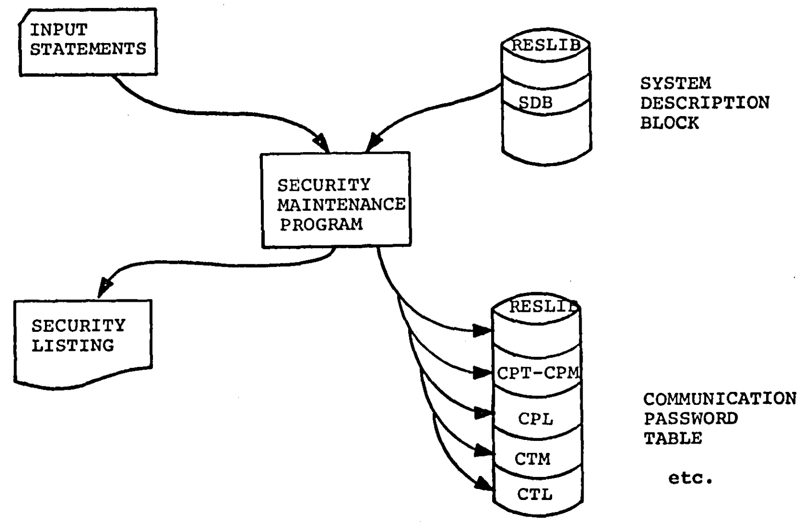


Figure 10. Security maintenance flow

The table following shows the job control statements by step necessary to execute the security maintenance utility.

STATEMENT	USAGE
JOB statement	Initiates security maintenance job
JOBLIB statement	Defines the partitioned data set named IMS2.RESLIB. Contains the members DFSCNUCn and DFSISMPO.
step S	
EXEC statement	Specifies the program name (PGM=DFSISMPO) and must contain a PARM keyword value of the form PARM = 'option,number' option LIST - validity check and list new security tables UPDATE - validity check, list, and update security tables in RESLIB number a value ranging from 0-9 which is the last character of the IMS/360 nucleus member name to be maintained

	where default values for PARM= 'UPDATE,0'
SYSPRINT DD statement	Defines a sequential message data set. The data set can be written to system output devices, magnetic tape, or direct access volumes. The following DCB parameters must be specified. RECFM=VBA LRECL=125 BLKSIZE = 129 or greater
SYSPUNCH DD statement	Defines a sequential output data set which contains assembler statements produced by step S. The data set may be passed to step C. The following DCB parameters must be specified. RECFM=F or FB LRECL=80 BLKSIZE=80 or multiple of 80
SYSLIN DD statement	Defines a sequential output data set which contains linkage editor control statements produced by step S. The data set may be passed to step L. The following DCB parameters must be specified. RECFM=F or FB LRECL=80 BLKSIZE=80 or multiple of 80
SYSUT1 DD statement	Defines a sequential work data set used only during step S. The following DCB parameters must be specified. RECFM=F or FB BLKSIZE=100 or multiple of 100
SYSUT2 DD statement	Defines a sequential work data set used only during step S. The following DCB parameters must be specified. RECFM=F or FB BLDSIZE=100 or multiple of 100
SYSIN DD statement	Defines a sequential data set or a member of a partitioned data set which contains security maintenance input statements. The following DCB parameters must be specified. RECFM=F or FB BLKSIZE=80 or multiple of 80

step C	
EXEC statement	Specifies the program name (PGM=IEUASM) of the assembler. Following parameters must be present. PARM='LOAD,NODECK' COND=(12,LT,S)
SYSPRINT DD statement	Defines a sequential message data set. The data set can be written to system output devices, magnetic tape, or direct access volumes. The following DCB parameters must be specified. RECFM=FM or FBM LRECL = 121 BLKSIZE=121 or multiple of 121
SYSGO DD statement	Defines a sequential temporary data set for object output from the assembler. The data set may be passed to step L.
SYSUT1 SYSUT2 SYSUT3 DD statements	Defines sequential data sets used for work space by the assembler only during step C.
SYSIN DD statement	Defines passed sequential input data set created in step S using DD name SYSPUNCH.
step L	
EXEC statement	Specifies the program name (PGM=IEWL) of the linkage editor. Following parameters must be present. PARM='LIST,NE,OL' COND=(4,LT,S)
SYSPRINT DD statement	Defines a sequential message data set for the linkage editor. The data set can be written to system output devices, magnetic tape, or direct access volumes. The following DCB parameters must be specified:

	RECFM=FA or FBA LRECL=121 BLKSIZE=121 or multiple of 121
SYSLMOD DD statement	Defines output partitioned data set, IMS2.RESLIB, for the linkage editor. Normally the same data set specified for DD name JOBLIB.
INPUT DD statement	Defines passed sequential temporary data set created using DD name SYSGO in step C.
SYSUT1 DD statement	Defines sequential temporary data set used in step L by the linkage editor.
SYSLIN DD statement	Defines passed sequential temporary data set created using DD name SYSLIN in step S.

Once created, these new matrices and the password table are not made available to the IMS/360 online system until a restart is performed. At normal restart time, the operator has the option of incorporating or not incorporating the newly created security tables. At either cold start, that is, NRESTART CHECKPOINT 0, or warm restart, NRESTART any checkpoint number, the new security tables are not included unless specifically requested by the system operator. The two keyword operands of the NRESTART command which are used to request new security are PASSWORD, for password security, and TERMINAL, for terminal security. Once these two keywords are used in a normal restart, the system checkpoint facility causes the new security maintenance to continue through subsequent warm starts. If the user desires, once a normal successful restart using the normal keywords has been accomplished, he may change his system security configuration. Again, these changes will not become effective until the user specifically requests them at normal restart time.

SECURITY MAINTENANCE EXAMPLE

The following is an example of the input cards for the security maintenance program that reflects the system definition example in this chapter. This example assumes:

- a password for each program
- a password for each data base
- a password for each transaction code, except INQUIRY
- a list of terminals that can use each transaction code, along with the required password limiting some IMS/360 terminal commands to the master terminal

* the master terminal can enter all IMS/360 terminal commands and transaction codes defined by the system definition example in this manual

<u>Example</u>

```
)( PROGRAM      ACCT
    PASSWORD    DOLLAR

)( PROGRAM      ENG560
    PASSWORD    PARTNO

)( PROGRAM      LOGREC
    PASSWORD    NONE

)( PROGRAM      AGC0568
    PASSWORD    MONEY

)( DATABASE     ACCTLOG
    PASSWORD    LOG

)( DATABASE     ACCTREC
    PASSWORD    REC

)( DATABASE     ACTIVITY
    PASSWORD    ACTIVE

)( DATABASE     ENGREC
    PASSWORD    PIERSQ

)( DATABASE     PARTSREC
    PASSWORD    PIERSQ

)( DATABASE     PARTSREC
    PASSWORD    ASSY

)( TRANSACT     ACCTCHG
    PASSWORD    CHARGE
    TERMINAL    A875111
    TERMINAL    C8751112
    TERMINAL    D8751113
    TERMINAL    A8751114
    TERMINAL    A8751115

)( TRANSACT     ACTY
    PASSWORD    GO
    TERMINAL    A8751111
    TERMINAL    C8751112
    TERMINAL    D8751113
    TERMINAL    A8751114
    TERMINAL    A8751115

)( TRANSACT     TNL
    PASSWORD    QTY
    TERMINAL    DEPT 650
    TERMINAL    DEPT 610
    TERMINAL    DEPT 620
    TERMINAL    DEPT 631
    TERMINAL    DEPT 632
    TERMINAL    DEPT 630
    TERMINAL    DEPT 640
    TERMINAL    DEPT 641
    TERMINAL    DEPT 642
```

```
) ( TRANSACT     ING
    PASSWORD     QUESTION
    TERMINAL     DEPT310
    TERMINAL     DEPT311
    TERMINAL     DEPT312
    TERMINAL     DEPT410
    TERMINAL     DEPT411
    TERMINAL     DEPT412
    TERMINAL     DEPT510
    TERMINAL     DEPT511
    TERMINAL     DEPT512
    TERMINAL     DEPT100
    TERMINAL     DEPT200
    TERMINAL     DEPT686
    TERMINAL     MASTER
    TERMINAL     ALTMAST
    TERMINAL     MAINT
    TERMINAL     DEPT710
    TERMINAL     DEPT720
    TERMINAL     DEPT848
    TERMINAL     DEPT850
    TERMINAL     DEPT900
    TERMINAL     TEST1
    TERMINAL     TEST2

) ( TRANSACT     INVNTRY
    PASSWORD     SUBASSY
    TERMINAL     DEPT310
    TERMINAL     DEPT311
    TERMINAL     DEPT312
    TERMINAL     DEPT410
    TERMINAL     DEPT411
    TERMINAL     DEPT412
    TERMINAL     DEPT510
    TERMINAL     DEPT511
    TERMINAL     DEPT512
    TERMINAL     DEPT100
    TERMINAL     DEPT200
    TERMINAL     DEPT686
    TERMINAL     MASTER
    TERMINAL     ALTMAST
    TERMINAL     MAINT
    TERMINAL     DEPT710
    TERMINAL     DEPT720
    TERMINAL     DEPT848
    TERMINAL     DEPT850
    TERMINAL     DEPT900
    TERMINAL     TEST1
    TERMINAL     TEST2

) ( TRANSACT     ACCT
    PASSWORD     LEDGER
    TERMINAL     DEPT310
    TERMINAL     DEPT311
    TERMINAL     DEPT312
    TERMINAL     DEPT410
    TERMINAL     DEPT411
    TERMINAL     DEPT412
    TERMINAL     DEPT510
    TERMINAL     DEPT511
    TERMINAL     DEPT512
    TERMINAL     DEPT100
    TERMINAL     DEPT200
    TERMINAL     DEPT686
    TERMINAL     MASTER
```

```
    TERMINAL     ALTMAST
    TERMINAL     MAINT
    TERMINAL     DEPT710
    TERMINAL     DEPT720
    TERMINAL     DEPT848
    TERMINAL     DEPT850
    TERMINAL     DEPT900
    TERMINAL     TEST1
    TERMINAL     TEST2

) ( TERMINAL     MASTER
    TRANSACT     ACCTCHG
    TRANSACT     ACTY
    TRANSACT     TNL
    TRANSACT     INQUIRY
    TRANSACT     INQ
    TRANSACT     ENG
    TRANSACT     ACCT
    COMMAND      BROADCAST
    COMMAND      START
    COMMAND      STOP
    COMMAND      PSTOP
    COMMAND      PURGE
    COMMAND      CHANGE
    COMMAND      DELETE
    COMMAND      ASSIGN
    COMMAND      CHECKPOINT
    COMMAND      CHECKPOINT PURGE
    COMMAND      CHECKPOINT FREEZE
    COMMAND      DBDUMP
    COMMAND      NRESTART
    COMMAND      ERESTART
    COMMAND      DBRECOVERY
    COMMAND      IDLE
    COMMAND      RSTART
    COMMAND      DISPLAY
    COMMAND      CHECKPOINT DUMPQ
```

CHAPTER 5. USER MODIFICATIONS AND EXTENSIONS TO THE CONTROL PROGRAM

This chapter describes the message editing capabilities available to the user of IMS/360. Transaction code edits, message switching, physical terminal (output), and user-defined edit routines are described and examples given. Also included are sections on how the user can obtain data base buffer pool statistics, on user generation of randomizing modules for use with HDAM file organizations, and on the IMS/360 data base log tape record format.

MESSAGE EDITING CAPABILITIES

IMS/360 provides edit functions for both message input and output. In addition, IMS/360 permits the inclusion of user-written routines to perform edit functions for both message input and output. User-supplied edit routines can be defined for input transaction code edit, input message switching edit, and physical terminal output edit. An edit routine for operation with input from a 1030 communication system is also required if the 1030 terminal has been specified in IMS/360 system definition. User Edit routine exits are provided for the 2980 and 2770 users also, however IMS/360 provides functional 'default' routines for these exits which may be replaced by the user if these defaults are not adequate for his operation. Except for an edit routine required for 1030 input terminal operation, all user-written edit routines are optional. All user-written edit routines are incorporated into the IMS/360 control program nucleus during Stage 2 of IMS/360 system definition.

BASIC IMS/360 EDIT FUNCTIONS

The IMS/360-supplied edit function performs the following for input messages:

- On the first segment of each message, whether the message type is a transaction, a command, or a message switch, removes leading control characters and blanks

- On all subsequent message segments, whether the message type is a transaction or a command (except /BROADCAST), removes leading control characters

- On _all_ segments, removes all line-control characters.

- On all segments, if the message type is a transaction, removes trailing carriage-return characters.

- On all segments, when the entering or transmission of backspaces is a normal correction procedure on the entering terminal, eliminates backspaces on a one-for-one basis

- Removes password and replaces it with a blank when necessary to provide separation between the transaction code, logical terminal, or command verb and following data.

- On the first segment of a message, inserts transaction code or logical terminal name as defined by prior SET command, ahead of entered data. A blank is inserted following the transaction code if necessary to obtain separation between the inserted transaction code and the entered data.

The IMS/360-supplied edit function for output messages inserts any necessary idle characters after new line, line feed, and tab characters. For the 1033, these characters are inserted after all characters, to facilitate proper terminal operation. Line control characters are also added for operation of the communication line.

MESSAGE FORMAT SERVICE EDITING

If the input has been processed by the IMS/360 message format service, the editing performed is dependent on the descriptions provided through the format language utility. Since input segments from the device may have no relationship to input message segments after message format service editing, the input segment from the device is not available to user-written edit routines.

TRANSACTION CODE (INPUT) EDITS

IMS/360 gives the user the ability to specify during system definition the inclusion in the IMS/360 control program nucleus of one or more user-supplied input edit routines. This provides the user with the capability of editing input messages before they are enqueued for scheduling. When IMS/360 is executed, this user edit function is performed in addition to the basic IMS/360 edit function or Message Format Service editing and subsequent to this function. The user may specify to system definition up to 255 editing routines and also which edit routine is to be used, by transaction type.

The user should know the contents and meaning of the various fields in the IMS/360 control blocks (defined in the _IMS/360 System Manual_), and may test them in an edit routine. However, under no circumstances may an edit routine modify any of these blocks. (Details of the system definition procedure are found in Chapter 3 of this manual.)

If specified, a user-supplied input edit routine gains control after each message data segment is processed by the IMS/360 basic input edit or Message Format Service. Transaction code validity and security will have already been checked.

Note: The user edit routine would not be entered if the transaction code were the only data in the message segment, and the transaction is a conversational transaction.

Upon entry to a user-supplied transaction code edit routine, the following interface applies:

Register Number	Content
1	The buffer location of the input message segment after translation to EBCDIC but prior to the IMS/360 basic editing. The first two bytes of the buffer contain a binary count of the message length. The third and fourth bytes of the buffer are binary zeros. The fifth byte contains the first byte of message text. The binary count includes the four-byte "prefix". As the input buffer has no relationship to the segment after it has been processed by the Message Format Service, this register will point to the resultant segment (same as DECAREA) if the message was processed by this service instead of the BASIC INPUT EDIT service. The fourth byte of the message

	segment (Z2) will be X'00' if the Basic edit service was used. It will contain a X'01', X'02', or X'03' if the Message Format Service was used, signifying that option 1, 2, or 3 respectively was selected for the message by the format designer.
7	The IMS/360 Communication Terminal Block (CTB) address for the physical terminal from which the message segment was entered.
9	The IMS/360 Communication Line Block (CLB) address for the communication line from which the message was entered. This control block starts with a BTAM DECB. The DECAREA field in the DECB contains the address of a buffer. This buffer contains the input message segment after IMS/360 editing. The first four bytes are two bytes of binary count (length of this message segment) and two bytes of binary zeros as above. The length of this buffer is equivalent to the binary count pointed to by Register 1 plus 10 if basic editing was performed. If the input was processed using the IMS/360 message format service, the length of this buffer is given by the first two bytes of the buffer (length of this message segment). No extra space is provided in this buffer for user-written edit routines.
13	Save area address for use by an edit routine.
14	Return address to IMS/360 control program
15	Entry point address to the invoked edit routine. The entry point name and load module name for an edit routine must be the same and equivalent to the name used for the edit routine in system definition.

Upon entry to a transaction code edit routine, all registers should be saved. Upon exit, all registers should be restored. The reader may wish to refer to the IMS/360 System Manual - Volume I (LY20-0629) for definition of IMS/360 control blocks.

If the input was processed by the IMS/360 basic edit, the user may use either the message segment in the buffer addressed by Register 1 or that addressed by the DECAREA field as input to edit. If the input was processed by the IMS/360 message format service, the user may use only the message segment addressed by the DECAREA field as input to edit.

The user-supplied edit routine must place the text of the user-edited message segment to be returned to IMS/360 in the buffer addressed by the DECAREA field. If the input was processed by the IMS/360 basic edit, this buffer is always 10 bytes greater than the two-byte binary count at the beginning of the message segment, and the user can expand the length of the message segment. Alternatively, the user can reduce the length of the message segment to any desired size. The format of the user-edited message segment in the buffer upon return to IMS/360 must be two bytes of binary count, two bytes of binary zeros (except when input was processed by the IMS/360 message format service - the second two bytes should not be changed), and edited text.

Upon return to the IMS/360 control program, Register 15 is used to define subsequent IMS/360 functions. If Register 15 is zero, IMS/360 will process the message segment normally. If Register 15 equals 4, this message segment is cancelled by IMS/360. If Register 15 is greater than 4, the entire message of which this segment is a part is cancelled.

EXAMPLE OF TRANSACTION CODE (INPUT) EDIT

Assume a multisegment transaction named 'ICS'. Normally, the first segment of this message will contain 'ICS GN', meaning to get the next segment of a given message, or 'ICS CAN', meaning to cancel this message. A user-supplied edit routine allows further input flexibility, as shown in the following decision table.

	MSG AS REC'D AND EDITED BY IMS/360	MSG AS REEDITED BY USER EDIT ROUTINE
First Segments	'ICS GN'	As received
	'ICS'	'ICS GN'
	'ICS CAN'	msg canceled
	Any other	msg canceled
Other Segment	'GN'	As received
	'CAN'	msg canceled
	Any other	segment canceled

The transaction code edit routine allows the input of a shortened format for the 'ICS GN' message segment.

The following is an example of the transaction code (input) edit.

```
STMT   SOURCE STATEMENT                                             F01FEB69

ICSEDIT  CSECT
*********************************************************************
*********************************************************************
*               USER  MESSAGE  EDIT  ROUTINE                        *
*********************************************************************
*          *                   *                                    *
*          * MESSAGE RECEIVED * RETURN CODE  RETURNED MESSAGE       *
*-------------------------------------------------------------------*
*          *  'ICS GN  '       *    RC=0        'ICS GN  '          *
* FIRST    *  'ICS CAN'        *    RC=8        'ICS CAN'           *
* SEGMENT  *  'ICS     '       *    RC=0        'ICS GN  '          *
*          *  ANY OTHER        *    RC=4        AS RECEIVED         *
*-------------------------------------------------------------------*
*          *                   *                                    *
* OTHER    *  'GN  '           *    RC=0        'GN  '              *
* SEGMENTS *  'CAN'            *    RC=8        'CAN'               *
*          *  ANY OTHER        *    RC=4        AS RECEIVED         *
*********************************************************************
* RETURN CODE       MEANING                                         *
*-------------------------------------------------------------------*
*   RC=0        PROCESS THIS MESSAGE SEGMENT                        *
*   RC=4        CANCEL THIS SEGMENT                                 *
*   RC=8        CANCEL THIS MESSAGE                                 *
*********************************************************************
*********************************************************************
*
         STM   14,12,12(13)                SAVE REGS
         LR    R12,R15                     ESTABLISH BASE REG
         USING ICSEDIT,R12
         SR    R15,R15                     CLEAR INDICATOR RETURN REG
         USING CTB,R7                      SET CTB BASE
         USING IECTDECB,R9                 SETUP CLB POINTERS
*** IS THIS THE FIRST SEGMENT OF THE MESSAGE? **********************
*
         L     R8,DECAREA                  POINT TO PRE-EDITED MESSAGE
         SR    R2,R2                       CLEAR WORK REG
         IC    R2,1(R8)                    LOAD LENGTH OF MESSAGE
         BCTR  R2,0                        DECREMENT BY ONE
         EX    R2,MAKUPER                  TRANSLATE MSG TO UPPER CASE
         TM    CTBFLAG3,CTB3SEG1           IS THIS FIRST SEG OF MSG?
         BZ    MULTSEG                     NO,BRANCH
FIRSEG   EQU   *                           YES, THIS IS FIRST SEG
         CLC   4(3,R8),ICS                 IS THIS A LEGIT ICS MESSAGE?
         BE    GUDICS1                     YES,BRANCH
*
*** SET RETURN CODE TO INDICATE 'CANCEL THIS SEGMENT' **************
*
CANSEG   LA    R15,4                       LOAD INDIC TO CANCEL THIS SEGM
         B     RETURN                      GO RETURN
GUDICS1  EQU   *
         CLC   7(4,R8),CAN                 CHECK FOR CANCEL MSG
         BNE   NOTCAN1                     IF NOT CAN MSG,GO PROCESS
*
*** SET RETURN CODE TO INDICATE 'CANCEL THIS MESSAGE' **************
*
```

```
STMT    SOURCE STATEMENT                                                    F15OCT70    3/27/72

   1            ICLI   CLBBASE=0,CNTBASE=0,CTBBASE=0,COBBASE=0,CTTBASE=0
   2+***********************************************************************************
   3+*
   4+*          CLB - COMMUNICATION LINE BLOCKS DSECT. DSECT NAME IS IECTDECB.
   5+*
   6+***********************************************************************************
   7+IECTDECB DSECT   DECB DUMMY SECTION
   8+*          -------------------------------------
   9+*          +                                   +
  10+*      0   +       STANDARD EVENT CONTROL BLOCK +
  11+*          +                                   +
  12+*          +-----------------------------------+
  13+*          +                 +                 +
  14+*      4   +  OPERATION TYPE +    AREA LENGTH  +
  15+*          +                 +                 +
  16+*          +-----------------------------------+
  17+*          + ON-LINE +                         +
  18+*      8   +TERMINAL +       ADDRESS OF DCB    +
  19+*          +  TEST   +                         +
  20+*          +-----------------------------------+
  21+*          +         +                         +
  22+*     12   +RESERVED +       ADDRESS OF AREA   +
  23+*          +         +                         +
  24+*          +-----------------------------------+
  25+*          +                 +                 +
  26+*     16   +    SENSE BYTES  +  RESIDUAL COUNT +
  27+*          +     1   &   2   +                 +
  28+*          +-----------------------------------+
  29+*          +         +                         +
  30+*     20   + COMMAND |  ADDRESS OF TERMINAL LIST +
  31+*          +   CODE  +                         +
  32+*          +-----------------------------------+
  33+*          +         + RELATIVE+        +      +
  34+*     24   +  STATUS +   LINE  + ADDRESS + VRC/LRC  +
  35+*          +  FLAGS  +  NUMBER + RESPONSE+ RESPONSE +
  36+*          +-----------------------------------+
  37+*          +         +         +               +
  38+*     28   +  TP-OP  +  ERROR  +    CSW STATUS +
  39+*          +   CODE  +  STATUS +               +
  40+*          +-----------------------------------+
  41+*          +         +                         +
  42+*     32   +RESERVED +     ADDRESS OF CURRENT  +
  43+*          +         +      ADDRESSING ENTRY   +
  44+*          +-----------------------------------+
  45+*          +         +                         +
  46+*     36   +RESERVED +     ADDRESS OF CURRENT  +
  47+*          +         +        POLLING ENTRY    +
  48+*          +-----------------------------------+
  49+*          +         +         +               +
  50+*     40   +RESERVED +RESERVED +  WRITE AREA LENGTH +
  51+*          +         +         +               +
  52+*          +-----------------------------------+
  53+*          +         +                         +
  54+*     44   +RESERVED +   ADDRESS OF WRITE AREA +
  55+*          +         +

STMT    SOURCE STATEMENT                                                    F15OCT70    3/27/72

  56+*          -------------------------------------
```

```
  58+DECSDECB DS     1F STATUS FLAG + ADDRESS OF THE TCB

  60+DECTYPE  DS     1H OPERATION TYPE

  62+DECLNGTH DS     1H AREA LENGTH

  64+DECONLTT DS     0CL1 RESERVED FOR ON-LINE TERMINAL TEST
  65+DECDCBAD DS     1F ADDRESS OF DCB

  67+DECAREA  DS     1F ADDRESS OF AREA

  69+DECSENS0 DS     1C 1ST SENSE BYTE

  71+DECSENS1 DS     1C 2ND SENSE BYTE

  73+DECCOUNT DS     1H RESIDUAL COUNT

  75+DECCMCOD DS     0CL1 COMMAND CODE
  76+DECENTRY DS     1F ADDRESS OF TERMINAL LIST


  79+DECFLAGS DS     1C STATUS FLAGS

  81+DECRLN   DS     1C RELATIVE LINE NUMBER

  83+DECRESPN DS     1H RESPONSE FIELDS

  85+DECTPCOD DS     1C TP-OP CODE

  87+DECERRST DS     1C ERROR STATUS

  89+DECCSWST DS     1H CSW STATUS

  91+DECADRPT DS     1F ADDRESS OF CURRENT ADDRESSING ENTRY

  93+DECPOLPT DS     1F ADDRESS OF CURRENT POLLING ENTRY

  95+         DS     2C RESERVED

  97+DECWLNG  DS     1H WRITE AREA LENGTH

  99+DECWAREA DS     1F ADDRESS OF WRITE AREA

 101+CLBDECB  EQU    DECSDECB
 102+CLBRCNTR DS     1C RETRY COUNTER

 104+CLBFLAG1 DS     C .           BIT USAGE
 105+CLB1COB  EQU    X'80' .     0 - COB REQUIRED FOR CTB'S ON THIS LINE
 106+CLB1TEST EQU    X'40' .     1 - TEST RECYCLE REQUEST
 107+CLB1IBUF EQU    X'20' .     2 - INPUT BUFFER ALLOCATED
 108+CLB1OBUF EQU    X'10' .     3 - OUTPUT BUFFER ALLOCATED
 109+*        EQU    X'08' .     4 - RESERVED FOR FUTURE USE
 110+CLB1QCRI EQU    X'04' .     5 - QCR ALLOCATED  (INPUT)
 111+CLB1RESP EQU    X'02' .     6 - RESPONSE WAITING
 112+CLB1QCRO EQU    X'01' .     7 - QCR ALLOCATED   (OUTPUT)
```

x


```
 114+CLBFLAG2 DS     C
 115+CLB2NOIN EQU    X'80' .     0 - NO INPUT ALLOWED
 116+CLB2NOOU EQU    X'40' .     1 - NO OUTPUT ALLOWED
 117+CLB2NOQU EQU    X'20' .     2 - DO NOT QUEUE ON THIS LINE
 118+*        EQU    X'10' .     3 - RESERVED FOR FUTURE USE
 119+CLB2IDLE EQU    X'08' .     4 - IDLE LINE
 120+CLB2INP  EQU    X'04' .     5 - INPUT IN CONTROL
 121+CLB2OUTP EQU    X'02' .     6 - OUTPUT IN CONTROL
 122+CLB2AUTO EQU    X'01' .     7 - AUTOPOLL

 124+CLBFLAG3 DS     C .
 125+CLB3INP  EQU    X'80' .     0 - 0=OUTPUT NEXT, 1=INPUT NEXT ON THIS LINE
 126+CLB3CSUB EQU    X'40' .     1 - COMMAND ENTERED AS NON-FIRST SEGMENT
 127+CLB3DOPN EQU    X'20' .     2 - LINE IS OPENED DYNAMICALLY
 128+CLB3STOP EQU    X'10' .     3 - LINE HAS BEEN IDLED FOR SHUTDOWN
 129+CLB3CBUF EQU    X'08' 4 - CONDENSED BUFFER ALLOCATED
 130+CLB3ICMD EQU    X'04' 5 - INTERNAL COMMAND IN PROGRESS
 131+*        EQU    X'02' .     6 - RESERVED FOR FUTURE USE
 132+*        EQU    X'01' .     7 - RESERVED FOR FUTURE USE

 134+CLBCNTPT DS     F . POINTER TO CNT FOR RESPONSE                         $
 135+CLBTEMP1 DS     F .
 136+CLBTEMP4 DS     F .              USED FOR TEMPORARY STORAGE
 137+CLBTEMP5 DS     F .              USED FOR TEMPORARY STORAGE

 139+CLBCSID  DS     0C . FIND DEST FIELD                                    $
 140+*                       0       DL/1 USE
 141+*                       1       RESTART IN PROGRESS
 142+CLBCSCVB EQU    X'20' . 2        CVB FOUND AS DESTINATION
 143+*                       3       ALWAYS 0 FOR CLB
 144+CLBCSFND EQU    X'08' . 4       CNT/SMB NOT FOUND
 145+CLBCSSMB EQU    X'04' . 5 = 0    CNT FOUND
 146+*                         = 1  SMB FOUND
 147+CLBCSDST EQU    X'02' . 6        FNDDST ENTERED (CLBCNTP POINTS TO DEST)
 148+CLBCSACT EQU    X'01' . 7       ACTIVE POLLING LIST

 150+CLBPOLAD DS     F .  POLLING LIST ADDRESS
 151+CLBCTBPT DS     H .  FIRST CTB ON LINE
 152+CLBCRCTB DS     H .  CURRENT CTB
 153+CLBCNTQB DS     2F . QCB FOR CNTS FOR OUTPUT
 154+CLBSMBPT DS     F .  POINTER TO BLOCK FOR QUEUEING CNT/SMB/CVB          $
 155+CLBCTBPS DS     H .
 156+CLBDPOLC DS     C
 157+CLBDCTL  DS     C .              CLB DEVICE CONTROL BYTE (USED BY DDM)
 158+CLBINBUF DS     F .        INPUT BUFFER POINTER
 159+CLBOUTBF DS     F .        OUTPUT BUFFER POINTER
 160+*
 161+* IF ($) FALLS IN COLUMN 71 THIS FIELD WILL BE SAVED IN A COB IF REQ
```

x


```
199+***********************************************************************
200+*
201+*         CTB - COMMUNICATION TERMINAL BLOCKS DSECT.
202+*
203+***********************************************************************
204+CTB      DSECT
205+CTBTYPE  DS     C .        CTT NUMBER
206+CTBLINE  DS     C .        LINE NUMBER
207+CTBTERM  DS     3C .       TERMINAL ADDRESS

209+CTBFLAG1 DS     C .
210+CTB1CONV EQU    X'80' .  0 THIS CTB IN CONVERSATION
211+CTB1MAST EQU    X'40' .  1 MASTER TERMINAL
212+CTB1SUBP EQU    X'20' .  2 SUBPOOL CTB
213+CTB1DIAL EQU    X'10' .  3 DIAL CTB IS PHYSICALLY CONNECTED
214+CTB1SIGN EQU    X'08' .  4 DIAL CTB IS LOGICALLY CONNECTED
215+CTB1HELD EQU    X'04' .  5 CONVERSATION IN PROGRESS WAS HELD
216+CTB1PCNT EQU    X'02' .  6 PRESET CNT
217+CTB1PSMB EQU    X'01' .  7 PRESET SMB

219+CTBFLAG2 DS     C .
220+CTB2NOIN EQU    X'80' .  0 NO INPUT
221+CTB2NOOU EQU    X'40' .  1 NO OUTPUT
222+CTB2NOQU EQU    X'20' .  2 NO QUEUEING
223+CTB2LOCK EQU    X'10' .  3 LOCKED
224+CTB2TEST EQU    X'08' .  4 TEST MODE
225+CTB2EXCL EQU    X'04' .  5 EXCLUSIVE MODE
226+CTB2INOP EQU    X'02' .  6 INOPERABLE
227+CTB2EDIT EQU    X'01' .  7 USER OUTPUT EDIT REQUESTED

229+CTBFLAG3 DS     C .
230+CTB3READ EQU    X'80' .  0 INPUT ONLY DEVICE
231+CTB3QSYS EQU    X'40' .  1 QUEUE ALL SYSTEM MESSAGES
232+CTB3LOOP EQU    X'20' .  2 LOOP TEST PENDING
233+CTB3CHE  EQU    X'10' .  3 LINE DISCONNECT REQUESTED
234+CTB3QERR EQU    X'08' .  4 ERROR ON GET NEXT FOR OUTPUT
235+CTB3QMSG EQU    X'04' .  5 QUEUE CAUSE OF ERROR WITH SYSMSG
236+CTB3LAST EQU    X'02' .  6 LAST CTB ON THIS LINE
237+CTB3SEG1 EQU    X'01' .  7 FIRST SEGMENT

239+CTBFLAG4 DS     C .
240+CTB4CNCL EQU    X'80' .  0  DEQUEUE MESSAGE IN PROCESS REQUESTED
241+CTB4OUTP EQU    X'40' .   OUTPUT ONLY TERMINAL
242+*        EQU    X'20' .  2  NOT USED
243+*        EQU    X'10' .  3  NOT USED
244+*        EQU    X'08' .  4  NOT USED
245+*        EQU    X'04' .  5  NOT USED
246+*        EQU    X'02' .  6  NOT USED
247+*        EQU    X'01' .  7  NOT USED

249+CTBACTL  DS     C .        FLAGS TO BE USED BY ANALYZER
250+CTBAINP  EQU    X'80' .  0  0=OUTPUT NEXT, 1= INPUT NEXT
251+CTBAEOM  EQU    X'40' .  1  1=END OF OUTPUT MSG
252+CTBASHUT EQU    X'20' .  2  1=SYSTEM SHUTDOWN
253+CTBAINC  EQU    X'10' .  3  1=INCORE SYSTEM MESSAGE
```

```
 254+CTBAMULT EQU    X'08' .  4   REJECT IF NOT MULTIPLE SEGMENT MESSAGE
 255+CTBAERR  EQU    X'04' .  5   ERROR FOUND ON LAST SEGMENT PASSED

 257+CTBDCTL  DS     C .        FLAGS USED BY DD ROUTINES
 258+CTBDCTL2 DS     C .        FLAGS USED BY DD ROUTINES

 260+CTBFEAT  DS     C .        FEATURES FLAGS
 261+CTBFSYN1 EQU    X'08' ASYNCHRONOUS DEVICE,  SYNC FLAG 1
 262+CTBFSYN2 EQU    X'04' ASYNCHRONOUS DEVICE,  SYNC FLAG 2
 263+CTBFPAGE EQU    X'02' PAGING IN PROCESS
 264+CTBFNAPD EQU    X'01' AUTOMATIC PAGE DELETION NOT REQUESTED

 266+CTBCOMP  DS     C .        COMPONENTS TWO BITS FOR EACH
 267+CTBC1NA  EQU    X'80' COMPONENT 1 IS NOT ATTACHED           3088
 268+CTBC1IP  EQU    X'40' COMPONENT 1 IS INOPERABLE             3088
 269+CTBC2NA  EQU    X'20' COMPONENT 2 IS NOT ATTACHED           3088
 270+CTBC2IP  EQU    X'10' COMPONENT 2 IS INOPERABLE             3088
 271+CTBC3NA  EQU    X'08' COMPONENT 3 IS NOT ATTACHED           3088
 272+CTBC3IP  EQU    X'04' COMPONENT 3 IS INOPERABLE             3088
 273+CTBC4NA  EQU    X'02' COMPONENT 4 IS NOT ATTACHED           3088
 274+CTBC4IP  EQU    X'01' COMPONENT 4 IS INOPERABLE             3088

 276+CTBINCT  DS     H .        INPUT MSG COUNT
 277+CTBOUTCT DS     H .        OUTPUT MSG COUNT
 278+CTBCPMP  DS     H .        PASSWORD MATRIX POINTER
 279+CTBCNTP  DS     H .        LOGICAL TERMINAL POINTER
 280+CTBPREST DS     H .        PRESET DESTINATION OFFSET
 281+CTBCOB   DS     H .        OFFSET TO COMMUNICATION OVERFLOW BLOCK
 282+CTBCCB   DS     H .        OFFSET TO CONVERSATIONAL CONTROL BLOCK
 283+CTBCXB   DS     H .        COMMUNICATION EXTENT BLOCK POINTER
 284+CTBPGNO  DS     H .        CURRENT PAGE BEING VIEWED
```

MESSAGE SWITCHING (INPUT) EDIT

A facility similar to the transaction code (input) edit is provided for message switching. The optionally supplied, user-written routine, whose CSECT and load module name must be DFSCNTE0, is included in the user's system at IMS/360 system definition time. Only one message switching edit routine may be specified for an IMS/360 online control program. This routine is specified for inclusion with the online control program during system definition. The interface between the IMS/360 control program and the user-supplied message switching edit routine is the same as previously defined for the transaction code edit routine.

EXAMPLE OF MESSAGE SWITCHING EDIT

The user-supplied edit routine might be used to identify, in the text of the output message to the output terminal, the logical terminal name and message number from which the message was entered.

Assume the following message being entered from a logical terminal named 'XSYSI' and is input message number one.

    ABC SEND ALL XYZ MSGS TO THIS TERMINAL

The message as received at the output terminal associated with logical terminal name ABC has the input logical terminal name and input message number appended to it by the user's edit routine.

    ABC SEND ALL XYZ MSGS TO THIS TERMINAL XSYSI 1

In this example, the logical terminal input name is used. This name exists within the IMS/360 control block for the input logical terminal, the Communication Name Table (CNT). The CNT is addressed by a field in the Communication Line Block called CLBCNTPT. The field in the CNT containing the logical terminal name is called CNTNAME. The reader is referred to the IMS/360 control blocks in Volume I of the _IMS/360 System Manual_ (LY20-0629).

Upon entry to the message switching edit routine, the register interface is identical to that for transaction code input edit routines.

The following is an example of the message switching edit.

```
  1 ***********************************************************************
  2 *               USER EDIT ROUTINE FOR MESSAGE SWITCHING               *
  3 *---------------------------------------------------------------------*
  4 *                                                                     *
  5 *         THE LOGICAL TERMINAL NAME OF THE INPUTTING TERMINAL AND     *
  6 *         THE MESSAGE NUMBER ARE ADDED TO THE END OF THE MESSAGE      *
  7 ***********************************************************************

  9 DFSCNTE0 CSECT
 10          SAVE  (14,12)
 11+         DS    0H
 12+         STM   14,12,12(13) SAVE REGISTERS
 13          LR    R12,R15             * ESTABLISH
 14          USING DFSCNTE0,R12        *  ADDRESSABILITY
 15          USING CNT,R6
 16          USING CTB,R7
 17          USING IECTDECB,R9         CLB POINTER
 18          USING SCD,R11
 19 *
 20          SR    R15,R15             CLEAR RETURN CODE

 22 ************     FIND THE END OF THE PRE-EDITED MESSAGE    ***************
 23 *                                                                     *
 24          L     R5,DECAREA          POINT TO MESSAGE
 25          SR    R4,R4               CLEAR WORK REG
 26          IC    R4,0(,R5)           LOAD D OF 'DL'
 27          SLL   R4,8                SHIFT TO MAKE ROOM FOR 'L'
 28          IC    R4,1(,R5)           INSERT 'L'
 29          AR    R5,R4               R5= END OF MESSAGE

 31 ************     GET LOGICAL TERMINAL NAME, AND ADD IT TO MSG   *********
 32 *                                                                     *
 33          LH    R6,CTBCNTP          OFFSET TO CNT
 34          A     R6,SCDCNT           R6= ADDR OF INPUTTING CNT
 35          MVC   1(5,R5),CNTNAME     INSERT 5 CHARS OF NAME

 37 ************     NOW FIND AND INSERT MESSAGE NUMBER   ******************
 38 *                                                                     *
 39          LH    R3,CTBINCT          LOAD MSG NUMBER
 40          CVD   R3,MSGNUMP
 41          UNPK  MSGNUM(4),MSGNUMP+4(4) *  CONVERT TO
 42          OI    MSGNUM+3,240          *   CHARACTERS
 43          MVC   7(3,R5),MSGNUM+1    SLIDE NUMBER NEXT TO NAME
 44          MVI   6(R5),C' '          BLANK SEPARATOR

 46 ************     CHANGE 'DL' TO REFLECT NEW MSG LENGTH   ****************
 47 *                                                                     *
 48          SR    R5,R4               R5= START OF MSG (DL)
 49          LA    R4,9(,R4)           NEW LENGTH IS 9 MORE
 50          STC   R4,1(,R5)           REPLACE 'L'
 51          SRL   R4,8
 52          STC   R4,0(,R5)           REPLACE 'D'
```

```
STMT   SOURCE STATEMENT                                                      F01MAY72   2/13/73

  54 ************    RETURN ROUTINE    ****************************************
  55 *                                                                       *
  56            L      R14,12(,R13)          FETCH RETURN ADDRESS
  57 *                                       R13 STILL GOOD, R15=RC
  58            RETURN (0,12)
  59+           LM     0,12,20(13) RESTORE THE REGISTERS
  60+           BR     14 RETURN


  62 ************    CONSTANTS    *********************************************
  63 *                                                                       *
  64 MSGNUMP    DS     D
  65 MSGNUM     DS     F
  66 *
  67            LTORG



  69            REQUATE
  70+**********************************************************************
  71+*                                                                    *
  72+*                    REGISTER EQUATES                                *
  73+*                                                                    *
  74+**********************************************************************


  76+R0         EQU    0
  77+R1         EQU    1
  78+R2         EQU    2
  79+R3         EQU    3
  80+R4         EQU    4
  81+R5         EQU    5
  82+R6         EQU    6
  83+R7         EQU    7
  84+R8         EQU    8
  85+R9         EQU    9
  86+R10        EQU    10
  87+R11        EQU    11
  88+R12        EQU    12
  89+R13        EQU    13
  90+R14        EQU    14
  91+R15        EQU    15
```

```
STMT   SOURCE STATEMENT                                               F01MAY72   2/13/73

 93          ICLI   CLBBASE=0,CNTBASE=0,CTBBASE=0
 94+***********************************************************************
 95+*
 96+*        CLB - COMMUNICATION LINE BLOCKS DSECT. DSECT NAME IS IECTDECB.
 97+*
 98+***********************************************************************
 99+IECTDECB DSECT  DECB DUMMY SECTION
100+*        -----------------------------------------
101+*        +                                       +
102+*     0  +       STANDARD EVENT CONTROL BLOCK    +
103+*        +                                       +
104+*        +---------------------------------------+
105+*        +                   +                   +
106+*     4  +  OPERATION TYPE   +    AREA LENGTH    +
107+*        +                   +                   +
108+*        +---------------------------------------+
109+*        + ON-LINE +                             +
110+*     8  +TERMINAL +        ADDRESS OF DCB       +
111+*        +  TEST   +                             +
112+*        +---------------------------------------+
113+*        +         +                             +
114+*    12  +RESERVED +        ADDRESS OF AREA      +
115+*        +         +                             +
116+*        +---------------------------------------+
117+*        +                   +                   +
118+*    16  +    SENSE BYTES    +   RESIDUAL COUNT  +
119+*        +     1   &    2    +                   +
120+*        +---------------------------------------+
121+*        +         +                             +
122+*    20  + COMMAND |   ADDRESS OF TERMINAL LIST  +
123+*        +   CODE  +                             +
124+*        +---------------------------------------+
125+*        +         + RELATIVE+         +         +
126+*    24  +  STATUS +   LINE  + ADDRESS + VRC/LRC +
127+*        +  FLAGS  +  NUMBER + RESPONSE+ RESPONSE +
128+*        +---------------------------------------+
129+*        +         +         +                   +
130+*    28  +  TP-OP  +  ERROR  +     CSW STATUS    +
131+*        +   CODE  +  STATUS +                   +
132+*        +---------------------------------------+
133+*        +         +                             +
134+*    32  +RESERVED +      ADDRESS OF CURRENT     +
135+*        +         +       ADDRESSING ENTRY      +
136+*        +---------------------------------------+
137+*        +         +                             +
138+*    36  +RESERVED +      ADDRESS OF CURRENT     +
139+*        +         +        POLLING ENTRY        +
140+*        +---------------------------------------+
141+*        +         +         +                   +
142+*    40  +RESERVED +RESERVED +  WRITE AREA LENGTH +
143+*        +         +         +                   +
144+*        +---------------------------------------+
145+*        +         +                             +
146+*    44  +RESERVED +   ADDRESS OF WRITE AREA     +
147+*        +         +                             +
```

STMT   SOURCE STATEMENT   F01MAY72   2/13/73

```
148+*        -----------------------------------------
```

```
STMT   SOURCE STATEMENT                                          F01MAY72   2/13/73

 150+DECSDECB DS     1F STATUS FLAG + ADDRESS OF THE TCB

 152+DECTYPE  DS     1H OPERATION TYPE

 154+DECLNGTH DS     1H AREA LENGTH

 156+DECONLTT DS     OCL1 RESERVED FOR ON-LINE TERMINAL TEST
 157+DECDCBAD DS     1F ADDRESS OF DCB

 159+DECAREA  DS     1F ADDRESS OF AREA

 161+DECSENS0 DS     1C 1ST SENSE BYTE

 163+DECSENS1 DS     1C 2ND SENSE BYTE

 165+DECCOUNT DS     1H RESIDUAL COUNT

 167+DECCMCOD DS     OCL1 COMMAND CODE
 168+DECENTRY DS     1F ADDRESS OF TERMINAL LIST


 171+DECFLAGS DS     1C STATUS FLAGS

 173+DECRLN   DS     1C RELATIVE LINE NUMBER

 175+DECRESPN DS     1H RESPONSE FIELDS

 177+DECTPCOD DS     1C TP-OP CODE

 179+DECERRST DS     1C ERROR STATUS

 181+DECCSWST DS     1H CSW STATUS

 183+DECADRPT DS     1F ADDRESS OF CURRENT ADDRESSING ENTRY

 185+DECPOLPT DS     1F ADDRESS OF CURRENT POLLING ENTRY

 187+         DS     2C RESERVED

 189+DECWLNG  DS     1H WRITE AREA LENGTH

 191+DECWAREA DS     1F ADDRESS OF WRITE AREA

 193+CLBDECB  EQU    DECSDECB
 194+CLBRCNTR DS     1C RETRY COUNTER

 196+CLBFLAG1 DS     C .            BIT USAGE
 197+CLB1COB  EQU    X'80' .    0 - COB REQUIRED FOR CTB'S ON THIS LINE
 198+CLB1TEST EQU    X'40' .    1 - TEST RECYCLE REQUEST
 199+CLB1IBUF EQU    X'20' .    2 - INPUT BUFFER ALLOCATED
 200+CLB1OBUF EQU    X'10' .    3 - OUTPUT BUFFER ALLOCATED
 201+CLB1TEMQ EQU    X'08' .    4 - USE TEMPORARY QUEUE
 202+*        EQU    X'04' .   5 - RESERVED FOR FUTURE USE
 203+CLB1RESP EQU    X'02' .    6 - RESPONSE WAITING
 204+*        EQU    X'01' .   7 - RESERVED FOR FUTURE USE
```

```
 206+CLBFLAG2 DS     C
 207+CLB2NOIN EQU    X'80' .     0 - NO INPUT ALLOWED
 208+CLB2NOOU EQU    X'40' .     1 - NO OUTPUT ALLOWED
 209+CLB2NOQU EQU    X'20' .     2 - DO NOT QUEUE ON THIS LINE
 210+CLB2RSS  EQU    X'10' .     4 - CONCENTRATOR LINE
 211+CLB2IDLE EQU    X'08' .     4 - IDLE LINE
 212+CLB2INP  EQU    X'04' .     5 - INPUT IN CONTROL
 213+CLB2OUTP EQU    X'02' .     6 - OUTPUT IN CONTROL
 214+CLB2AUTO EQU    X'01' .     7 - AUTOPOLL

 216+CLBFLAG3 DS     C .
 217+CLB3INP  EQU    X'80' .     0 - 0=OUTPUT NEXT, 1=INPUT NEXT ON THIS LINE
 218+CLB3CSUB EQU    X'40' .     1 - COMMAND ENTERED AS NON-FIRST SEGMENT
 219+CLB3DOPN EQU    X'20' .     2 - LINE IS OPENED DYNAMICALLY
 220+CLB3STOP EQU    X'10' .     3 - LINE HAS BEEN IDLED FOR SHUTDOWN
 221+CLB3CBUF EQU    X'08' 4 - CONDENSED BUFFER ALLOCATED
 222+CLB3ICMD EQU    X'04' 5 - INTERNAL COMMAND IN PROGRESS
 223+CLB3CL40 EQU    X'02' .    6 - STATUS CHANGED BY CL40
 224+*        EQU    X'01' .     7 - RESERVED FOR FUTURE USE

 226+CLBCIB   DS     F .              POINTER TO CURRENT CIB
 227+CLBTEMP1 DS     F .
 228+CLBTEMP4 DS     F .              USED FOR TEMPORARY STORAGE
 229+CLBTEMP5 DS     F .              USED FOR TEMPORARY STORAGE

 231+CLBCSID  DS     0C . FIND DEST FIELD
 232+*                      0      DL/1 USE
 233+*                      1      RESTART IN PROGRESS
 234+CLBCSCVB EQU    X'20' . 2       CVB FOUND AS DESTINATION
 235+*                      3      ALWAYS 0 FOR CLB
 236+CLBCSFND EQU    X'08' . 4      CNT/SMB NOT FOUND
 237+CLBCSSMB EQU    X'04' . 5 = 0   CNT FOUND
 238+*                        = 1 SMB FOUND
 239+CLBCSDST EQU    X'02' . 6       FNDDST ENTERED (CLBCNTP POINTS TO DEST)
 240+CLBCSACT EQU    X'01' . 7      ACTIVE POLLING LIST

 242+CLBPOLAD DS     F . POLLING LIST ADDRESS
 243+CLBCTBPT DS     H . FIRST CTB ON LINE
 244+CLBCRCTB DS     H . CURRENT CTB
 245+CLBCNTQB DS     2F . QCB FOR CNTS FOR OUTPUT
 246+CLBCTBPS DS     H .
 247+CLBDPOLC DS     C
 248+CLBDCTL  DS     C .              CLB DEVICE CONTROL BYTE (USED BY DDM)
 249+CLBINBUF DS     F .        INPUT BUFFER POINTER
 250+CLBOUTBF DS     F .        OUTPUT BUFFER POINTER
 251+*
```

```
253+***********************************************************************
254+*
255+*        CTB - COMMUNICATION TERMINAL BLOCKS DSECT.
256+*
257+***********************************************************************
258+CTB      DSECT
259+CTBTYPE  DS    C .         CTT NUMBER
260+CTBLINE  DS    C .         LINE NUMBER
261+CTBTERM  DS    3C .        TERMINAL ADDRESS

263+CTBFLAG1 DS    C .
264+CTB1CONV EQU   X'80' .  0 THIS CTB IN CONVERSATION
265+CTB1MAST EQU   X'40' .  1 MASTER TERMINAL
266+CTB1SUBP EQU   X'20' .  2 SUBPOOL CTB
267+CTB1DIAL EQU   X'10' .  3 DIAL CTB IS PHYSICALLY CONNECTED
268+CTB1SIGN EQU   X'08' .  4 DIAL CTB IS LOGICALLY CONNECTED
269+CTB1HELD EQU   X'04' .  5 CONVERSATION IN PROGRESS WAS HELD
270+CTB1PCNT EQU   X'02' .  6 PRESET CNT
271+CTB1PSMB EQU   X'01' .  7 PRESET SMB

273+CTBFLAG2 DS    C .
274+CTB2NOIN EQU   X'80' .  0 NO INPUT
275+CTB2NOOU EQU   X'40' .  1 NO OUTPUT
276+CTB2NOQU EQU   X'20' .  2 NO QUEUEING
277+CTB2LOCK EQU   X'10' .  3 LOCKED
278+CTB2TEST EQU   X'08' .  4 TEST MODE
279+CTB2EXCL EQU   X'04' .  5 EXCLUSIVE MODE
280+CTB2INOP EQU   X'02' .  6 INOPERABLE
281+CTB2EDIT EQU   X'01' .  7 USER OUTPUT EDIT REQUESTED

283+CTBFLAG3 DS    C .
284+CTB3READ EQU   X'80' .  0 INPUT ONLY DEVICE
285+CTB3QSYS EQU   X'40' .  1 QUEUE ALL SYSTEM MESSAGES
286+CTB3LOOP EQU   X'20' .  2 LOOP TEST PENDING
287+CTB3CHE  EQU   X'10' .  3 LINE DISCONNECT REQUESTED
288+CTB3QERR EQU   X'08' .  4 ERROR ON GET NEXT FOR OUTPUT
289+CTB3QMSG EQU   X'04' .  5 QUEUE CAUSE OF ERROR WITH SYSMSG
290+CTB3LAST EQU   X'02' .  6 LAST CTB ON THIS LINE
291+CTB3SEG1 EQU   X'01' .  7 FIRST SEGMENT

293+CTBFLAG4 DS    C .
294+CTB4CNCL EQU   X'80' .  0  DEQUEUE MESSAGE IN PROCESS REQUESTED
295+CTB4OUTP EQU   X'40' .   OUTPUT ONLY TERMINAL
296+CTB4TRAC EQU   X'20' .  2  TRACE REQUESTED
297+*        EQU   X'10' .  3  NOT USED
298+*        EQU   X'08' .  4  NOT USED
299+*        EQU   X'04' .  5  NOT USED
300+*        EQU   X'02' .  6  NOT USED
301+*        EQU   X'01' .  7  NOT USED

303+CTBACTL  DS    C .         FLAGS TO BE USED BY ANALYZER
304+CTBAINP  EQU   X'80' .  0  0=OUTPUT NEXT, 1= INPUT NEXT
305+CTBAEOM  EQU   X'40' .  1  1=END OF OUTPUT MSG
306+CTBASHUT EQU   X'20' .  2  1=SYSTEM SHUTDOWN
307+CTBAINC  EQU   X'10' .  3  1=INCORE SYSTEM MESSAGE
```

```
STMT   SOURCE STATEMENT                                              F01MAY72   2/13/73

 308+CTBAMULT EQU    X'08' .  4   REJECT IF NOT MULTIPLE SEGMENT MESSAGE
 309+CTBAERR  EQU    X'04' .  5   ERROR FOUND ON LAST SEGMENT PASSED
 310+CTBASAVL EQU    X'02' .  6  STATION AVAILABLE

 312+CTBDCTL  DS     C .        FLAGS USED BY DD ROUTINES
 313+CTBDCTL2 DS     C .        FLAGS USED BY DD ROUTINES

 315+CTBFEAT  DS     C .        FEATURES FLAGS
 316+CTBFSYN1 EQU    X'08' .    4 ASYNCHRONOUS DEVICE, SYNC FLAG 1
 317+CTBFSYN2 EQU    X'04' .    5 ASYNCHRONOUS DEVICE, SYNC FLAG 2
 318+CTBFPAGE EQU    X'02' .    6 PAGING IN PROCESS
 319+CTBFNAPD EQU    X'01' .    7 AUTOMATIC PAGE DELETION NOT REQUESTED

 321+CTBCOMP  DS     C .        COMPONENTS TWO BITS FOR EACH
 322+CTBC1NA  EQU    X'80' .    0 COMPONENT 1 IS NOT ATTACHED
 323+CTBC1IP  EQU    X'40' .    1 COMPONENT 1 IS INOPERABLE
 324+CTBC2NA  EQU    X'20' .    2 COMPONENT 2 IS NOT ATTACHED
 325+CTBC2IP  EQU    X'10' .    3 COMPONENT 2 IS INOPERABLE
 326+CTBC3NA  EQU    X'08' .    4 COMPONENT 3 IS NOT ATTACHED
 327+CTBC3IP  EQU    X'04' .    5 COMPONENT 3 IS INOPERABLE
 328+CTBC4NA  EQU    X'02' .    6 COMPONENT 4 IS NOT ATTACHED
 329+CTBC4IP  EQU    X'01' .    7 COMPONENT 4 IS INOPERABLE

 331+CTBINCT  DS     H .        INPUT MSG COUNT
 332+CTBOUTCT DS     H .        OUTPUT MSG COUNT
 333+CTBCPMP  DS     H .        PASSWORD MATRIX POINTER
 334+CTBCNTP  DS     H .        LOGICAL TERMINAL POINTER
 335+CTBPRT   DS     0H .       CURRENT PRINT POINTER
 336+CTBPREST DS     H .        PRESET DESTINATION OFFSET
 337+CTBCOB   DS     H .        OFFSET TO COMMUNICATION OVERFLOW BLOCK
 338+CTBCCB   DS     H .        OFFSET TO CONVERSATIONAL CONTROL BLOCK
 339+CTBCRB   DS     0H .       COMMUNICATION RESTART BLOCK POINTER
 340+CTBCHAIN DS     0H .
 341+CTBCXB   DS     H .        COMMUNICATION EXTENT BLOCK POINTER
 342+CTBCIB   DS     0H .       COMMUNICATION INTERFACE BLOCK POINTER
 343+CTBPGNO  DS     H .        CURRENT PAGE BEING VIEWED
```

```
345+******************************************************************
346+*
347+*        CNT - COMMUNICATION NAME TABLES DSECT.
348+*
349+******************************************************************
350+CNT      DSECT
351+CNTQE    DS    2F .     QE FOR CNT Q OFF CLB
352+CNTQCBDQ DS    A DRRN OF NEXT MESSAGE TO READ
353+CNTQCBEQ DS    A DRRN OF LAST MESSAGE WRITTEN

355+CNTQFLG1 DS    C CNT QUEUE FLAG 1
356+*        EQU   X'80'        QUEUE READ IN PROCESS
357+*        EQU   X'40'        QUEUE 1 HAS MESSAGE ENQUEUED
358+*        EQU   X'20'        QUEUE 2 HAS MESSAGE ENQUEUED
359+*        EQU   X'10'        QUEUE 3 HAS MESSAGE ENQUEUED
360+*        EQU   X'08'        QUEUE 4 HAS MESSAGE ENQUEUED
361+*        EQU   X'04'        MESSAGE EXISTS IN BACKUP QUEUE
362+*        EQU   X'02'        QCBDQ POINTS TO QUEUE BLOCK RECORD

364+CNTQFLG2 DS    C CNT QUEUE FLAG 2
365+*        EQU   X'80'        THIS DESTINATION IS PERMANENT
366+*        EQU   X'40'        AVERAGE LENGTH IS AVAILABLE
367+*        EQU   X'20'        ENQ/DEQ COUNT IS AVAILABLE
368+*        EQU   X'10'        NAME FIELD EXISTS
369+*        EQU   X'0F'        DESTINATION TYPE 0 THRU 15

371+CNTQAVGL DS    H AVERAGE MESSAGE LENGTH FOR DESTINATION
372+CNTDQCT  DS    H NUMBER OF MESSAGES DEQUEUED
373+CNTNQCT  DS    H NUMBER OF MESSAGES ENQUEUED
374+CNTNAME  DC    D'0' LOGICAL TERMINAL NAME

376+CNTFLAG1 DS    C CNT FLAG 1
377+CNT1EDIT EQU   X'80' USER EDIT ROUTINE
378+CNT1MAST EQU   X'40' MASTER TERMINAL CNT
379+CNT1NOQU EQU   X'20' DO NOT QUEUE ON THIS CNT
380+CNT1NOSE EQU   X'10' DO NOT SEND TO THIS CNT
381+CNT1LOCK EQU   X'08' CNT IS LOCKED
382+CNT1SIGN EQU   X'04' SIGN ON SPECIFIED LTERM ONLY
383+CNT1QERR EQU   X'02' I/O ERROR OCCURED ON QUEUE
384+CNT1CNT  EQU   X'01' ALWAYS ZERO, INDICATES CNT

386+CNTFLAG2 DS    C CNT FLAG 2
387+CNT2UPP  EQU   X'80' UPPER CASE TRANSLATION REQUESTED
388+CNT2PAGE EQU   X'40' PAGED MESSAGE IN PROCESS
389+CNT2AERR EQU   X'20' SYSTEM ERROR MSG PLACED IN Q BY DFSCLMR0
390+*        EQU   X'10'        RESERVED FOR FUTURE USE
391+*        EQU   X'08'        RESERVED FOR FUTURE USE
392+*        EQU   X'04'        RESERVED FOR FUTURE USE
393+CNT2SUB  EQU   X'02' SUB POOL CNT
394+CNT2INQ  EQU   X'01' INQUIRY ONLY CNT

396+CNTCTBP  DS    H CTB POINTER
397+CNTCNTP  DS    H POINTER TO NEXT CNT ON SAME CTB
398+CNTCPMP  DS    H PASSWORD MATRIX POINTER
399+CNTCMPNT DS    C COMPONENT POINTER
```



```
400+          DS    CL3 RESERVED FOR FUTURE USE
```

```
 402          ISCD  SCDBASE=0
 403+SCD      DSECT
 404+*** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ***
 405+** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **
 406+**                                                                 **
 407+**             IMS SYSTEM CONTENTS DIRECTORY  (SCD)                **
 408+**                                                                 **
 409+** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **
 410+*** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ***


 412+SSCDORG  DS    0D SECONDARY SCD ORIGIN
 413+SSCDSVC1 SVC   0 USER TYPE 1 SVC NUMBER
 414+         BR    14 RETURN ON REGISTER 14
 415+SSCDSVC2 SVC   0 USER TYPE 2 SVC NUMBER
 416+         BR    14 RETURN ON REGISTER 14
 417+SSCDSVC3 SR    15,15 CLEAR REGISTER
 418+         BR    14 RETURN ON REGISTER 14
 419+SSCDSVC4 SVC   255 USER TYPE 4 SVC NUMBER
 420+         BR    14 RETURN ON REGISTER 14
 421+SSCDOCEP DC    CL2'Z9' OSAM CHANNEL END APPENDAGE SUFFIX
 422+SSCDOSPS DC    X'10' OS/360 SYSTEM CONFIGURATION
 423+SSCDOSLV DC    AL1(18) OS/360 RELEASE LEVEL SPECIFIED
 424+SSCDIMSR DC    AL1(0) IMS RELEASE NUMBER.
 425+SSCDIMSL DC    AL1(0) IMS MOD LEVEL NUMBER.
 426+SSCDASPS DC    X'00' SYSTEM CONFIGURATION - ACTUAL
 427+SSCDASLV DC    AL1(0) OS/360 RELEASE LEVEL - ACTUAL
 428+SSCDDESC DC    AL1(0) MCS DESCRIPTOR CODE
 429+         DC    AL3(0) RESERVED FOR OS/360 EXPANSION
 430+SSCDROUT DC    AL2(0) MCS ROUTING CODES
 431+         DC    AL2(0) RESERVED FOR OS/360 EXPANSION
 432+DFSXWAIT DC    V(DFSIWAIT) IWAIT EP ADDRESS
 433+DFSXIOBA DC    V(DFSIOS40) OSAM IOB QCB
 434+DFSXOPEN DC    A(DFSXOPEN) OSAM OPEN EP
 435+SSCDMCID DC    AL1(0) CONSOLE ID - START IMS COMMAND.
 436+         DC    AL3(0) RESERVED
 437+         DC    F'0' RESERVED
```

```
440+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
441+*                                                                         *
442+*        R E C O R D E R    L O G    S E C T I O N :                      *
443+*                                                                         *
444+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

446+SCDREENT DC    V(DFSFLOIO) RECORDER ENTRY POINT
447+SCDREDCB DC    A(0) DCB AND WKAREA LIST ADDRESS
448+SCDREWRK DC    V(DFSILOGW) WORK AREA FOR THE LOGGER
449+SCDREPLN DC    H'28' LENGTH OF LOG PREFIX IN BYTES

451+*** LOG CONTROL BYTE ***
452+SCDRECTL DC    X'0000'
453+SCDLGOPN EQU   X'80' LOG IS OPEN
454+SCDLGTRM EQU   X'40' LOG SMBTASK DETACHED

456+SCDRELCT DC    H'500' CHECKPOINT LOG FREQUENCY CONTROL REFERENCE
457+SCDRECCT DC    H'500' CURRENT LOG FREQUENCY COUNTER, CPT AT ZERO
458+SCDRPENT DC    A(0) ENTRY TO LOG WRITER
459+SCDRTECB DC    F'0' LOG TASK TERMINATE ECB
460+SCDRTCB  DC    A(0) LOG TASK CONTROL BLOCK ADDRESS
461+SCDRETXR DC    A(0) RESERVED
462+SCDRPFX  DC    V(DFSIPREF) ENTRY TO LOG PREFIX BUILDER
463+SCDDBLNT DC    A(SCDDBLNT) ENTRY TO DATABASE CHANGE LOGGING ROUTINE
464+SCDCWRK  DC    F'0' CHECKPOINT/DB LOG WORK AREA ADDRESS
465+SCDCWRKL DC    H'512' LENGTH OF CHECKPOINT/DB LOG WORK AREA
466+SCDCPNO  DC    H'0' CHECKPOINT NUMBER
467+SCDDATE  DC    F'0' DATE FOR RESTART/DATABASE LOG
468+SCDTIME  DC    F'0' TIME FOR RESTART/DATABASE LOG
469+SCDSEQ   DC    H'0' DATABASE LOG SEQUENCE NUMBER
470+SCDRGTYP DC    AL1(0,0) REGION TYPE (0=CTL,3=DLI)
471+SCDDBDCB DC    A(0) ADDRESS OF DATABASE LOG DCB
472+SCDDBLBA DC    F'0' ADDRESS OF BUFFER AREA
473+SCDDBWRP DC    F'0' ADDRESS OF DATA SET WRAPAROUND POINTER
474+SCDDBLBN DC    H'0' NUMBER OF BUFFERS
475+SCDLLOGL DC    H'0' LENGTH OF LOG AREA
476+SCDLLREA DC    F'0' REAL ADDRESS OF LOG WKAREA
```

```
STMT   SOURCE STATEMENT                                          F01MAY72   2/13/73
 478+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 479+*                                                                          *
 480+*  D A T A   L A N G U A G E / I   &   O S A M   S E C T I O N :  *
 481+*                                                                          *
 482+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

 484+SCDDLICT DC    A(SCDDLICT) DL/I ANALYZER ENTRY POINT (DFSDLA00)
 485+SCDDLIMQ DC    A(0) RESERVED
 486+SCDISAM  DC    A(SCDISAM) ISAM SIMULATOR ENTRY POINT
 487+SCDDLARE EQU   22 RETURN POINT FROM ANALYZER ENTRY POINT
 488+SCDDLIRE DC    A(SCDDLIRE) RETRIEVE MODULE ENTRY POINT
 489+SCDDLIIN DC    A(SCDDLIIN) INSERT MODULE ENTRY POINT
 490+SCDDLIDR DC    A(SCDDLIDR) DELETE/REPLACE MODULE ENTRY POINT
 491+SCDDLDAE EQU   18 ALTERNATE ENTRY TO DELETE/REPLACE
 492+SCDDLIHS DC    A(SCDDLIHS) HSAM ENTRY POINT
 493+SCDDLIPS DC    V(DFSIDIR0) BEGINNING OF PSB DIRECTORY ENTRIES
 494+SCDPSBFR DC    A(0) EP TP PSB FREE POOL SPACE ROUTINE
 495+SCDDLILN DC    H'40' LENGTH OF PSB DIRECTORY ENTRY
 496+SCDDLINO DC    H'0' NUMBER OF PSB DIRECTORY ENTRIES
 497+SCDPSBSW DS    0XL1 0 - FREE PSB POOL SPACE
 498+SCDDLIDM DC    A(0) BEGINNING OF DMB DIRECTORY ENTRIES
 499+SCDDLIDL DC    H'36' LENGTH OF EACH DMB DIRECTORY ENTRY
 500+SCDDLIDN DC    H'0' NUMBER OF DMB DIRECTORY ENTRIES
 501+SCDDLIPA DC    V(DFSIPST) START OF PST BLOCKS
 502+SCDDLIPL DC    H'448' LENGTH OF EACH PST
 503+SCDDLIPN DC    H'0' NUMBER OF PST ENTRIES
 504+SCDDLIBD DC    A(0)
 505+SCDDMBSW DS    0XL1 0 - RELEASE DMB SPACE, 1 - RESERVED
 506+SCDDMBFR DC    A(0) ENTRY TO DMB FREE POOL SPACE ROUTINE
 507+SCDPSBMU DC    V(DFSIPSBQ) PSB MOST USED QCB ADDRESS
 508+SCDDPDM  DC    V(DFSDPDM0) PSB/DMB POOL MANAGER
 509+SCDACBDC DC    V(ACBDCB) ACBLIB DCB POINTER
 510+SCDPSBPL DC    F'0'
 511+SCDDMBPL DC    F'0'
 512+SCDLLOGD DC    7F'0'
 513+SCDDLIMV DC    V(DFSDBLM0) BLOCK MOVER ENTRY POINT
 514+SCDDSST  DC    V(DFSDSST0) SEGMENT INTENT RESERVATION CSECT
 515+SCDDLICL DC    A(SCDDLICL) DATA MANAGEMENT OPEN/CLOSE ENTRY POINT
 516+SCDDLI07 DC    A(SCDDLI07) ENTRY TO DL/I LOAD PROGRAM
 517+SCDWQCB  DC    V(DFSPWQCB) DL/I MOVER QCB ADDRESS
 518+SCDBPARM DC    A(0) DL/I REGION PARMS (PXPARMS)
 519+SCDASE   DC    V(DFSIASE0) APPLICATION SCHEDULER TERMINATOR EP
 520+SCDDBFPL DS    A LOCATION OF DL/I BUFFER POOL
 521+SCDDDBH0 DC    A(0) ENTRY POINT OF DL/I BUFFER HANDLER
 522+SCDDHDS0 DC    A(0) ENTRY POINT OF GET/FREE SPACE
 523+SCDDXMT0 DC    A(0) ENTRY POINT OF INDEX MAINTENANCE
 524+SCDDBCVT DC    A(0) ENTRY POINT OF BYTE BLOCK CONVERSION
 525+SCDOIOBS EQU   DFSXIOBA OSAM IOB QCB
 526+SCDOOPEN EQU   DFSXOPEN OSAM OPEN EP
 527+SCDIWAIT EQU   DFSXWAIT IWAIT ENTRY POINT ADDRESS
 528+SCDSAVE  DC    V(DFSISAV0) ORIGIN OF SAVE AREA SET
 529+SCDFLOS  DC    A(0) STAE EXIT ROUTINE ENTRY POINT
 530+SCDFLOSR DC    A(0) STAE RETRY ROUTINE ENTRY POINT
 531+SCDNAVID DS    0F
 532+SCDRLDTE DC    X'00'
```

```
533+SCDSYNCN DC     AL3(0)
534+SCDLOWID DC     F'0'
535+SCDFIX   DC     F'0' PGFIX LIST CHAIN POINTER          8966
536+         DC     5F'0' RESERVED FOR SYSTEM EXPANSION     8966
```



```
538+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
539+*                                                                        *
540+*      S T O R A G E   M A N A G E M E N T   S E C T I O N :             *
541+*                                                                        *
542+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

544+SCDSMMLO DC     A(0) ADDRESS OF START OF SMM CORE AREA
545+SCDSMML  DC     F'0' LENGTH OF STORAGE MANAGEMENT CORE AREA
546+SCDSMMHI DC     A(0) ADDRESS OF END OF SMM CORE AREA
547+SCDSMMGC DC     A(0)
548+SCDSMMFC DC     A(0)
549+SCDSMMCP DC     V(DFSICRET) ICREAT ENTRY POINT
550+SCDSMMDP DC     V(DFSIDEST) IDESTROY ENTRY POINT
551+SCDSMMFB DC     V(DFSIFBUF) IFREEBUF ENTRY POINT
552+SCDSMMGB DC     V(DFSIGBUF) IGETBUF ENTRY POINT
553+SCDMFBPA DC     A(0) ADDRESS OF FORMAT BUFFER
554+SCDMFBPS DC     A(0) SIZE OF FORMAT BUFFER POOL
555+SCDFRENR DC     AL2(0) NUMBER OF FORMAT REQUEST ELEMENTS
556+SCDILMSK DC     X'0001' PROGRAM LOAD MASK
557+SCDBFPTR DC     V(DFSFXC20) USED BY DFSXXXX0                        8977


559+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
560+*                                                                        *
561+*         E N Q U E U E / D E Q U E U E   S E C T I O N :                *
562+*                                                                        *
563+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

565+SCDIDEQ  DC     V(DFSIIDEO) IDEQ ENTRY POINT
566+SCDIENQ  DC     V(DFSIIENO) IENQ ENTRY POINT
567+         DS     2F
```

```
569+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
570+*                                                                       *
571+*          Q U E U E   M A N A G E M E N T   S E C T I O N :
572+*                                                                       *
573+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

575+SCDIRWQE DC    V(DFSQMGRO) IREAD/IWRITE QUEUE ENTRY POINT
576+SCDIQDCB DC    V(DFSIQDCB) ADDRESS OF INPUT QCR DCB
577+SCDSMDCB DC    V(DFSSMDCB) ADDRESS OF IMS2.SHMSG DCB
578+SCDLMDCB DC    V(DFSLMDCB) ADDRESS OF IMS2.LGMSG DCB
579+SCDSPDCB DC    V(DFSICDCB) ADDRESS OF SCRATCH PAD AREA (SPA) DCB
580+SCDQPOOL DC    A(0) POINTER TO QUEUE BLOCK POOL
581+SCDPRLN1 DC    H'40' LENGTH OF FULL MESSAGE PREFIX
582+SCDPRLN2 DC    H'16' LENGTH OF CONDENSED MESSAGE PREFIX
583+SCDQCRSZ DC    H'0' SIZE OF QCR BUFFER
584+SCDMSGSZ DC    H'0' SIZE OF MSG BUFFER
585+SCDQMREB DC    F'0' SAVE AREA PTR TO BUFFER ALLOC FOR REUSE
586+SCDCKCL  DC    F'0' COMM FOR INTERNAL CHECKPOINTS
587+SCDQMSRB DC    H'0' RECORDS RESERVED FOR SHUTDOWN/REUSE
588+SCDPRDEF DC    AL1(128) PREFIX DEFINITION BYTE
589+SCDPDBS  EQU   X'80' BASIC PREFIX INFORMATION INCLUDED
590+SCDPDFNM EQU   X'40' FORMAT NAME INCLUDED
591+*        EQU   X'3F'               RESERVED
592+SCDPRADL DC    AL1(24,0,0,0,0,0,0,0) PREFIX OPTIONS LENGTHS
593+         DC    AL1(0) RESERVED
```

STMT SOURCE STATEMENT F01MAY72 2/13/73

```
595+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
596+*                                                                           *
597+*            C O M M U N I C A T I O N   S E C T I O N :                    *
598+*                                                                           *
599+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

601+SCDCTT    DC    V(DFSICTT) COMMUNICATION TRANSLATION TABLE BASE ADDR
602+SCDCTTL   DC    H'28' LENGTH OF EACH CTT
603+SCDCTTN   DC    H'0' NUMBER OF CTTS
604+SCDCLB    DC    V(DFSICLB) COMMUNICATION LINE BLOCK BASE ADDRESS
605+SCDCLBL   DC    H'96' LENGTH OF EACH CLB
606+SCDCLBN   DC    H'0' NUMBER OF CLBS
607+SCDCTB    DC    V(DFSICTB) COMMUNICATION TERMINAL BLOCK BASE ADDRESS
608+SCDCTBL   DC    H'32' LENGTH OF EACH CTB
609+SCDCTBN   DC    H'0' NUMBER OF CTBS
610+SCDCIB    DC    A(0) COMMUNICATION INTERFACE BLOCK BASE ADDR
611+SCDCIBL   DC    H'48' LENGTH OF EACH CIB
612+SCDCIBN   DC    H'0' NUMBER OF CIBS
613+SCDCRB    DC    A(0) COMMUNICATION RESTART BLOCK BASE ADDRESS
614+SCDCRBL   DC    H'20' LENGTH OF EACH CRB
615+SCDCRBN   DC    H'0' NUMBER OF CRBS
616+SCDCXB    DC    A(0) COMMUNICATION EXTENT BLOCK BASE ADDRESS
617+SCDCXBL   DC    H'20' LENGTH OF EACH CXB
618+SCDCXBN   DC    H'0' NUMBER OF CXBS
619+SCDCNT    DC    V(DFSICNT) COMMUNICATION NAME TABLE BASE ADDRESS
620+SCDCNTL   DC    H'44' LENGTH OF EACH CNT
621+SCDCNTN   DC    H'0' NUMBER OF CNTS
622+SCDCDB    DC    V(DFSICDB) COMMUNICATION DIRECTORY BLOCK BASE ADDR
623+SCDCDBL   DC    H'4' LENGTH OF A CDB
624+SCDCDBN   DC    H'0' NUMBER OF CDBS
625+SCDCVB    DC    V(DFSICVB) COMMUNICATION VERB BLOCK BASE ADDRESS
626+SCDCVBL   DC    H'12' LENGTH OF EACH CVB
627+SCDCVBN   DC    H'0' NUMBER OF CVBS
628+SCDCPT    DC    A(0) COMMUNICATION PRIORITY TABLE BASE ADDRESS
629+SCDCPTL   DC    H'0' LENGTH OF EACH CPT
630+SCDCPTN   DC    H'0' NUMBER OF CPTS
631+SCDCPM    DC    A(0) COMMUNICATION PRIORITY MATRIX BASE ADDR
632+SCDCPML   DC    H'0' LENGTH OF EACH CPM
633+SCDCPMN   DC    H'0' NUMBER OF CPMS
634+SCDCTM    DC    V(DFSICTM) COMMUNICATION TERMINAL MATRIX BASE ADDR
635+SCDCTML   DC    H'0' LENGTH OF EACH CTM
636+SCDCTMN   DC    H'0' NUMBER OF CTMS
637+SCDCCB    DC    A(0) CONVERSATIONAL CONTROL BLOCK BASE ADDRESS
638+SCDCCBL   DC    H'32' CCB LENGTH
639+SCDCCBN   DC    H'0' NUMBER OF CCBS
640+SCDCCBND  DC    AL2(0) OFFSET TO FIRST DISK CCB
641+SCDCCBNC  DC    AL2(0) OFFSET TO FIRST INCORE CCB
642+SCDCCBDL  DC    H'0' MAXIMUM LENGTH OF DISK SPA
643+SCDCCBCL  DC    H'0' MAXIMUM LENGTH OF INCORE SPA
644+SCDCAN01  DC    V(DFSCI001) COMMUNICATION ANALYZER ENTRY # 1
645+SCDCAN02  DC    V(DFSCI002) COMMUNICATION ANALYZER ENTRY # 2
646+SCDCAN03  DC    V(DFSCI003) COMMUNICATION ANALYZER ENTRY # 3
647+SCDCAN04  DC    V(DFSCI004) COMMUNICATION ANALYZER ENTRY # 4
648+SCDCAN05  DC    V(DFSCI005) COMMUNICATION ANALYZER ENTRY # 5
649+SCDCAN06  DC    V(DFSCI006) COMMUNICATION ANALYZER ENTRY # 6
```

```
650+SCDCAN07 DC     V(DFSCI007) COMMUNICATION ANALYZER ENTRY # 7
651+SCDCAN08 DC     V(DFSCI008) COMMUNICATION ANALYZER ENTRY # 8
652+SCDCAN09 DC     V(DFSCI009) COMMUNICATION ANALYZER ENTRY # 9
653+SCDCAN10 DC     V(DFSCI010) COMMUNICATION ANALYZER ENTRY # 10
654+SCDCAN11 DC     V(DFSCI011) COMMUNICATION ANALYZER ENTRY # 11
655+SCDCANOC DC     V(DFSCIOC0) COMMUNICATION ANALYZER SERVICE ROUTINE
656+SCDCANOD DC     V(DFSCIOD0) COMMUNICATION ANALYZER SERVICE ROUTINE
657+SCDCPCUR DC     A(0) FORMAT EDITIOR I/OP CURRENT
658+SCDCPHWM DC     A(0) FORMAT EDITIOR I/OP HI WATER MARK
659+SCDFNDST DC     V(DFSICLF0) FIND DESTINATION ROUTINE ENTRY POINT
660+SCDMSGRT DC     V(DFSICLR0) MESSAGE ROUTER ROUTINE ENTRY POINT
661+SCDMTRM  DC     V(DFSCTBMT) MASTER TERMINAL CTB ADDRESS
662+SCDCLISO DC     V(DFSICLS0) SECURITY ROUTINE ENTRY POINT
663+SCDTRANS DC     V(DFSICLT0) TRANSLATE ROUTINE ENTRY POINT
664+SCDCONV  DC     A(0) CONVERSATION PROCESSOR ENTRY POINT
665+SCDCOB   DC     A(0) COMMUNICATION OVERFLOW BLOCK BASE ADDRESS
666+SCDCOBL  DC     H'32' LENGTH OF EACH COB
667+SCDCOBN  DC     H'0' NUMBER OF COBS
668+SCDCTCLB DC     V(DFSICLB) CONSOLE TYPEWRITER CLB
669+SCDREPOL DC     V(DFSICLX0) RESET POLL MODULE ENTRY POINT
670+SCDNTB   DC     V(DFSINTB0)
671+SCDEDITC DC     A(0) USER EDIT ROUTINE FOR CNT DESTINATIONS
672+SCDEDITS DC     V(DFSLIST0) USER EDIT ROUTINES FOR SMB DESTINATIONS
673+SCDINCT  DC     H'0'
674+SCDOUTCT DC     H'0'
675+SCDCPCTL DC     X'00' CHECKPOINT POST SYNC BITS
676+SCDPCP01 EQU    X'80'
677+SCDPCP02 EQU    X'40'
678+SCDPCP03 EQU    X'20'
679+SCDCTR01 DC     AL1(0)
680+SCDCTRL1 EQU    X'80' .    TRACE BLOCKS LEVEL 1
681+SCDCTRL2 EQU    X'40' .    TRACE BLOCKS LEVEL 2
682+SCDCTRL3 EQU    X'20' .    TRACE WORK AREAS
683+SCDCTRL4 EQU    X'10' .    TRACE SAVE SETS
684+SCDCTR02 DC     AL1(0)
685+SCDCTRA  EQU    X'80' .    TRACE ALL POINTS
686+SCDCTRF  EQU    X'40' .    TRACE DEVICE INFO
687+         DC     AL1(0) RESERVED FOR SYSTEM EXPANSION
688+         DC     F'0' RESERVED FOR SYSTEM EXPANSION
```

```
690+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
691+*                                                                          *
692+*            D I S P A T C H E R   S E C T I O N :                         *
693+*                                                                          *
694+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

696+SCDEXTWQ DC    V(DFSEXTWQ) EXTERNAL WAIT QUEUE ADDRESS
697+SCDINTWQ DC    V(DFSINTWQ) INTERNAL WAIT QUEUE ADDRESS
698+SCDINTSQ DC    V(DFSINTSQ) INTERNAL SAVE QUEUE ENTRY POINT ADDRESS
699+SCDTECBR DC    V(DFSITERM) TERMINATION ROUTINE ENTERED IF TECB POST
700+SCDPARM  DC    A(0) ADDRESS OF PARM LIST FROM SYSTEM
701+SCDTECB  DC    F'0' IMS TERMINATION ECB POSTED BY CHECKPOINT
702+SCDSAVEL DC    H'508' LENGTH OF SAVE AREA SETS INCLUDING QE
703+SCDSLEV  EQU   7 NUMBER OF SAVE AREAS IN A SET
704+SCDSAVEN DC    H'0' NUMBER OF SAVE AREA SETS
705+SCDSRQE  DC    V(DFSIRQE0) ORIGIN OF RQE LIST
706+SCDSRQEL DC    H'8' LENGTH OF EACH RQE
707+SCDSRQEN DC    H'0' NUMBER OF RQES
708+SCDWAITQ DC    V(DFSIWATQ) ORIGIN OF WAIT QUEUE
709+SCDEXTWL DC    H'4' LENGTH OF EACH ENTRY IN EXT WAIT QUEUE
710+SCDEXTWN DC    H'0' NUMBER OF EXTERNAL EVENTS
711+SCDSUBSV DC    A(0) ADDRESS OF LAST SUBTASK 1ST LEV SAVE AREA
712+SCDCFAC  DC    X'00' CURRENT FIRST LEVEL FACILITY
713+SCDCMOD  DC    X'00' CURRENT MODULE WITHIN FACILITY
714+SCDTEST  DC    X'80' TEST MODE CONTROL INDICATOR
715+SCDTLEV  DC    X'00' TEST MODE LEVEL INDICATOR
716+SCDRES2  DC    F'0' PADDING
717+         DC    4F'0' RESERVED FOR SYSTEM EXPANSION
```

```
719+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
720+*                                                                         *
721+*   A P P L I C A T I O N   S C H E D U L E R   S E C T I O N :           *
722+*                                                                         *
723+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

725+SCDASINT DC    V(DFSASINT) APPLICATION SCHEDULER INITIATOR ENTRY
726+SCDASTER DC    V(DFSASTRM) APPLICATION SCHEDULER TERMINATOR ENTRY
727+SCDSMBEP DC    V(DFSISMB) SYSTEM MESSAGE BLOCKS BASE ADDRESS
728+SCDSMBL  DC    H'56' LENGTH OF EACH SMB
729+SCDSMBN  DC    H'0' NUMBER OF SMBS
730+SCDSMBDQ DC    V(DFSSMBDQ) SMB DEQUEUE ENTRY POINT ADDRESS
731+SCDSMBNQ DC    V(DFSSMBEQ) SMB ENQUEUE ENTRY POINT ADDRESS
732+SCDCHKXB DC    A(SCDCHKEB) ADDRESS OF ECB TO BE POSTED BY DFSIAS00
733+SCDTCTEP DC    V(DFSITCT) TRANSACTION CLASS TABLES BASE ADDRESS
734+SCDTCTL  DC    H'80' LENGTH OF EACH TCT
735+SCDTCTN  DC    H'0' NUMBER OF TCTS
736+SCDMRQEP DC    V(DFSIMRQ) ADDRESS OF MESSAGE REQUEST QUEUE
737+SCDPRQEP DC    V(DFSIPRQ) ADDRESS OF PARTITION REQUEST QUEUE
738+SCDBPRQE DC    V(DFSIBPRQ) ADDRESS OF BATCH PARTITION REQUEST QUEUE
739+SCDEXTSQ DC    V(DFSEXTSQ) EXTERNAL SAVE QUEUE ENTRY POINT ADDRESS
740+SCDPCBSZ DC    A(0) MAX SIZE OF PCB AREA IN MSG REGION
741+         DC    3F'0' RESERVED
```

```
STMT   SOURCE STATEMENT                                              F01MAY72   2/13/73

 743+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 744+*                                                                       *
 745+*   C H E C K P O I N T / R E S T A R T    S E C T I O N :              *
 746+*                                                                       *
 747+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

 749+*** CHECKPOINT STATUS BYTE ***
 750+SCDCKCTL DC     X'00'
 751+*                        DESCRIPTION OF BIT SETTINGS
 752+*                             0  NOT USED
 753+*                             1  DATABASE RECOVERY
 754+*                             2  PURGE REQUEST
 755+*                             3  SYSTEM SHUTDOWN REQUEST
 756+*                             4  DUMP QUEUE REQUEST
 757+*                             5  MASTER TERMINAL CHECKPOINT REQUEST
 758+*                             6 DATABASE DUMP REQUEST
 759+*                             7  SYSTEM SCHEDULED CHECKPOINT (LOG)

 761+*** RESTART STATUS BYTE ***
 762+SCDRSCTL DC     X'00'
 763+*                        DESCRIPTION OF BIT SETTINGS
 764+*                             0  PASSWORD SECURITY REQUESTED
 765+*                             1  QUEUE ONLY RESTART
 766+*                             2  PSB/DMB CHANGED - REBUILD BLOCKS
 767+*                             3  ALLOW COMMANDS EXCEPT RESTART
 768+*                             4  TERMINAL SECURITY REQUESTED
 769+*                             5  EMERGENCY RESTART
 770+*                             6 BUILD QUEUES
 771+*                             7  NORMAL RESTART

 773+*** COMMON SYSTEM SHUTDOWN STATUS BYTE ***
 774+SCDSTOP1 DC     X'0000'
 775+*              BYTE 1    DESCRIPTION OF BIT SETTINGS - USER
 776+*                        0  STOP INPUT                COMMUNICATIONS
 777+*                        1  STOP OUTPUT               COMMUNICATIONS
 778+*                        2  SEND ALL OUTPUT           COMMUNICATIONS
 779+*                        3  PROCESS ALL MESSAGES      SCHEDULER
 780+*                        4  QUEUE'S AVAILABLE         READ/WRITE Q
 781+*                        5  FREE MESSAGE REGIONS      SCHEDULER
 782+*                        6  TERMINATE ALL REGIONS     SCHEDULER
 783+*                        7  STOP SMB INPUT            COMMUNICATIONS
 784+*
 785+*              BYTE 2    DESCRIPTION OF BIT SETTINGS - USER
 786+*                        0  PROCESS AT SYSTEM PRIORITY SCHEDULER
 787+*                        1  DATABASE STOP              CHECKPOINT
 788+*                        2  FORCE EOV ON LOG FILE      CHECKPOINT
 789+*                        3  ABEND AT TERMINATION       COMMUNICATIONS
 790+*                        4  DISCONNECT ALL LINES       COMMUNICATIONS
 791+*                        5  ALL OUTPUT SENT            COMMUNICATIONS
 792+*                        6  SYSTEM CLOSE DOWN          CHECKPOINT
 793+*                        7  FINAL ENTRY CONTROL        CHECKPOINT

 795+SCDCKDBN DS     OCL1 # OF ENTERED DATABASE NAMES FOR RECOVERY
 796+SCDCKDB  DC     A(0) POINTER TO POOL CONTAINING DATABASE NAMES
 797+SCDLOGN  DS     OCL1 COUNT OF RESTART SERIAL NUMBERS
```

```
 798+SCDLOG    DC    A(0) POINTER TO POOL CONTAINING SERIAL NUMBERS
 799+SCDCLIPN DC    H'0000' INPUT TERMINAL ACTIVE COUNT
 800+SCDCLOPN DC    H'0000' OUTPUT TERMINAL ACTIVE COUNT
 801+SCDCPC01 DC    A(0) SCRATCH PAD AREA LOG ROUTINE ENTRY POINT
 802+SCDCHKEP DS    0F
 803+SCDRSTEP DC    V(DFSRST00) RESTART ROUTINE ENTRY POINT ADDRESS
 804+SCDCHKEB DS    0F
 805+SCDRSTEB DC    V(DFSRTECB) RESTART ECB ADDRESS
 806+SCDDBTBL DC    A(0) RESTART DATABASE NAME TABLE
 807+          DC    4F'0'
```



```
 809+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 810+*                                                                       *
 811+*      S T A R T / S T O P   R E G I O N   S E C T I O N :              *
 812+*                                                                       *
 813+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

 815+SCDREGOP DC    A(0) ECB TO OPEN REGION
 816+SCDREGCL DC    A(0) ECB TO CLOSE REGION
 817+SCDCMDCT DC    X'00' # OF START/STOP REGION COMMANDS ISSUED
 818+SCDREGST DC    X'00' # OF PENDING STOP COMMANDS
 819+SCDREGCT DC    H'0000' COUNT OF OPEN REGIONS
 820+SCDSRDR  DC    A(SCDPROC1) ADDRESS OF START READER COMMAND
 821+SCDSRMBR DC    A(SCDPROC2) ADDRESS OF MEMBER FIELD IN START READER
 822+SCDSMMBR DC    A(SCDPROC3) ADDRESS OF MSG REG PROC MEMBER NAME
 823+SCDSINIT DC    A(SCDPROC4) ADDRESS OF START INITIATOR COMMAND
 824+SCDPINIT DC    A(SCDPROC5) ADDRESS OF STOP INITIATOR COMMAND
 825+          DC    2F'0'
 826+SCDPSYS  EQU   SSCDOSPS PROGRAMMING SYSTEM OPTION BYTE FROM CVT
 827+SCDPSMVT EQU   X'10' VALUE IF MVT
 828+SCDPSMFT EQU   X'20' VALUE IF MFT-II
 829+SCDPSPCP EQU   X'40' VALUE IF PCP
 830+SCDPSREL EQU   X'02' VALUE IF AOS
 831+SCDIMSR  EQU   SSCDIMSR IMS RELEASE NUMBER
 832+SCDIMSL  EQU   SSCDIMSL IMS MOD LEVEL
 833+          DC    2F'0' RESERVED FOR SYSTEM EXPANSION
```

```
STMT   SOURCE STATEMENT                                                   F01MAY72    2/13/73

 835+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 836+*                                                                         *
 837+*        C O M M A N D    S E C T I O N :                                 *
 838+*                                                                         *
 839+*                   *** NOTE ***                                          *
 840+*                         THE COMMAND SECTION MUST RESIDE LAST            *
 841+*                         IN THE SYSTEM CONTENTS DIRECTORY SINCE          *
 842+*                         THE COMMANDS ARE VARIABLE IN LENGTH.            *
 843+*                                                                         *
 844+*                         ANY ADDITION TO THE SCD                         *
 845+*                         MUST PRECEDE THIS SECTION.                      *
 846+*                                                                         *
 847+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

 849+SCDPROC1 DS     OF
 850+         DC     AL2(17) TEXT LENGTH
 851+         DC     B'0000000000000000' MCS FLAGS
 852+         DC     C'S IMSRDR,MBR='
 853+SCDPROC2 DS     CL9 MEMBER NAME FIELD OF START READER COMMAND
 854+SCDPROC3 DC     AL1(6) LENGTH IN BYTES OF PROC MEMBER NAME
 855+         DC     C'IMSMSG' MSG PROC MEMBER NAME
 856+SCDPROC4 DS     OF
 857+         DC     AL2(26) TEXT LENGTH
 858+         DC     B'0000000000000000' MCS FLAGS
 859+         DC     C'S INIT.CLASSB,,,(B,A) '
 860+SCDPROC5 DS     OF
 861+         DC     AL2(12) TEXT LENGTH
 862+         DC     B'0000000000000000' MCS FLAGS
 863+         DC     C'P CLASSB'
 864+         DS     4F

 866                *,*****************************************************
 867                *,*                IMS SYSTEM
 868                *,*      VERSION 2   RELEASE 3   MOD LEVEL 0      *
 869                *,*****************************************************

 872          END
```

PHYSICAL TERMINAL (OUTPUT) EDITS

The user can also specify at system definition time a physical terminal output edit routine to edit output messages just before they are sent to a terminal. One physical terminal output routine may be specified for each BTAM telecommunication line group. During system definition, the user specifies which physical terminals, within each line group, use the defined edit routine for output editing. These edit routines may be used to provide special user editing needs by communciation terminal types. An output message may be processed by a physical terminal output edit routine and the basic IMS/360 edit routine on the Message Format Service. Output editing is performed in this sequence. Therefore if the Message Format Service is used the output provided by the edit routine must be the format defined to the Message Format Service instead of the format created by the application program.

Upon entry to a user-supplied physical terminal output edit routine, the following interface applies:

Register Number	Content
1	The address of a buffer containing the output message segment to be edited. The first two bytes are a binary count of message segment length. The second two bytes are control information provided by the application program which constructed the message. The text of the output message starts in byte five. The count includes the first four bytes in length.
7	The address of the output physical terminal block (CTB) for the destination terminal.
9	The address of the IMS/360 control block describing the output communication line (CLB). This block starts with a BTAM DECB. The content of DECAREA field in the DECB is equivalent to Register 1 content.
11	IMS/360 System Contents Directory (SCD) address.
13	The address of a save area for use by the edit routine. All registers must be saved upon entry and restored upon exit. The first three words in the save area may not be changed.
14	The address by which the edit routine should return to IMS/360.
15	The entry point address to the invoked edit routine.

The resultant output message segment returned to IMS/360 from the user's edit routine must be pointed to by the content of the BTAM DECB, DECAREA field. The first four bytes must be in a format as received at input with the binary count updated to the edited message segment length inclusive of the four bytes of prefix. Upon return to IMS/360, all registers must be restored. If the message is to be edited in place, the length may not be increased by more than 10 bytes.

When the last segment of a message has been edited, IMS/360 returns control to the user's edit routine once more. The edit routine may wish to perform some housekeeping activities at this time. Upon entry to the user's edit routine, Registers 7, 9, 11, 12, 13, 14, and 15 are as described above.

Whenever a physical terminal output edit is invoked, the Communication Terminal Block is addressed by Register 7. A field in the block, CTBACTL, which is one byte in length, will contain a one in the second bit position if this entry to the user's edit routine is for end of message. The reader is referred to the IMS/360 System Manual for definition of IMS/360 control blocks.

EXAMPLE OF PHYSICAL TERMINAL EDIT

The following example illustrates how any output message can be extended in length and a prefix attached. Two capabilities within IMS/360 are used. One allows the edit routine to obtain a buffer area. This is called ICREATE. When ICREATE is used, an identifier of four bytes is provided in Register 2. The length of the requested area is placed in Register 3. The address of the buffer area is returned to the edit routine in Register 3. This area is used to build the output message. The prefix '*IMS*' is added to the message. The edited output message is addressed by DECAREA. When the end of message entry to the edit routine is made (CTBACTL=40), the buffer area obtained by the edit routine is returned to IMS/360. This is performed by the second IMS/360 facility called IDESTROY. Register 2 is used to symbolically identify the area to IMS/360. This example applies to single-line messages only and to only one terminal at a time. As the addition of '*IMS*' did not sequence the extension of the message by more than 10 bytes, it could have been performed in place without the creation of an additional buffer area. The following is an example of how this could be accomplished.

STMT   SOURCE STATEMENT

```
DFSEDCTO CSECT
*  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
*  NOTE THIS MODULE IS NOT RE ENTRANT AND MAY NOT BE USED BY MORE
       THAN ONE TERMINAL AT A TIME. TO MODIFY IT TO ALLOW USE BY
*    MULTIPLE TERMINALS THE TABLE MUST BE EXPANDED TO THE NUMBER OF
*    USING TERMINALS AND A UNIQUE ID FOR EACH MUST BE USED FOR THE
*    ICREATE OF THE BUFFER.
*  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
         SAVE  (14,12),,DCT1111     SAVE ENTRY REGISTERS
         B     12(0,15) BRANCH AROUND ID
         DC    AL1(7) LENGTH OF IDENTIFIER
         DC    CL7'DCT1111' IDENTIFIER
         STM   14,12,12(13) SAVE REGISTERS
         L     R13,8(R13)           POINT TO NEXT SAVAREA
         LR    R12,R15              LOAD AND
         USING DFSEDCTO,R12         .ESTABLISH MODULE BASE
         USING IECTDECB,R9          CLB POINTERS
         USING CTB,R7               CTB POINTERS
         USING SCD,R11              SCD POINTERS
*----------------------------------------------------------------------*
*        GET AREA FOR EDIT IF FIRST SEGMENT                            *
*----------------------------------------------------------------------*
         BAL   R5,DESTROY     DELETE THE USED BUFFER
         TM    CTBACTL,CTBAEOM        END OF MSG
         BO    RETURN          YES
         CLI   DECTYPE,X'02'       1ST SEGMENT?
         BE    RETURN          NO
CREATE   EQU   *
         L     R5,DECAREA           YES,POINT TO MESSAGE
         SR    R3,R3
         IC    R3,0(R5)             LOAD MSG DL (LENGTH)
         SLL   R3,8                 .
         IC    R3,1(R5)             .
         LR    R6,R3                SAVE MSG LENGTH
         LA    R3,9(R3)             ALLOW FOR INCR MSG LENGTH
         LR    R5,R3               SAVE NEW DL
*ICREATE                            SETUP FOR ICREATE
         LR    R0,R11               POINT TO SCD
         LR    R1,R9                POINT TO CLB
         L     R2,ID                LOAD ID
         SR    R4,R4                ZERO R4
         L     R15,SCDSMMCP         POINT TO ICREATE
         BALR  R14,R15
*                                   STORE ID IN TABLE:
*                                   ID WOULD INCL. CTB # AND WOULD
*                                   BE STORED IN THE TABLE POSITION
*                                   CORRESPONDING TO THAT #.
         ST    R2,TABLE             STORE ID
*----------------------------------------------------------------------*
*        EDIT MESSAGE INTO AREA                                        *
*----------------------------------------------------------------------*
         STH   R5,0(R3)             SETUP NEW DL00
         XC    2(2,R3),2(R3)        .
         MVC   4(9,R3),EDMSG        MOVE IN EDITING
         SH    R6,=H'5'             CALC LENGTH OF MOVE
```

```
STMT   SOURCE STATEMENT

            L      R5,DECAREA          PT TO OLD MSG
            EX     R6,MOVE              MOVE MSG LESS DLOO
            B      ARCUND
   MOVE     MVC    13(0,R3),4(R5)
   AROUND   EQU    *
            ST     R3,DECAREA           SETUP PTR TO NEW MSG AREA
            B      RETURN
   *-----------------------------------------------------------------------*
   *        RELEASE EDIT AREA IF NECESSARY                                 *
   *-----------------------------------------------------------------------*
   DESTROY  EQU    *
   *                                    CHECK APPRCPRIATE TABLE ENTRY
            CLC    TABLE,=F'0'          HAS AREA BEEN RELEASED?
            BCR    8,R5      YES- BRANCH
   *                                    SETUP IDESTROY
            LR     RC,R11               POINT TO SCD
            LR     R1,R9                POINT TO CLB
            L      R2,TABLE             LOAD ID
            L      R15,SCDSMMDP
            BALR   R14,R15
   *
            XC     TABLE(4),TABLE       ZERO TABLE ENTRY
            BR     R5
   *-----------------------------------------------------------------------*
   *        RETURN ROUTINE                                                 *
   *-----------------------------------------------------------------------*
   RETURN   EQU    *
            L      R13,4(R13)
            LM     0,12,20(R13)
            L      R14,12(R13)
            BCR    15,R14
   *

   TABLE    DC     F'0'                 ID TABLE
   ID       DC     C'EDIT'             AREA ID
   EDMSG    DC     C'    IMS            EDIT MESSAGE
   R1       EQU    1
   RC       EQU    0
   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
            LTORG
                   =F'0'
                   =H'5'
```

```
STMT   SOURCE STATEMENT

 110           ICLI  CLBBASE=0,CTBBASE=0
 111+*********************************************************************
 112+*
 113+*         CLB - COMMUNICATION LINE BLOCKS DSECT. DSECT NAME IS IECTDECB.
 114+*
 115+*********************************************************************
 116+IECTDECB DSECT  DECB DUMMY SECTION
 117+*         ----------------------------------------
 118+*         +                                      +
 119+*     0   +       STANDARD EVENT CONTROL BLOCK    +
 120+*         +                                      +
 121+*         +--------------------------------------+
 122+*         +                  +                   +
 123+*     4   +  OPERATION TYPE  +    AREA LENGTH    +
 124+*         +                  +                   +
 125+*         +--------------------------------------+
 126+*         + ON-LINE +                            +
 127+*     8   +TERMINAL +       ADDRESS OF DCB       +
 128+*         +  TEST   +                            +
 129+*         +--------------------------------------+
 130+*         +         +                            +
 131+*    12   +RESERVED +       ADDRESS OF AREA      +
 132+*         +         +                            +
 133+*         +--------------------------------------+
 134+*         +                  +                   +
 135+*    16   +    SENSE BYTES   +   RESIDUAL COUNT  +
 136+*         +     1    &   2   +                   +
 137+*         +--------------------------------------+
 138+*         +         +                            +
 139+*    20   + COMMAND |   ADDRESS OF TERMINAL LIST  +
 140+*         +   CODE  +                            +
 141+*         +--------------------------------------+
 142+*         +         + RELATIVE+         +        +
 143+*    24   +  STATUS +   LINE  + ADDRESS + VRC/LRC +
 144+*         +  FLAGS  +  NUMBER + RESPONSE+ RESPONSE+
 145+*         +--------------------------------------+
 146+*         +         +         +                  +
 147+*    28   +  TP-OP  +  ERROR  +     CSW STATUS   +
 148+*         +   CODE  +  STATUS +                  +
 149+*         +--------------------------------------+
 150+*         +         +                            +
 151+*    32   +RESERVED +     ADDRESS OF CURRENT     +
 152+*         +         +      ADDRESSING ENTRY      +
 153+*         +--------------------------------------+
 154+*         +         +                            +
 155+*    36   +RESERVED +     ADDRESS OF CURRENT     +
 156+*         +         +        POLLING ENTRY       +
 157+*         +--------------------------------------+
 158+*         +         +         +                  +
 159+*    40   +RESERVED +RESERVED +  WRITE AREA LENGTH +
 160+*         +         +         +                  +
 161+*         +--------------------------------------+
 162+*         +         +                            +
 163+*    44   +RESERVED +   ADDRESS OF WRITE AREA    +
 164+*         +         +                            +
 165+*         ----------------------------------------
```

```
    1          ICLI   CLBBASE=C,CNTBASE=0,CTBBASE=0,COBBASE=0,CTTBASE=0
    2+**********************************************************************
    3+*
    4+*        CLB - COMMUNICATION LINE BLOCKS DSECT. DSECT NAME IS IECTDECB.
    5+*
    6+**********************************************************************
    7+IECTDECB DSECT  DECB DUMMY SECTION
    8+*        -------------------------------------------
    9+*        +                                         +
   10+*    0   +       STANDARC EVENT CONTROL BLOCK      +
   11+*        +                                         +
   12+*        +-----------------------------------------+
   13+*        +                    +                    +
   14+*    4   +  OPERATION TYPE    +     AREA LENGTH    +
   15+*        +                    +                    +
   16+*        +-----------------------------------------+
   17+*        + ON-LINE  +                              +
   18+*    8   +TERMINAL  +        ADDRESS OF DCB        +
   19+*        +  TEST    +                              +
   20+*        +-----------------------------------------+
   21+*        +          +                              +
   22+*   12   +RESERVED  +        ADDRESS OF AREA       +
   23+*        +          +                              +
   24+*        +-----------------------------------------+
   25+*        +                    +                    +
   26+*   16   +    SENSE BYTES     +   RESIDUAL COUNT   +
   27+*        +    1     &    2    +                    +
   28+*        +-----------------------------------------+
   29+*        +          +                              +
   30+*   20   + COMMAND  |    ADDRESS OF TERMINAL LIST  +
   31+*        +   CODE   +                              +
   32+*        +-----------------------------------------+
   33+*        +          + RELATIVE+          +         +
   34+*   24   +  STATUS  +   LINE  + ADDRESS + VRC/LRC +
   35+*        +  FLAGS   +  NUMBER + RESPONSE+ RESPONSE+
   36+*        +-----------------------------------------+
   37+*        +          +         +                    +
   38+*   28   +  TP-OP   +  ERROR  +     CSW STATUS     +
   39+*        +   CODE   +  STATUS +                    +
   40+*        +-----------------------------------------+
   41+*        +          +                              +
   42+*   32   +RESERVED  +      ADDRESS OF CURRENT      +
   43+*        +          +       ADDRESSING ENTRY       +
   44+*        +-----------------------------------------+
   45+*        +          +                              +
   46+*   36   +RESERVED  +      ADDRESS OF CURRENT      +
   47+*        +          +         POLLING ENTRY        +
   48+*        +-----------------------------------------+
   49+*        +          +         +                    +
   50+*   40   +RESERVED  +RESERVED +  WRITE AREA LENGTH +
   51+*        +          +         +                    +
   52+*        +-----------------------------------------+
   53+*        +          +                              +
   54+*   44   +RESERVED  +    ADDRESS OF WRITE AREA     +
   55+*        +          +                              +
```



```
   56+*        -------------------------------------------
```

```
  58+DECSDECB DS     1F STATUS FLAG + ADDRESS OF THE TCB

  60+DECTYPE  DS     1H OPERATION TYPE

  62+DECLNGTH DS     1H AREA LENGTH

  64+DECONLTT DS     0CL1 RESERVED FOR ON-LINE TERMINAL TEST
  65+DECDCBAD DS     1F ADDRESS OF DCB

  67+DECAREA  DS     1F ADDRESS OF AREA

  69+DECSENS0 DS     1C 1ST SENSE BYTE

  71+DECSENS1 DS     1C 2ND SENSE BYTE

  73+DECCOUNT DS     1H RESIDUAL COUNT

  75+DECCMCOD DS     0CL1 COMMAND CODE
  76+DECENTRY DS     1F ADDRESS OF TERMINAL LIST


  79+DECFLAGS DS     1C STATUS FLAGS

  81+DECRLN   DS     1C RELATIVE LINE NUMBER

  83+DECRESPN DS     1H RESPONSE FIELDS

  85+DECTPCOD DS     1C TP-OP CODE

  87+DECERRST DS     1C ERROR STATUS

  89+DECCSWST DS     1H CSW STATUS

  91+DECADRPT DS     1F ADDRESS OF CURRENT ADDRESSING ENTRY

  93+DECPOLPT DS     1F ADDRESS OF CURRENT POLLING ENTRY

  95+         DS     2C RESERVED

  97+DECWLNG  DS     1H WRITE AREA LENGTH

  99+DECWAREA DS     1F ADDRESS OF WRITE AREA

 101+CLBDECB  EQU    DECSDECB
 102+CLBRCNTR DS     1C RETRY COUNTER                                    $

 104+CLBFLAG1 DS     C .          BIT USAGE
 105+CLB1COB  EQU    X'80' .    0 - COB REQUIRED FOR CTB'S ON THIS LINE
 106+CLB1TEST EQU    X'40' .    1 - TEST RECYCLE REQUEST
 107+CLB1IBUF EQU    X'20' .    2 - INPUT BUFFER ALLOCATED
 108+CLB1OBUF EQU    X'10' .    3 - OUTPUT BUFFER ALLOCATED
 109+*        EQU    X'08' .    4 - RESERVED FOR FUTURE USE
 110+CLB1QCRI EQU    X'04' .    5 - QCR ALLOCATED   (INPUT)              $
 111+CLB1RESP EQU    X'02' .    6 - RESPONSE WAITING
 112+CLB1QCRO EQU    X'01' .    7 - QCR ALLOCATED    (OUTPUT)            $
```

```
114+CLBFLAG2 DS     C
115+CLB2NOIN EQU    X'80' .     0 - NO INPUT ALLOWED
116+CLB2NOOU EQU    X'40' .     1 - NO OUTPUT ALLOWED
117+CLB2NOQU EQU    X'20' .     2 - DO NOT QUEUE ON THIS LINE
118+*        EQU    X'10' .     3 - RESERVED FOR FUTURE USE
119+CLB2IDLE EQU    X'08' .     4 - IDLE LINE
120+CLB2INP  EQU    X'04' .     5 - INPUT IN CONTROL
121+CLB2OUTP EQU    X'02' .     6 - OUTPUT IN CONTROL
122+CLB2AUTO EQU    X'01' .     7 - AUTOPOLL

124+CLBFLAG3 DS     C .
125+CLB3INP  EQU    X'80' .     0 - 0=OUTPUT NEXT, 1=INPUT NEXT ON THIS LINE
126+CLB3CSUB EQU    X'40' .     1 - COMMAND ENTERED AS NON-FIRST SEGMENT
127+CLB3DOPN EQU    X'20' .     2 - LINE IS OPENED DYNAMICALLY
128+CLB3STOP EQU    X'10' .     3 - LINE HAS BEEN IDLED FOR SHUTDOWN
129+CLB3CBUF EQU    X'08' 4 - CONDENSED BUFFER ALLOCATED
130+CLB3ICMD EQU    X'04' 5 - INTERNAL COMMAND IN PROGRESS
131+*        EQU    X'02' .     6 - RESERVED FOR FUTURE USE
132+*        EQU    X'01' .     7 - RESERVED FOR FUTURE USE

134+CLBCNTPT DS     F . POINTER TO CNT FOR RESPONSE                            $
135+CLBTEMP1 DS     F .
136+CLBTEMP4 DS     F .                 USED FOR TEMPORARY STORAGE
137+CLBTEMP5 DS     F .                 USED FOR TEMPORARY STORAGE

139+CLBCSID  DS     0C . FIND DEST FIELD                                       $
140+*                         0     DL/1 USE
141+*                         1     RESTART IN PROGRESS
142+CLBCSCVB EQU    X'20' .   2      CVB FOUND AS DESTINATION
143+*                         3     ALWAYS 0 FOR CLB
144+CLBCSFND EQU    X'08' .   4     CNT/SMB NOT FOUND
145+CLBCSSMB EQU    X'04' .   5 = 0  CNT FOUND
146+*                           = 1 SMB FOUND
147+CLBCSDST EQU    X'02' .   6      FNDDST ENTERED (CLBCNTP POINTS TO DEST)
148+CLBCSACT EQU    X'01' .   7     ACTIVE POLLING LIST

150+CLBPOLAD DS     F .  POLLING LIST ADDRESS
151+CLBCTBPT DS     H .  FIRST CTB ON LINE
152+CLBCRCTB DS     H .  CURRENT CTB
153+CLBCNTQB DS     2F . QCB FOR CNTS FOR OUTPUT
154+CLBSMBPT DS     F .  POINTER TO BLOCK FOR QUEUEING CNT/SMB/CVB            $
155+CLBCTBPS DS     H .
156+CLBDPOLC DS     C
157+CLBDCTL  DS     C .                 CLB DEVICE CONTROL BYTE (USED BY DDM)
158+CLBINBUF DS     F .          INPUT BUFFER POINTER
159+CLBOUTBF DS     F .          OUTPUT BUFFER POINTER
160+*
161+* IF ($) FALLS IN COLUMN 71 THIS FIELD WILL BE SAVED IN A COB IF REQ
```

```
199+************************************************************************
200+*
201+*          CTB - COMMUNICATION TERMINAL BLOCKS DSECT.
202+*
203+************************************************************************
204+CTB      DSECT
205+CTBTYPE  DS     C .        CTT NUMBER
206+CTBLINE  DS     C .        LINE NUMBER
207+CTBTERM  DS     3C .       TERMINAL ADDRESS

209+CTBFLAG1 DS     C .
210+CTB1CONV EQU    X'80' .  0 THIS CTB IN CONVERSATION
211+CTB1MAST EQU    X'40' .  1 MASTER TERMINAL
212+CTB1SUBP EQU    X'20' .  2 SUBPOOL CTB
213+CTB1DIAL EQU    X'10' .  3 DIAL CTB IS PHYSICALLY CONNECTED
214+CTB1SIGN EQU    X'08' .  4 DIAL CTB IS LOGICALLY CONNECTED
215+CTB1HELD EQU    X'04' .  5 CONVERSATION IN PROGRESS WAS HELD
216+CTB1PCNT EQU    X'02' .  6 PRESET CNT
217+CTB1PSMB EQU    X'01' .  7 PRESET SMB

219+CTBFLAG2 DS     C .
220+CTB2NOIN EQU    X'80' .  0 NO INPUT
221+CTB2NOOU EQU    X'40' .  1 NO OUTPUT
222+CTB2NOQU EQU    X'20' .  2 NO QUEUEING
223+CTB2LOCK EQU    X'10' .  3 LOCKED
224+CTB2TEST EQU    X'08' .  4 TEST MODE
225+CTB2EXCL EQU    X'04' .  5 EXCLUSIVE MODE
226+CTB2INOP EQU    X'02' .  6 INOPERABLE
227+CTB2EDIT EQU    X'01' .  7 USER OUTPUT EDIT REQUESTED

229+CTBFLAG3 DS     C .
230+CTB3READ EQU    X'80' .  0 INPUT ONLY DEVICE
231+CTB3QSYS EQU    X'40' .  1 QUEUE ALL SYSTEM MESSAGES
232+CTB3LOOP EQU    X'20' .  2 LOOP TEST PENDING
233+CTB3CHE  EQU    X'10' .  3 LINE DISCONNECT REQUESTED
234+CTB3QERR EQU    X'08' .  4 ERROR ON GET NEXT FOR OUTPUT
235+CTB3QMSG EQU    X'04' .  5 QUEUE CAUSE OF ERROR WITH SYSMSG
236+CTB3LAST EQU    X'02' .  6 LAST CTB ON THIS LINE
237+CTB3SEG1 EQU    X'01' .  7 FIRST SEGMENT

239+CTBFLAG4 DS     C .
240+CTB4CNCL EQU    X'80' .  0  DEQUEUE MESSAGE IN PROCESS REQUESTED
241+CTB4OUTP EQU    X'40' .   OUTPUT ONLY TERMINAL
242+*        EQU    X'20' .  2  NOT USED
243+*        EQU    X'10' .  3  NOT USED
244+*        EQU    X'08' .  4  NOT USED
245+*        EQU    X'04' .  5  NOT USED
246+*        EQU    X'02' .  6  NOT USED
247+*        EQU    X'01' .  7  NOT USED

249+CTBACTL  DS     C .        FLAGS TO BE USED BY ANALYZER
250+CTBAINP  EQU    X'80' .  0  0=OUTPUT NEXT, 1= INPUT NEXT
251+CTBAEOM  EQU    X'40' .  1  1=END OF OUTPUT MSG
252+CTBASHUT EQU    X'20' .  2  1=SYSTEM SHUTDOWN
253+CTBAINC  EQU    X'10' .  3  1=INCORE SYSTEM MESSAGE
```

```
 254+CTBAMULT EQU    X'08' .  4   REJECT IF NOT MULTIPLE SEGMENT MESSAGE
 255+CTBAERR  EQU    X'04' .  5   ERROR FOUND ON LAST SEGMENT PASSED

 257+CTBDCTL  DS     C .          FLAGS USED BY DD ROUTINES
 258+CTBDCTL2 DS     C .          FLAGS USED BY DD ROUTINES

 260+CTBFEAT  DS     C .          FEATURES FLAGS
 261+CTBFSYN1 EQU    X'08' ASYNCHRONOUS DEVICE,  SYNC FLAG 1
 262+CTBFSYN2 EQU    X'04' ASYNCHRONOUS DEVICE,  SYNC FLAG 2
 263+CTBFPAGE EQU    X'02' PAGING IN PROCESS
 264+CTBFNAPD EQU    X'01' AUTOMATIC PAGE DELETION NOT REQUESTED

 266+CTBCOMP  DS     C .          COMPONENTS TWO BITS FOR EACH
 267+CTBC1NA  EQU    X'80' COMPONENT 1 IS NOT ATTACHED            3088
 268+CTBC1IP  EQU    X'40' COMPONENT 1 IS INOPERABLE              3088
 269+CTBC2NA  EQU    X'20' COMPONENT 2 IS NOT ATTACHED            3088
 270+CTBC2IP  EQU    X'10' COMPONENT 2 IS INOPERABLE              3088
 271+CTBC3NA  EQU    X'08' COMPONENT 3 IS NOT ATTACHED            3088
 272+CTBC3IP  EQU    X'04' COMPONENT 3 IS INOPERABLE              3088
 273+CTBC4NA  EQU    X'02' COMPONENT 4 IS NOT ATTACHED            3088
 274+CTBC4IP  EQU    X'01' COMPONENT 4 IS INOPERABLE              3088

 276+CTBINCT  DS     H .          INPUT MSG COUNT
 277+CTBOUTCT DS     H .          OUTPUT MSG COUNT
 278+CTBCPMP  DS     H .          PASSWORD MATRIX POINTER
 279+CTBCNTP  DS     H .          LOGICAL TERMINAL POINTER
 280+CTBPREST DS     H .          PRESET DESTINATION OFFSET
 281+CTBCOB   DS     H .          OFFSET TO COMMUNICATION OVERFLOW BLOCK
 282+CTBCCB   DS     H .          OFFSET TO CONVERSATIONAL CONTROL BLOCK
 283+CTBCXB   DS     H .          COMMUNICATION EXTENT BLOCK POINTER
 284+CTBPGNO  DS     H .          CURRENT PAGE BEING VIEWED
```

```
367            ISCD  SCDBASE=0
368+SCD        DSECT
369+*** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ***
370+** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **
371+**                                                                   **
372+**              IMS SYSTEM CONTENTS DIRECTORY  (SCD)                 **
373+**                                                                   **
374+** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **
375+*** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ***


377+SSCDORG   DS    0D SECONDARY SCD ORIGIN
378+SSCDSVC1  SVC   0 USER TYPE 1 SVC NUMBER
379+         BR    14 RETURN ON REGISTER 14
380+SSCDSVC2  SVC   0 USER TYPE 2 SVC NUMBER
381+         BR    14 RETURN ON REGISTER 14
382+SSCDSVC3  SR    15,15 CLEAR REGISTER
383+         BR    14 RETURN ON REGISTER 14
384+SSCDSVC4  SVC   255 USER TYPE 4 SVC NUMBER
385+         BR    14 RETURN ON REGISTER 14
386+SSCDOCEP  DC    CL2'Z9' CSAM CHANNEL END APPENDAGE SUFFIX
387+SSCDOSPS  DC    X'10' OS/360 SYSTEM CONFIGURATION
388+SSCDOSLV  DC    AL1(18) CS/360 RELEASE LEVEL SPECIFIED
389+SSCDIMSR  DC    AL1(0) IMS RELEASE NUMBER.
390+SSCDIMSL  DC    AL1(0) IMS MOD LEVEL NUMBER.
391+SSCDASPS  DC    X'00' SYSTEM CONFIGURATION - ACTUAL
392+SSCDASLV  DC    AL1(0) OS/360 RELEASE LEVEL - ACTUAL
393+SSCDDESC  DC    AL1(0) MCS DESCRIPTOR CODE
394+         DC    AL3(0) RESERVED FOR OS/360 EXPANSION
395+SSCDROUT  DC    AL2(0) MCS ROUTING CODES
396+         DC    AL2(0) RESERVED FOR OS/360 EXPANSION
397+DFSXWAIT  DC    V(DFSIWAIT) IWAIT EP ADDRESS
398+DFSXIOBA  DC    V(DFSIOS40) OSAM IOB QCB
399+DFSXOPEN  DC    A(DFSXOPEN) OSAM OPEN EP
400+SSCDMCID  DC    AL1(0) CCNSOLE ID - START IMS COMMAND.
401+         DC    AL3(0) RESERVED
402+         DC    F'0' RESERVED
```

```
405+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
406+*                                                                         *
407+*       R E C O R D E R   L O G   S E C T I O N :                         *
408+*                                                                         *
409+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

411+SCDREENT DC    V(DFSFLOI0) RECORDER ENTRY POINT
412+SCDREDCB DC    A(0) DCB AND WKAREA LIST ADDRESS
413+SCDREWRK DC    V(DFSILOGW) WORK AREA FOR THE LOGGER
414+SCDREPLN DC    H'28' LENGTH OF LOG PREFIX IN BYTES

416+*** LOG CONTROL BYTE ***
417+SCDRECTL DC    X'0000'
418+SCDLGOPN EQU   X'80' LOG IS OPEN
419+SCDLGTRM EQU   X'40' LOG SMBTASK DETACHED

421+SCDRELCT DC    H'500' CHECKPOINT LOG FREQUENCY CONTROL REFERENCE
422+SCDRECCT DC    H'500' CURRENT LOG FREQUENCY COUNTER, CPT AT ZERO
423+SCDRPENT DC    A(0) ENTRY TO LOG WRITER
424+SCDRTECB DC    F'0' LOG TASK TERMINATE ECB
425+SCDRTCB  DC    A(0) LOG TASK CONTROL BLOCK ADDRESS
426+SCDRETXR DC    A(0) RESERVED
427+SCDRPFX  DC    V(DFSIPREF) ENTRY TO LOG PREFIX BUILDER
428+SCDDBLNT DC    A(SCDDBLNT) ENTRY TO DATABASE CHANGE LOGGING ROUTINE
429+SCDCWRK  DC    F'0' CHECKPOINT/DB LOG WORK AREA ADDRESS
430+SCDCWRKL DC    H'512' LENGTH OF CHECKPOINT/DB LOG WORK AREA
431+SCDCPNO  DC    H'0' CHECKPOINT NUMBER
432+SCDDATE  DC    F'0' DATE FOR RESTART/DATABASE LOG
433+SCDTIME  DC    F'0' TIME FOR RESTART/DATABASE LOG
434+SCDSEQ   DC    H'0' DATABASE LOG SEQUENCE NUMBER
435+SCDRGTYP DC    AL1(0,0) REGION TYPE (0=CTL,3=DLI)
436+SCDDBDCB DC    A(0) ADDRESS OF DATABASE LOG DCB
437+         DC    4F'0' RESERVED FOR SYSTEM EXPANSION
```

```
 439+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 440+*                                                                             *
 441+*  D A T A    L A N G U A G E / I    &    O S A M    S E C T I O N :          *
 442+*                                                                             *
 443+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

 445+SCDDLICT DC    A(SCDDLICT) DL/I ANALYZER ENTRY POINT (DFSDLA00)
 446+SCDDLIMQ DC    A(0) RESERVED
 447+SCDISAM  DC    A(SCDISAM) ISAM SIMULATOR ENTRY POINT
 448+SCDDLARE EQU   22 RETURN POINT FROM ANALYZER ENTRY POINT
 449+SCDDLIRE DC    A(SCDDLIRE) RETRIEVE MODULE ENTRY POINT
 450+SCDDLIIN DC    A(SCDDLIIN) INSERT MODULE ENTRY POINT
 451+SCDDLIDR DC    A(SCDDLIDR) DELETE/REPLACE MODULE ENTRY POINT
 452+SCDDLDAE EQU   18 ALTERNATE ENTRY TO DELETE/REPLACE
 453+SCDDLIHS DC    A(SCDDLIHS) HSAM ENTRY POINT
 454+SCDDLIPS DC    V(DFSIDIR0) BEGINNING OF PSB DIRECTORY ENTRIES
 455+SCDPSBFR DC    A(0) EP TP PSB FREE POOL SPACE ROUTINE
 456+SCDDLILN DC    H'40' LENGTH OF PSB DIRECTORY ENTRY
 457+SCDDLINC DC    H'0' NUMBER OF PSB DIRECTORY ENTRIES
 458+SCDPSBSW DS    0XL1 0 - FREE PSB POOL SPACE
 459+SCDDLIDM DC    A(0) BEGINNING OF DMB DIRECTORY ENTRIES
 460+SCDDLIDL DC    H'36' LENGTH OF EACH DMB DIRECTORY ENTRY
 461+SCDDLIDN DC    H'0' NUMBER OF DMB DIRECTORY ENTRIES
 462+SCDDLIPA DC    V(DFSIPST) START OF PST BLOCKS
 463+SCDDLIPL DC    H'424' LENGTH OF EACH PST
 464+SCDDLIPN DC    H'0' NUMBER OF PST ENTRIES
 465+SCDDLIBD DC    A(0)
 466+SCDDMBSW DS    0XL1 0 - RELEASE DMB SPACE, 1 - RESERVED
 467+SCDDMBFR DC    A(0) ENTRY TO DMB FREE POOL SPACE ROUTINE
 468+SCDPSBMU DC    V(DFSIPSBQ) PSB MOST USED QCB ADDRESS
 469+SCDDPDM  DC    V(DFSDPDM0) PSB/DMB POOL MANAGER
 470+SCDACBDC DC    V(ACBDCB) ACBLIB DCB POINTER
 471+SCDPSBPL DC    F'0'
 472+SCDDMBPL DC    F'0'
 473+SCDLLOGD DC    7F'0'
 474+SCDDLIMV DC    V(DFSDBLM0) BLOCK MOVER ENTRY POINT
 475+SCDDSST  DC    V(DFSDSST0) SEGMENT INTENT RESERVATION CSECT
 476+SCDDLICL DC    A(SCDDLICL) DATA MANAGEMENT OPEN/CLOSE ENTRY POINT
 477+SCDDLI07 DC    A(SCDDLI07) ENTRY TO DL/I LOAD PROGRAM
 478+SCDWQCB  DC    V(DFSPWQCB) DL/I MOVER QCB ADDRESS
 479+SCDBPARM DC    A(0) DL/I REGION PARMS (PXPARMS)
 480+SCDASE   DC    V(DFSIASE0) APPLICATION SCHEDULER TERMINATOR EP
 481+SCDDBFPL DS    A LOCATION OF DL/I BUFFER POOL
 482+SCDDDBH0 DC    A(0) ENTRY POINT OF DL/I BUFFER HANDLER
 483+SCDDHDS0 DC    A(0) ENTRY POINT OF GET/FREE SPACE
 484+SCDDXMT0 DC    A(0) ENTRY POINT OF INDEX MAINTENANCE
 485+SCDDBCVT DC    A(0) ENTRY POINT OF BYTE BLOCK CONVERSION
 486+SCDOIOBS EQU   DFSXIOBA OSAM IOB QCB
 487+SCDOOPEN EQU   DFSXOPEN OSAM OPEN EP
 488+SCDIWAIT EQU   DFSXWAIT IWAIT ENTRY POINT ADDRESS
 489+SCDSAVE  DC    V(DFSISAV0) ORIGIN OF SAVE AREA SET
 490+SCDFLOS  DC    A(0) STAE EXIT ROUTINE ENTRY POINT
 491+SCDFLOSR DC    A(0) STAE RETRY ROUTINE ENTRY POINT
 492+SCDNAVID DS    0F
 493+SCDRLDTE DC    X'00'
```



```
 494+SCDSYNCN DC    AL3(0)
 495+SCDLOWID DC    F'0'
 496+         DC    6F'0' RESERVED FOR SYSTEM EXPANSION
```

```
498+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
499+*                                                                     *
500+*     S T O R A G E   M A N A G E M E N T   S E C T I O N :           *
501+*                                                                     *
502+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

504+SCDSMMLC DC     A(0) ADDRESS OF START OF SMM CORE AREA
505+SCDSMML  DC     F'0' LENGTH OF STORAGE MANAGEMENT CORE AREA
506+SCDSMMHI DC     A(0) ADDRESS OF END OF SMM CORE AREA
507+SCDSMMGC DC     A(0)
508+SCDSMMFC DC     A(0)
509+SCDSMMCP DC     V(DFSICRET) ICREAT ENTRY POINT
510+SCDSMMDP DC     V(DFSIDEST) IDESTROY ENTRY POINT
511+SCDSMMFB DC     V(DFSIFBUF) IFREEBUF ENTRY POINT
512+SCDSMMGB DC     V(DFSIGBUF) IGETBUF ENTRY POINT
513+SCDILMSK DC     X'0001' PROGRAM LOAD MASK
514+         DS     XL2 NOT USED
515+         DS     3F PADDING



517+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
518+*                                                                     *
519+*         E N Q U E U E / D E Q U E U E   S E C T I O N :             *
520+*                                                                     *
521+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

523+SCDIDEQ  DC     V(DFSIIDEQ) IDEQ ENTRY POINT
524+SCDIENQ  DC     V(DFSIIENQ) IENQ ENTRY POINT
525+         DS     2F
```



```
527+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
528+*                                                                     *
529+*         Q U E U E   M A N A G E M E N T   S E C T I O N :
530+*                                                                     *
531+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

533+SCDIRWQE DC     V(DFSQMGR0) IREAD/IWRITE QUEUE ENTRY POINT
534+SCDIQDCB DC     V(DFSIQDCB) ADDRESS OF INPUT QCR DCB
535+SCDSMDCB DC     V(DFSSMDCB) ADDRESS OF IMS2.SHMSG DCB
536+SCDLMDCB DC     V(DFSLMDCB) ADDRESS OF IMS2.LGMSG DCB
537+SCDSPDCB DC     V(DFSICDCB) ADDRESS OF SCRATCH PAD AREA (SPA) DCB
538+SCDQPOOL DC     A(0) POINTER TO QUEUE BLOCK POOL
539+SCDPRLN1 DC     H'40' LENGTH OF FULL MESSAGE PREFIX
540+SCDPRLN2 DC     H'16' LENGTH OF CONDENSED MESSAGE PREFIX
541+SCDQCRSZ DC     H'0' SIZE OF QCR BUFFER
542+SCDMSGSZ DC     H'0' SIZE OF MSG BUFFER
543+SCDQMREB DC     F'0' SAVE AREA PTR TO BUFFER ALLOC FOR REUSE
544+SCDCKCL  DC     F'0' COMM FOR INTERNAL CHECKPOINTS
545+SCDQMSRB DC     H'0' RECORDS RESERVED FOR SHUTDOWN/REUSE
546+         DC     5H'0' RESERVED FOR SYSTEM EXPANSION
```

```
548+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
549+*                                                                        *
550+*              C O M M U N I C A T I O N    S E C T I O N :              *
551+*                                                                        *
552+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

554+SCDCTT    DC    V(DFSICTT) COMMUNICATION TRANSLATION TABLE BASE ADDR
555+SCDCTTL   DC    H'28' LENGTH OF EACH CTT
556+SCDCTTN   DC    H'0' NUMBER OF CTTS
557+SCDCLB    DC    V(DFSICLB) COMMUNICATION LINE BLOCK BASE ADDRESS
558+SCDCLBL   DC    H'100' LENGTH OF EACH CLB
559+SCDCLBN   DC    H'0' NUMBER OF CLBS
560+SCDCTB    DC    V(DFSICTB) COMMUNICATION TERMINAL BLOCK BASE ADDRESS
561+SCDCTBL   DC    H'32' LENGTH OF EACH CTB
562+SCDCTBN   DC    H'0' NUMBER OF CTBS
563+SCDCXB    DC    A(0) COMMUNICATION EXTENT BLOCK BASE ADDRESS
564+SCDCXBL   DC    H'20' LENGTH OF EACH CXB
565+SCDCXBN   DC    H'0' NUMBER OF CXBS
566+SCDCNT    DC    V(DFSICNT) COMMUNICATION NAME TABLE BASE ADDRESS
567+SCDCNTL   DC    H'44' LENGTH OF EACH CNT
568+SCDCNTN   DC    H'0' NUMBER OF CNTS
569+SCDCDB    DC    V(DFSICDB) COMMUNICATION DIRECTORY BLOCK BASE ADDR
570+SCDCDBL   DC    H'4' LENGTH OF A CDB
571+SCDCDBN   DC    H'0' NUMBER OF CDBS
572+SCDCVB    DC    V(DFSICVB) COMMUNICATION VERB BLOCK BASE ADDRESS
573+SCDCVBL   DC    H'12' LENGTH OF EACH CVB
574+SCDCVBN   DC    H'0' NUMBER OF CVBS
575+SCDCPT    DC    A(0) COMMUNICATION PRIORITY TABLE BASE ADDRESS
576+SCDCPTL   DC    H'0' LENGTH OF EACH CPT
577+SCDCPTN   DC    H'0' NUMBER OF CPTS
578+SCDCPM    DC    A(0) COMMUNICATION PRIORITY MATRIX BASE ADDR
579+SCDCPML   DC    H'0' LENGTH OF EACH CPM
580+SCDCPMN   DC    H'0' NUMBER OF CPMS
581+SCDCTM    DC    V(DFSICTM) COMMUNICATION TERMINAL MATRIX BASE ADDR
582+SCDCTML   DC    H'0' LENGTH OF EACH CTM
583+SCDCTMN   DC    H'0' NUMBER OF CTMS
584+SCDCCB    DC    A(0) CONVERSATIONAL CONTROL BLOCK BASE ADDRESS
585+SCDCCBL   DC    H'32' CCB LENGTH
586+SCDCCBN   DC    H'0' NUMBER OF CCBS
587+SCDCCBND  DC    AL2(0) OFFSET TO FIRST DISK CCB
588+SCDCCBNC  DC    AL2(0) OFFSET TO FIRST INCORE CCB
589+SCDCCBDL  DC    H'0' MAXIMUM LENGTH OF DISK SPA
590+SCDCCBCL  DC    H'0' MAXIMUM LENGTH OF INCORE SPA
591+SCDCAN01  DC    V(DFSCIOC1) COMMUNICATION ANALYZER ENTRY # 1
592+SCDCAN02  DC    V(DFSCIOC2) COMMUNICATION ANALYZER ENTRY # 2
593+SCDCAN03  DC    V(DFSCIOC3) COMMUNICATION ANALYZER ENTRY # 3
594+SCDCAN04  DC    V(DFSCIOC4) COMMUNICATION ANALYZER ENTRY # 4
595+SCDCAN05  DC    V(DFSCIOC5) COMMUNICATION ANALYZER ENTRY # 5
596+SCDCAN06  DC    V(DFSCIOC6) COMMUNICATION ANALYZER ENTRY # 6
597+SCDCAN07  DC    V(DFSCIOC7) COMMUNICATION ANALYZER ENTRY # 7
598+SCDCAN08  DC    V(DFSCIOC8) COMMUNICATION ANALYZER ENTRY # 8
599+SCDCAN09  DC    V(DFSCIO09) COMMUNICATION ANALYZER ENTRY # 9
600+SCDCAN10  DC    V(DFSCIO10) COMMUNICATION ANALYZER ENTRY # 10
601+SCDCAN11  DC    V(DFSCIO11) COMMUNICATION ANALYZER ENTRY # 11
602+SCDFNDST  DC    V(DFSICLFO) FIND DESTINATION ROUTINE ENTRY POINT
```

```
 603+SCDMSGRT DC     V(DFSICLRO) MESSAGE ROUTER ROUTINE ENTRY POINT
 604+SCDMTRM  DC     V(DFSCTBMT) MASTER TERMINAL CTB ADDRESS
 605+SCDCLISO DC     V(DFSICLSO) SECURITY ROUTINE ENTRY POINT
 606+SCDTRANS DC     V(DFSICLTO) TRANSLATE ROUTINE ENTRY POINT
 607+SCDCONV  DC     A(0) CONVERSATION PROCESSOR ENTRY POINT
 608+SCDCOB   DC     A(0) COMMUNICATION OVERFLOW BLOCK BASE ADDRESS
 609+SCDCOBL  DC     H'32' LENGTH OF EACH COB
 610+SCDCOBN  DC     H'0' NUMBER OF COBS
 611+SCDCTCLB DC     V(DFSICLB) CONSOLE TYPEWRITER CLB
 612+SCDREPOL DC     V(DFSICLXO) RESET POLL MODULE ENTRY POINT
 613+SCDNTB   DC     V(DFSINTBO)
 614+SCDEDITC DC     A(0) USER EDIT ROUTINE FOR CNT DESTINATIONS
 615+SCDEDITS DC     V(DFSLISTO) USER EDIT ROUTINES FOR SMB DESTINATIONS
 616+SCDINCT  DC     H'0'
 617+SCDOUTCT DC     H'0'
 618+SCDCPCTL DC     X'00' CHECKPOINT POST SYNC BITS
 619+SCDPCP01 EQU    X'80'
 620+SCDPCP02 EQU    X'40'
 621+SCDPCP03 EQU    X'20'
 622+         DC     AL3(0) RESERVED FOR SYSTEM EXPANSION
 623+         DC     9F'0' RESERVED FOR SYSTEM EXPANSION
```



```
 625+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 626+*                                                                       *
 627+*              D I S P A T C H E R    S E C T I O N :                   *
 628+*                                                                       *
 629+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

 631+SCDEXTWQ DC     V(DFSEXTWQ) EXTERNAL WAIT QUEUE ADDRESS
 632+SCDINTWQ DC     V(DFSINTWQ) INTERNAL WAIT QUEUE ADDRESS
 633+SCDINTSQ DC     V(DFSINTSQ) INTERNAL SAVE QUEUE ENTRY POINT ADDRESS
 634+SCDTECBR DC     V(DFSITERM) TERMINATION ROUTINE ENTERED IF TECB POST
 635+SCDPARM  DC     A(0) ADDRESS OF PARM LIST FROM SYSTEM
 636+SCDTECB  DC     F'0' IMS TERMINATION ECB POSTED BY CHECKPOINT
 637+SCDSAVEL DC     H'508' LENGTH OF SAVE AREA SETS INCLUDING QE
 638+SCDSLEV  EQU    7 NUMBER OF SAVE AREAS IN A SET
 639+SCDSAVEN DC     H'0' NUMBER OF SAVE AREA SETS
 640+SCDSRQE  DC     V(DFSIRQEO) ORIGIN OF RQE LIST
 641+SCDSRQEL DC     H'8' LENGTH OF EACH RQE
 642+SCDSRQEN DC     H'0' NUMBER OF RQES
 643+SCDWAITQ DC     V(DFSIWATQ) ORIGIN OF WAIT QUEUE
 644+SCDEXTWL DC     H'4' LENGTH OF EACH ENTRY IN EXT WAIT QUEUE
 645+SCDEXTWN DC     H'0' NUMBER OF EXTERNAL EVENTS
 646+SCDSUBSV DC     A(0) ADDRESS OF LAST SUBTASK 1ST LEV SAVE AREA
 647+SCDCFAC  DC     X'00' CURRENT FIRST LEVEL FACILITY
 648+SCDCMOD  DC     X'00' CURRENT MODULE WITHIN FACILITY
 649+SCDTEST  DC     X'80' TEST MODE CONTROL INDICATOR
 650+SCDTLEV  DC     X'00' TEST MODE LEVEL INDICATOR
 651+SCDRES2  DC     F'0' PADDING
 652+         DC     4F'0' RESERVED FOR SYSTEM EXPANSION
```

```
654+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
655+*                                                                         *
656+*   A P P L I C A T I O N    S C H E D U L E R    S E C T I O N :         *
657+*                                                                         *
658+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

660+SCDASINT DC    V(DFSASINT) APPLICATION SCHEDULER INITIATOR ENTRY
661+SCDASTER DC    V(DFSASTRM) APPLICATION SCHEDULER TERMINATOR ENTRY
662+SCDSMBEP DC    V(DFSISMB) SYSTEM MESSAGE BLOCKS BASE ADDRESS
663+SCDSMBL  DC    H'56' LENGTH OF EACH SMB
664+SCDSMBN  DC    H'0' NUMBER OF SMBS
665+SCDSMBDQ DC    V(DFSSMBDQ) SMB DEQUEUE ENTRY POINT ADDRESS
666+SCDSMBNQ DC    V(DFSSMBEQ) SMB ENQUEUE ENTRY POINT ADDRESS
667+SCDCHKXB DC    A(SCDCHKEB) ADDRESS OF ECB TO BE POSTED BY DFSIAS00
668+SCDTCTEP DC    V(DFSITCT) TRANSACTION CLASS TABLES BASE ADDRESS
669+SCDTCTL  DC    H'80' LENGTH OF EACH TCT
670+SCDTCTN  DC    H'0' NUMBER OF TCTS
671+SCDMRQEP DC    V(DFSIMRQ) ADDRESS OF MESSAGE REQUEST QUEUE
672+SCDPRQEP DC    V(DFSIPRQ) ADDRESS OF PARTITION REQUEST QUEUE
673+SCDBPRQE DC    V(DFSIBPRQ) ADDRESS OF BATCH PARTITION REQUEST QUEUE
674+SCDEXTSQ DC    V(DFSEXTSQ) EXTERNAL SAVE QUEUE ENTRY POINT ADDRESS
675+         DC    4F'0'
```

```
677+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
678+*                                                                       *
679+*   C H E C K P O I N T / R E S T A R T     S E C T I O N :             *
680+*                                                                       *
681+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

683+*** CHECKPOINT STATUS BYTE ***
684+SCDCKCTL DC      X'00'
685+*                         DESCRIPTION OF BIT SETTINGS
686+*                              0  NOT USED
687+*                              1  DATABASE RECOVERY
688+*                              2  PURGE REQUEST
689+*                              3  SYSTEM SHUTDOWN REQUEST
690+*                              4  DUMP QUEUE REQUEST
691+*                              5  MASTER TERMINAL CHECKPOINT REQUEST
692+*                              6 DATABASE DUMP REQUEST
693+*                              7  SYSTEM SCHEDULED CHECKPOINT (LOG)

695+*** RESTART STATUS BYTE ***
696+SCDRSCTL DC      X'00'
697+*                         DESCRIPTION OF BIT SETTINGS
698+*                              0  PASSWORD SECURITY REQUESTED
699+*                              1  QUEUE ONLY RESTART
700+*                              2  PSB/DMB CHANGED - REBUILD BLOCKS
701+*                              3  ALLOW COMMANDS EXCEPT RESTART
702+*                              4  TERMINAL SECURITY REQUESTED
703+*                              5  EMERGENCY RESTART
704+*                              6 BUILD QUEUES
705+*                              7  NORMAL RESTART

707+*** COMMON SYSTEM SHUTDOWN STATUS BYTE ***
708+SCDSTOP1 DC      X'0000'
709+*                BYTE 1    DESCRIPTION OF BIT SETTINGS - USER
710+*                           0  STOP INPUT                 COMMUNICATIONS
711+*                           1  STOP OUTPUT                COMMUNICATIONS
712+*                           2  SEND ALL OUTPUT            COMMUNICATIONS
713+*                           3  PROCESS ALL MESSAGES       SCHEDULER
714+*                           4  QUEUE'S AVAILABLE          READ/WRITE Q
715+*                           5  FREE MESSAGE REGIONS       SCHEDULER
716+*                           6  TERMINATE ALL REGIONS      SCHEDULER
717+*                           7  STOP SMB INPUT             COMMUNICATIONS
718+*
719+*                BYTE 2    DESCRIPTION OF BIT SETTINGS - USER
720+*                           0  PROCESS AT SYSTEM PRIORITY SCHEDULER
721+*                           1  DATABASE STOP               CHECKPOINT
722+*                           2  FORCE EOV ON LOG FILE       CHECKPOINT
723+*                           3  ABEND AT TERMINATION        COMMUNICATIONS
724+*                           4  DISCONNECT ALL LINES        COMMUNICATIONS
725+*                           5  ALL OUTPUT SENT             COMMUNICATIONS
726+*                           6  SYSTEM CLOSE DOWN           CHECKPOINT
727+*                           7  FINAL ENTRY CONTROL         CHECKPOINT

729+SCDCKDBN DS      OCL1 # OF ENTERED DATABASE NAMES FOR RECOVERY
730+SCDCKDB  DC      A(0) POINTER TO POOL CONTAINING DATABASE NAMES
731+SCDLOGN  DS      OCL1 COUNT OF RESTART SERIAL NUMBERS
```

```
732+SCDLOG    DC    A(0) POINTER TO POOL CONTAINING SERIAL NUMBERS
733+SCDCLIPN  DC    H'0000' INPUT TERMINAL ACTIVE COUNT
734+SCDCLOPN  DC    H'0000' OUTPUT TERMINAL ACTIVE COUNT
735+SCDCPC01  DC    A(0) SCRATCH PAD AREA LOG ROUTINE ENTRY POINT
736+SCDCHKEP  DS    0F
737+SCDRSTEP  DC    V(DFSRSTC0) RESTART ROUTINE ENTRY POINT ADDRESS
738+SCDCHKEB  DS    0F
739+SCDRSTEB  DC    V(DFSRTECB) RESTART ECB ADDRESS
740+SCDDBTBL  DC    A(0) RESTART DATABASE NAME TABLE
741+          DC    4F'0'
```



```
743+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
744+*                                                                         *
745+*        S T A R T / S T O P   R E G I O N   S E C T I O N :             *
746+*                                                                         *
747+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

749+SCDREGOP  DC    A(0) ECB TO OPEN REGION
750+SCDREGCL  DC    A(0) ECB TO CLOSE REGION
751+SCDCMDCT  DC    X'00' # OF START/STOP REGION COMMANDS ISSUED
752+SCDREGST  DC    X'00' # OF PENDING STOP COMMANDS
753+SCDREGCT  DC    H'0000' COUNT OF OPEN REGIONS
754+SCDSRDR   DC    A(SCDPROC1) ADDRESS OF START READER COMMAND
755+SCDSRMBR  DC    A(SCDPROC2) ADDRESS OF MEMBER FIELD IN START READER
756+SCDSMMBR  DC    A(SCDPROC3) ADDRESS OF MSG REG PROC MEMBER NAME
757+SCDSINIT  DC    A(SCDPROC4) ADDRESS OF START INITIATOR COMMAND
758+SCDPINIT  DC    A(SCDPROC5) ADDRESS OF STOP INITIATOR COMMAND
759+          DC    2F'0'
760+SCDPSYS   EQU   SSCDOSPS PROGRAMMING SYSTEM OPTION BYTE FROM CVT
761+SCDPSMVT  EQU   X'10' VALUE IF MVT
762+SCDPSMFT  EQU   X'20' VALUE IF MFT-II
763+SCDPSPCP  EQU   X'40' VALUE IF PCP
764+SCDIMSR   EQU   SSCDIMSR IMS RELEASE NUMBER
765+SCDIMSL   EQU   SSCDIMSL IMS MOD LEVEL
766+          DC    2F'0' RESERVED FOR SYSTEM EXPANSION
```

```
768+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
769+*                                                                       *
770+*        C O M M A N D   S E C T I O N :                                *
771+*                                                                       *
772+*                  *** NOTE ***                                         *
773+*                           THE COMMAND SECTION MUST RESIDE LAST        *
774+*                           IN THE SYSTEM CONTENTS DIRECTORY SINCE      *
775+*                           THE COMMANDS ARE VARIABLE IN LENGTH.        *
776+*                                                                       *
777+*                           ANY ADDITION TO THE SCD                     *
778+*                           MUST PRECEDE THIS SECTION.                  *
779+*                                                                       *
780+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

782+SCDPROC1 DS    0F
783+         DC    AL2(IHB0C05-*) MESSAGE LENGTH
784+         DC    B'0000000000000000' MCSFLAGS FIELD
785+         DC    C'S IMSRDR,MBR=' MESSAGE
786+IHB0005  EQU   *
787+SCDPROC2 DS    CL9 MEMBER NAME FIELD OF START READER COMMAND
788+SCDPROC3 DC    AL1(6) LENGTH IN BYTES OF PROC MEMBER NAME
789+         DC    C'IMSMSG' MSG PROC MEMBER NAME
790+SCDPROC4 DS    0F
791+         DC    AL2(IHB0C06-*) MESSAGE LENGTH
792+         DC    B'0000000000000000' MCSFLAGS FIELD
793+         DC    C'S INIT.CLASSB,,,(B,A) ' MESSAGE
794+IHB0006  EQU   *
795+SCDPROC5 DS    0F
796+         DC    AL2(IHB0C07-*) MESSAGE LENGTH
797+         DC    B'0000000000000000' MCSFLAGS FIELD
798+         DC    C'P CLASSB' MESSAGE
799+IHB0007  EQU   *
800+         DS    4F


802                *,*****************************************************
803                *,*                 IMS/360 SYSTEM
804                *,*      VERSION 2   RELEASE 2   MOD LEVEL 0       *
805                *,*****************************************************


807             IAPS  SMBBASE=C
808+**********************************************************************
809+*                                                                    *
810+*         APPLICATION PROGRAM SCHEDULER DUMMY SECTIONS               *
811+*                                                                    *
812+**********************************************************************
813+*
814+*
815+*
816+**********************************************************************
817+*                                                                    *
818+*         SMB                                                        *
819+*                                                                    *
```

```
820+*****************************************************************************
821+SMB      DSECT
822+SMBQEBP  DS    A BACKWARD POINTER
823+SMBQEFP  DS    A FORWARD POINTER
824+SMBQCBDQ DS    A DRRN OF NEXT MESSAGE TO READ
825+SMBQCBEQ DS    A DRRN OF LAST MESSAGE WRITTEN

827+SMBQFLG1 DS    C SMB QUEUE FLAG 1
828+*        EQU   X'80'             QUEUE READ IN PROCESS
829+*        EQU   X'40'             QUEUE 1 HAS MESSAGE ENQUEUED
830+*        EQU   X'20'             QUEUE 2 HAS MESSAGE ENQUEUED
831+*        EQU   X'10'             QUEUE 3 HAS MESSAGE ENQUEUED
832+*        EQU   X'08'             QUEUE 4 HAS MESSAGE ENQUEUED
833+*        EQU   X'04'             MESSAGE EXISTS IN BACKUP QUEUE
834+*        EQU   X'02'             QCBDQ POINTS TO QUEUE BLOCK RECORD

836+SMBQFLG2 DS    C SMB QUEUE FLAG 2
837+*        EQU   X'80'             THIS DESTINATION IS PERMANENT
838+*        EQU   X'40'             AVERAGE LENGTH IS AVAILABLE
839+*        EQU   X'20'             ENQ/DEQ COUNT IS AVAILABLE
840+*        EQU   X'10'             NAME FIELD EXISTS
841+*        EQU   X'0F'             DESTINATION TYPE 0 THRU 15

843+SMBQAVGL DS    H AVERAGE MESSAGE LENGTH FOR DESTINATION
844+SMBDQCT  DS    H NUMBER OF MESSAGES DEQUEUED
845+SMBNQCT  DS    H NUMBER OF MESSAGES ENQUEUED
846+*                               NOTE: SMBNQCT-SMBDQCT=OLD SMBNQCT FIELD
847+SMBTRNCD DC    D'0' SMB TRANSACTION CODE
848+SMBSTATS DC    X'00' SMB STATUS BYTE

850+SMBFLAG1 DS    C SMB FLAG 1
851+SMB1CONV EQU   X'80' CONVERSATIONAL SMB
852+SMB1UPP  EQU   X'40' UPPER CASE TRANSLATION REQUESTED
853+SMB1CSPA EQU   X'20' INCORE SPA REQUEST
854+SMB1INIT EQU   X'08' INIT PGM BFRS WITH EACH NEW MESSAGE
855+SMBAGE   EQU   X'04' SMB IS AN AGING SMB
856+SMBENQ   EQU   X'02' SMB IS ENQUEUED ON TCT
857+SMBPST   EQU   X'01' SMB IS ENQUEUED ON PST

859+         DS    C RESERVED FOR FUTURE USE - 3270 FORMAT
860+         DS    C RESERVED FOR FUTURE USE - 3270 FORMAT

862+SMBQCBPR DS    C SMB CURRENT PRIORITY
863+SMBSLTE  DS    C USER EDIT ROUTINE NUMBER
864+SMBPRIOR DS    C LIMIT AND NORMAL PRIORITY
865+SMBCLASS DS    C SMB MESSAGE CLASS FOR SCHEDULING
866+SMBCTMP  DS    CL2 COMMUNICATION TERMINAL MATRIX PTR (REL)
867+SMBCPMP  DS    CL2 COMMUNICATION PASSWORD MATRIX PTR (REL)
868+SMBSPAL  DS    H LENGTH OF SPA
869+SMBLMTCT DS    H ENQUEUED LIMIT COUNT
870+SMBCOUNT DS    H PROCESSING LIMIT COUNT
871+SMBTIME  DS    H TIME SLICE
872+SMBTQE   DS    H RELATIVE TQE POINTER
873+SMBPSBDP DS    CL2 PSB DIRECTORY POINTER (RELATIVE)
```

USER EDIT ROUTINE INCLUSION DURING SYSTEM DEFINITION

All user-supplied edit routines should be placed in the OS partitioned data set defined by the USERLIB= operand of the IMSGEN macro-instruction of IMS/360 system definition. This must be performed prior to execution of IMS/360 system definition Stage 2. If the user does not specify a value for the USERLIB= operand, IMS/360 system definition assumes the IMS2.RESLIB data set contains any user-defined edit routines. System definition will attempt to obtain any user-specified edit routines from the specified library during Stage 2 of execution and link edit them as part of the IMS/360 control program nucleus. The names of the edit routines specified to IMS/360 system definition should be the same as the CSECT and load module names for the edit routine modules in the library specified by USERLIB=. The message switch edit routine must have a CSECT and load module name of DFSCNTE0.

1030 EDIT ROUTINE

Since a 1030 terminal can not always check for erroneous input data and normally can not specify input message destination, IMS/360 requires that an edit routine be defined at system definition time for input from 1030 terminals. Only one routine may be defined. This routine is invoked for each input message entered from a 1031 or 1035 to enable the user to verify the validity of his data and define message destination. A minimum validity check would be for the length of the message. The routine should also insert the IMS/360 transaction code, logical terminal name, or command verb at the beginning of the message to define destination. This user-supplied edit routine gains control before any IMS/360 security checking, validity checking, or editing functions are performed. The message text has previously been converted to EBCDIC.

The 1030 edit routine may look at the IMS/360 blocks, but may not change them. If the message is valid, the edit routine should insert a valid IMS/360 transaction code and return a return code of zero in Register 15. Register 2 must contain the final edited message length and the message must be placed in the buffer, beginning at the address specified in Register 1, at entry to the edit routine. If the length returned in Register 2 is zero, no IMS/360 processing is performed.

In the 1030 edit routine, the user may write his messages to a QSAM file rather than process them through IMS/360. If this is done, a length of zero must be returned to IMS/360 in Register 2. In addition, the user must define and provide access to the QSAM data set. The user should provide restart capabilities for such data sets himself, as IMS/360 is not aware of their existence. If a length of zero is returned to IMS/360, the message is not placed on the IMS/360 system log. Multiple output buffers should be used for QSAM files to prevent Operating System waits which degrade IMS/360 performance.

If the message is invalid, the edit routine should return a return code of 4 in Register 15. This prevents a positive acknowledge by IMS/360 to the terminal and causes the Repeat light to be set on.

Any return code except 0 or 4 causes the input terminal to be made inoperable. An error message is also generated and sent to the IMS/360 master terminal.

Interface to user edit routine:

CSECT name: DFS10300

Registers at entry:

0	Buffer length in binary. The resultant edited message cannot exceed this size.
1	Buffer address: The edited message text must start at this address.
2	Message length: This length includes the leading 10 bytes described later under buffer interface.
7	Communication Terminal Block (CTB address
8	Communication Translate Table (CTT) address
11	System Contents Directory (SCD) address
13	Save area address. The first three words in the save area may not be changed.
14	Return address
15	Entry point address to edit routine

Registers at exit:

2	Edited message length (the message must start in the address given in Register 1 at entry)
13	Save area address (must be the same as at entry)
15	Return code

Message buffer format and content at entry:

Byte	
1	IMS/360 communication line number
2	Relative terminal address (terminals are numbered from 1 on each line)
3-10	blanks
11-n	message; n equals value in Register 2

Note: Buffer content, beyond the definition above, is unpredictable, since the buffer area following the message is not initialized.

Buffer at exit:

Byte	
1-n	message; n equals value in Register 2

1030 EDIT ROUTINE INCLUSION DURING SYSTEM DEFINITION

The user-supplied 1030 edit routine should be placed in the operating system partitioned data set defined by the USERLIB= operand of the IMSGEN macro-instruction under the member name DFS10300. This should be performed prior to execution of IMS/360 system definition Stage 2. If the user does not specify a data set name for the USERLIB= parameter, the IMS2.RESLIB is assumed to contain the 1030 edit routine named or defined in system definition. System definition will attempt to obtain the user-defined edit routine from the specified library during system definition Stage 2 and link edit it into the IMS/360 online control program nucleus. The 1030 edit routine must exist as a load module under the name DFS10300 and must have the CSECT name of DFS10300.

SAMPLE 1030 EDIT ROUTINE - DFS10300

DFS10300 illustrates the functions that are to be performed by the user's own 1030 edit routine. DFS10300 was written to test the 1030 support in IMS/360, and is tailored for the test installation. The concept should be modified to the user's needs and environment.

IMS/360 1030 TEST INSTALLATION ENVIRONMENT

Communication Line #	Physical Terminal #	Type
12	1	1031 + 2 1035s
12	2	1031
13	1	1031
13	2	1033

All 1031s have a card reader, a badge reader, and a manual entry unit. The 1031 transaction code has the same meaning on all 1031s. All badge readers read 10 columns, and all manual entry units read 12 columns.

DFS10300 is capable of blocking "badge" input from one of the 1031s, and of creating a QSAM data set of "badge" input from another. Most 1030 input is prefixed with a transaction code and passed on to IMS/360 for processing. A special switch card is used to start and stop blocking and to open and close the QSAM data set.

1031 Card Column 1 Transaction Code	Reads Following Units
1	card reader, badge reader, manual entry unit
2	card reader, badge reader
3	card reader, manual entry unit
4	card reader
6	badge reader
7	manual entry unit

Other values of the transaction code are invalid.

CARD FORMATS

Two card formats are used in the system. One is transaction input, and the other is a switch card to control the QSAM data set and the blocking of "badge" input.

Transaction Card Format

Column	Contents
1	Transaction Code
2-4	Length of card input data
5-80	Data

Switch Card Format

Column	Contents
1	Must be "4"
2	Switch character
3	End of card character

Valid switch characters are:

- O Open the QSAM data set and send "badge" input from the 1031 on Line 13 to the QSAM data set.
- C Close the QSAM data set.
- B Start blocking "badge" input from Line 12 terminal 1.
- P Stop blocking "badge" input

All input is checked for both valid 1030 card column 1 transaction code and correct length. If an error is encountered, the edit routine returns to IMS/360 with a return code of 4. This causes IMS/360 to continue operations on the inputting line without sending a positive acknowledgment to the terminal. The Repeat light of the terminal is turned on.

DATA INPUT

Badge-Only Input

Line 12 terminal 2: A transaction code is added and the message is returned to IMS/360 for processing.

Line 12 terminal 1: If blocking is not requested, a transaction code is added and the message is returned to IMS/360 for processing. If blocking is requested, the message is moved to a blocking buffer within the edit routine. If the buffer thereby gets filled, the contents of the buffer are moved to the I/O buffer area, a transaction code is added, and the entire block is passed to IMS/360 for processing. If the buffer did not get filled, control is returned to IMS/360 with a zero return code and a zero length in Register 2. This indicates to IMS/360 that the edit routine has taken control of the message and IMS/360 continues processing on the inputting line with a positive acknowledgment.

Line 13 terminal 1: If the QSAM file is closed, a transaction code is added and the message is returned to IMS/360. If the QSAM file is open, the message is written to the QSAM file and control is returned to IMS/360 with a zero return code and a zero length. Positive acknowledgment is made to the terminal. No message is processed by IMS/360.

Manual Entry Input

A transaction code is added, and the inputting communication line and terminal number are appended. The message is thereafter returned to IMS/360 for processing.

Other Data Input

A transaction code is added, and the message is returned to IMS/360 for processing.

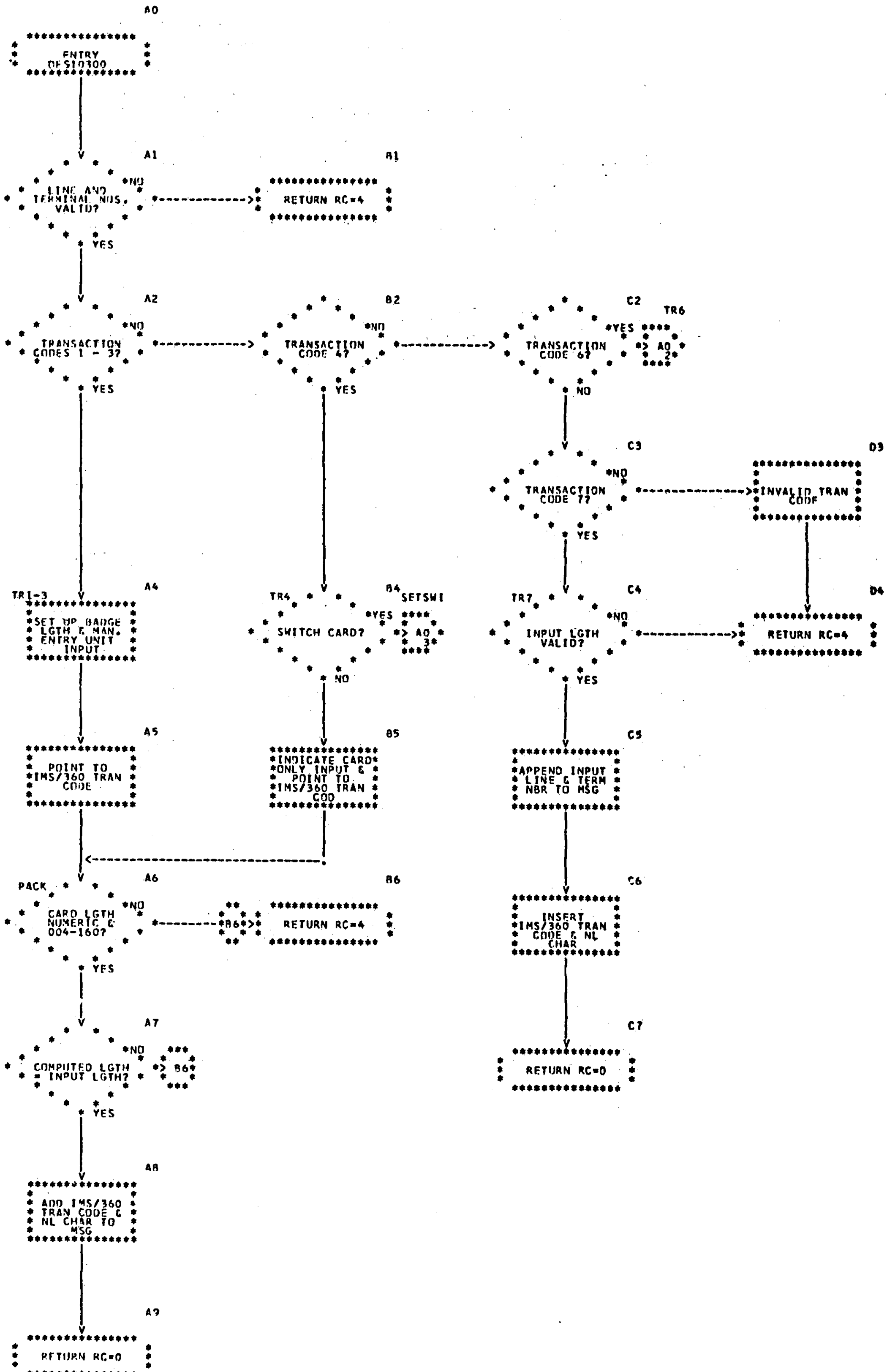
Switch Card Input

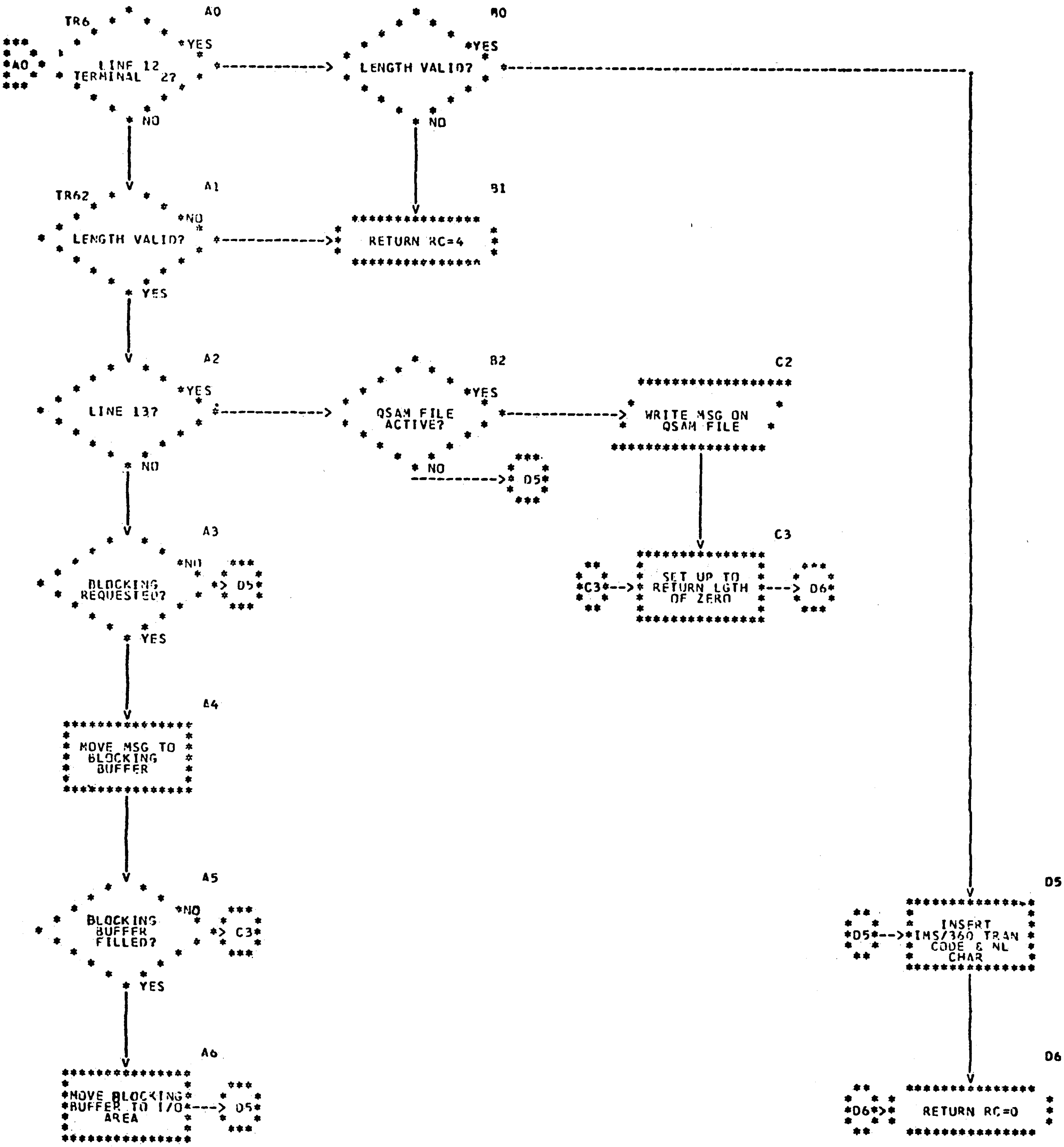
The requested action is taken and control is returned to IMS/360 with a zero return code and a zero length.

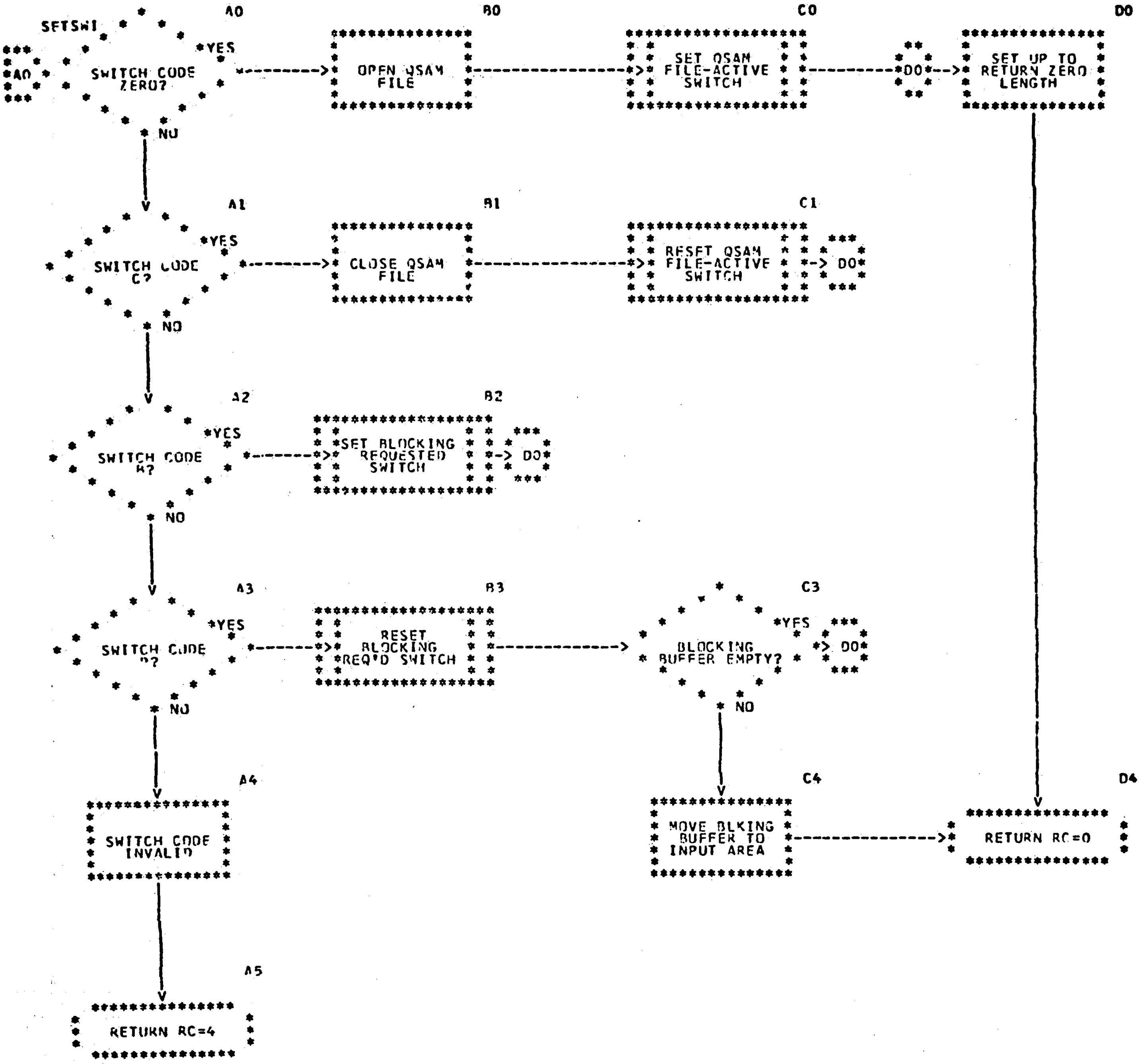
SAMPLE EDIT ROUTINE FLOWCHART

The flowchart for the sample problem follows.

The pages following the flowchart show the output received from the preceding sample problem.


A0
ENTRY DFSI0300
A1
LINE AND TERMINAL NOS. VALID?
NO
YES
B1
RETURN RC=4
A2
TRANSACTION CODES 1 - 3?
NO
YES
B2
TRANSACTION CODE 4?
NO
YES
C2
TR6
TRANSACTION CODE 6?
YES
A0 2
NO
C3
TRANSACTION CODE 7?
NO
YES
D3
INVALID TRAN CODE
TR1-3
A4
SET UP BADGE LGTH & MAN. ENTRY UNIT INPUT
TR4
B4
SETSWI
SWITCH CARD?
YES
A0 3
NO
TR7
C4
INPUT LGTH VALID?
NO
YES
D4
RETURN RC=4
A5
POINT TO IMS/360 TRAN CODE
B5
INDICATE CARD ONLY INPUT & POINT TO IMS/360 TRAN COD
C5
APPEND INPUT LINE & TERM NBR TO MSG
PACK
A6
CARD LGTH NUMERIC & 004-160?
NO
B6
YES
B6
RETURN RC=4
C6
INSERT IMS/360 TRAN CODE & NL CHAR
A7
COMPUTED LGTH = INPUT LGTH?
NO
B6
YES
C7
RETURN RC=0
A8
ADD IMS/360 TRAN CODE & NL CHAR TO MSG
A9
RETURN RC=0


TR6
A0
LINE 12 TERMINAL 2?
YES
NO
B0
LENGTH VALID?
YES
NO
TR62
A1
LENGTH VALID?
NO
YES
B1
RETURN RC=4
A2
LINE 13?
YES
NO
B2
QSAM FILE ACTIVE?
YES
NO
D5
C2
WRITE MSG ON QSAM FILE
A3
BLOCKING REQUESTED?
NO
D5
YES
C3
C3
SET UP TO RETURN LGTH OF ZERO
D6
A4
MOVE MSG TO BLOCKING BUFFER
A5
BLOCKING BUFFER FILLED?
NO
C3
YES
A6
MOVE BLOCKING BUFFER TO I/O AREA
D5
D5
D5
INSERT IMS/360 TRAN CODE & NL CHAR
D6
D6
RETURN RC=0


SETSWI
A0
SWITCH CODE ZERO?
YES
NO
B0
OPEN QSAM FILE
C0
SET QSAM FILE-ACTIVE SWITCH
D0
SET UP TO RETURN ZERO LENGTH
A1
SWITCH CODE C?
B1
CLOSE QSAM FILE
C1
RESET QSAM FILE-ACTIVE SWITCH
A2
SWITCH CODE B?
B2
SET BLOCKING REQUESTED SWITCH
A3
SWITCH CODE D?
B3
RESET BLOCKING REQ'D SWITCH
C3
BLOCKING BUFFER EMPTY?
A4
SWITCH CODE INVALID
C4
MOVE BLKING BUFFER TO INPUT AREA
D4
RETURN RC=0
A5
RETURN RC=4

SAMPLE EDIT ROUTINE LISTING

The following pages are the assembler output from the preceding sample problem.

```
EDIT     TITLE 'DFS10300, 1030 SAMPLE EDIT ROUTINE.'
DFS10300 CSECT
         PRINT NOGEN
         SAVE  (14,12),,DFS10300
         USING MSGBUF,R1
         USING DFS10300,R12
         LR    R12,R15
         CLI   MSGLINE,12
         BE    CHTERM2         BRANCH IF INPUT FROM LINE 12
         CLI   MSGLINE,13
         BE    CHTERM1         BRANCH IF INPUT FROM LINE 13
RC4      RETURN (14,12),RC=4
CHTERM1  CLI   MSGTERM,1
         BE    OKTERM   BRANCH IF INPUT FROM TERMINAL # 1 ON THE LINE
CHTERM2  CLI   MSGTERM,2
         BNE   RC4  BRANCH IF INPUT NOT FROM TERMINAL # 2 ON THE LINE
OKTERM   CLI   MSGTRAN,C'1'
         BL    RC4             BRANCH IF INVALID 1030 TRANSACTION CODE
         BE    TR1             BRANCH IF 1030 TRANSACTION CODE IS 1
         CLI   MSGTRAN,C'3'
         BL    TR2             BRANCH IF 1030 TRANSACTION CODE IS 2
         BE    TR3             BRANCH IF 1030 TRANSACTION CODE IS 3
         CLI   MSGTRAN,C'5'
         BL    TR4             BRANCH IF 1030 TRANSACTION CODE IS 4
         BE    RC4             BRANCH IF INVALID 1030 TRANSACTION CODE
         CLI   MSGTRAN,C'7'
         BL    TR6             BRANCH IF 1030 TRANSACTION CODE IS 6
         BH    RC4             BRANCH IF INVALID 1030 TRANSACTION CODE
         CH    R2,=H'23' MANUAL ENTRY INPUT 13 BYTES; OVERHEAD 10 BYTES
         BNE   RC4             BRANCH IF INVALID LENGTH
         MVC   MSGTRAN+13(24),=C' FROM LINE 12 TERMINAL 2'
         CLI   MSGLINE,12
         BE    TR7A            BRANCH IF LINE 12
         MVI   MSGTRAN+25,C'3' CHANGE TO LINE 13
         CLI   MSGTERM,2
         BE    TR7A            BRANCH IF TERMINAL 2
         MVI   MSGTRAN+36,C'1' CHANGE TO TERMINAL 1
TR7A     LA    R2,24(,R2)      INCREASE LENGTH TO INCLUDE ADDED DATA
         MVC   MSGLINE(6),=C'T1033B' INSERT TRANSACTION CODE
RETURN   AR    R1,R2           INSERT NEW LINE
         MVI   0(R1),X'15'      CHARACTER
         LA    R2,1(,R2)       INCREASE LENGTH TO INCLUDE NEW LINE CHAR.
RC0      L     R14,12(R13)         LOAD RETURN ADDRESS
         RETURN (3,12),RC=0
         EJECT
TR1      LA    R5,22           CARD & BADGE & MANUAL ENTRY INPUT
         SR    R6,R6           POINT TO TRANSACTION CODE
         B     PACK
TR2      LA    R5,10           CARD & BADGE INPUT
         LA    R6,8            POINT TO TRANSACTION CODE
         B     PACK
TR3      LA    R5,12           CARD & MANUAL ENTRY INPUT
         LA    R6,16           POINT TO TRANSACTION CODE
         B     PACK
```

```
TR4      CH    R2,=H'12'
         BE    SETSWI          BRANCH IF BLOCKING SWITCH CARD
         SR    R5,R5           CARD ONLY INPUT
         LA    R6,24           POINT TO TRANSACTION CODE
PACK     LA    R3,MSGCARDL     POINT TO THE LENGTH FIELD IN THE MESSAGE
         LA    R4,3            SET UP THE LENGTH OF THE LENGTH FIELD
PACK1    CLI   0(R3),C'0'
         BL    RC4             BRANCH IF NON NUMERIC LENGTH FIELD
         CLI   0(R3),C'9'
         BH    RC4             BRANCH IF NON NUMERIC LENGTH FIELD
         LA    R3,1(,R3)       POINT TO THE NEXT BYTE IN THE LENGTH
         BCT   R4,PACK1 BRANCH IF THE LENGTH IS NOT COMPLETELY TESTED
         CLC   MSGCARDL,=C'004'
         BL    RC4      BRANCH IF THE LENGTH OF THE CARD IS TOO SHORT
         CLC   MSGCARDL,=C'160'
         BH    RC4      BRANCH IF THE LENGTH OF THE CARD IS TOO LONG
         PACK  DBLWORD,MSGCARDL  CONVERT THE LENGTH OF THE CARD
         CVB   R3,DBLWORD          TO BINARY
         LA    R3,10(R5,R3)    INCLUDE LEADING OVERHEAD & BADGE & MANUAL
         CR    R2,R3
         BNE   RC4             BRANCH IF INVALID LENGTH
         LA    R6,TRANCODE(R6) POINT TO TRANSACTION CODE
         MVC   MSGLINE(L'TRANCODE),0(R6) INSERT TRANSACTION CODE
         B     RETURN
         EJECT
TR6      CLC   MSGLINE(2),=X'0D02'
         BNE   TR62            BRANCH IF NOT LINE 13 TERMINAL 2
         CH    R2,=H'22'
         BE    TR61            BRANCH IF 1 BADGE
         CH    R2,=H'33'
         BE    TR61            BRANCH IF 2 BADGES
         CH    R2,=H'44'
         BNE   RC4             BRANCH IF NOT 3 BADGES
TR61     MVC   MSGLINE(6),=C'MASTER' INSERT TRANSACTION CODE
         B     RETURN
TR62     CH    R2,=H'21'
         BNE   RC4             BRANCH IF INVALID LENGTH
         CLI   MSGLINE,12
         BE    TR63            BRANCH IF INPUT FROM LINE 12
         TM    SWITCH,SWBLOCK
         BZ    TR61            BRANCH IF BLOCKING NOT REQUESTED
         LA    R3,BLKBUF       POINT TO BLOCKING BUFFER
         LH    R4,BLKOFF       LOAD OFFSET TO NEXT BUFFER POSITION
         AR    R3,R4           POINT TO NEXT BUFFER POSITION
         MVC   0(10,R3),MSGTRAN+1 MOVE INPUT MSG TO BLOCKING BUFFER
         MVI   10(R3),C','     INSERT SEPARATOR
         LA    R4,11(,R4)      UPDATE OFFSET
         STH   R4,BLKOFF       STORE OFFSET
         SR    R2,R2           SET LENGTH FOR NON-DATA RETURN
         CH    R4,=H'88'
         BNE   RC0             BRANCH IF BLOCKING BUFFER NOT FILLED
         STH   R2,BLKOFF       MAKE OFFSET ZERO
         MVC   MSGTRAN(88),BLKBUF MOVE BLOCKED MSGS TO INPUT BUFFER
         LA    R2,98           SET UP MESSAGE LENGTH
         B     TR61
TR63     TM    SWITCH,SWQSAM
         BZ    TR61            BRANCH IF QSAM FILE NOT ACTIVE
         LA    R3,MSGTRAN      POINT TO MESSAGE
         PUT   QSAMFILE,(R3)   WRITE QSAM FILE
         SR    R2,R2           SET LENGTH FOR NON-DATA RETURN
         B     RC0
         EJECT
```

```
SETSWI   CLI   MSGTRAN+1,C'O'
         BE    OPEN            BRANCH IF ACTIVATION OF QSAM FILE
         CLI   MSGTRAN+1,C'C'
         BE    CLOSE           BRANCH IF DEACTIVATION OF QSAM FILE
         CLI   MSGTRAN+1,C'B'
         BE    BLOCK           BRANCH IF START BLOCKING
         CLI   MSGTRAN+1,C'P'
         BE    PURGE           BRANCH IF STOP BLOCKING
         CLI   MSGTRAN+1,C'I'
         BNE   RC4             BRANCH IF INVALID SWITCH CODE
         RETURN (14,12),RC=8   GIVE INVALID RETURN CODE
OPEN     OPEN  (QSAMFILE,OUTPUT)
         OI    SWITCH,SWQSAM   INDICATE ACTIVE QSAM FILE
SWRET    SR    R2,R2           SET LENGTH FOR NON-DATA RETURN
         B     RC0
CLOSE    CLOSE (QSAMFILE)
         NI    SWITCH,X'FF'-SWQSAM INDICATE NON-ACTIVE QSAM FILE
         B     SWRET
BLOCK    OI    SWITCH,SWBLOCK INDICATE BLOCKING REQUESTED
         B     SWRET
PURGE    NI    SWITCH,X'FF'-SWBLOCK INDICATE BLOCKING NOT REQUESTED
         LH    R3,BLKOFF       LOAD LENGTH OF BLOCKED MSGS
         LTR   R3,R3
         BZ    SWRET           BRANCH IF NO MSGS BLOCKED
         EX    R3,MVC          MOVE BLOCKED MSGS TO INPUT BUFFER
         LA    R2,10(,R3)      SET UP MSG LENGTH
         B     TR61
MVC      MVC   MSGTRAN(0),BLKBUF
         EJECT
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
SWITCH   DC    X'00'
SWBLOCK  EQU   X'80'           BLOCKING REQUESTED
SWQSAM   EQU   X'40'           QSAM FILE ACTIVE
DBLWORD  DS    D               WORK AREA FOR CONVERT TO BINARY
TRANCODE DC    C'ERNE    '
         DC    C'MODEL2  '
         DC    C'L2740SM1'
         DC    C'L2740S2 '
BLKOFF   DC    H'0'
BLKBUF   DC    CL88' '
QSAMFILE DCB   BUFNO=4,DDNAME=QSAM1030,DSORG=PS,LRECL=11,MACRF=(PM),    C
               RECFM=FB
*        INPUT BUFFER DSECT
MSGBUF   DSECT
MSGLINE  DS    C               LINE NUMBER
MSGTERM  DS    C               TERMINAL NUMBER
         DS    CL8             BLANKS
MSGTRAN  DS    C               1030 TRANSACTION CODE
MSGCARDL DS    CL3             LENGTH OF CARD INPUT
         END
/*
```

7770-3 SIGNON EXIT ROUTINE - DFSS7770

Since the 7770-3 is a switched device and the calling terminal may not have the capability to generate the alphameric characters required to form a /IAM command to sign on for LTERM, IMS/360 requires that a signon routine be defined at system definition time for the 7770-3 lines in the system. This routine is invoked by the 7770-3 device-dependent module any time an input message or message segment is received from the line and a logical connection does not exist. Only one routine may be defined, and it applies to all 7770-3 lines in the system. A minimum user routine should validity check the input data received from the line and use the data to develop a /IAM command to be passed on to IMS/360. The user routine gains control before any IMS/360 security checking, validity checking, or editing functions are performed. The message text is in EDCDIC.

The signon routine may build a /IAM command in the input buffer or may place a response message in the input buffer. Any response to be sent back to the caller must be in 7770-3 output vocabulary drum address form.

Through return codes to the device module, the signon routine may cause the contents of the input buffer to be passed on into the system (/IAM command in buffer) or cause the contents of the buffer to be sent to the caller followed by a read to allow retry, or this routine may cause the contents of the input buffer to be sent to the caller with a reset to the line to disconnect the caller after the response is sent.

INTERFACE - DFSS7770

REGISTERS

On Entry:

R1	-	Address of input data received from line
R2	-	Length of input data
R7	-	CTB address
R8	-	CTT address
R9	-	CLB address
R11	-	SCD address
R13	-	Save area address. The first three words in the save area may not be changed.
R14-R15	-	Standard OS linkage registers
Note:		The save area chain fields for the save area pointed to by R13 on entry must not be modified.

Data Format

See Figure 10A for the data format at entry and R1 and R2 relationship to the data.

All registers must be restored except R0, R1, R2, and R15. The contents of R0 and R1 are ignored by the device-dependent routine.

R2 - Must contain the count of data now in the input buffer area that was pointed to by R1 on entry.

R15 - Contains the return code.

RETURN CODES

00 - Continue input processing with the contents of the input buffer.

04 - Send the contents of the input buffer to the caller, followed by a read. Allows retry of signon operation.

08 - Send the contents of the input buffer to the caller, followed by a disable to disconnect the caller.

Note: For return codes 04 and 08, the contents of the input buffer to be sent to the caller must be in drum address form, as no translation will be performed before the data is sent to the caller. It is also the user's responsibility to determine when a sequence of signon attempts should be terminated with a reset operation.

ERROR CONDITIONS

The line will be stopped and a message generated to the master terminal for either of the following signon routine error conditions:

1. The return code from the signon routine exceeds 8.
2. The count value returned in R2 is greater than the available space in the buffer.

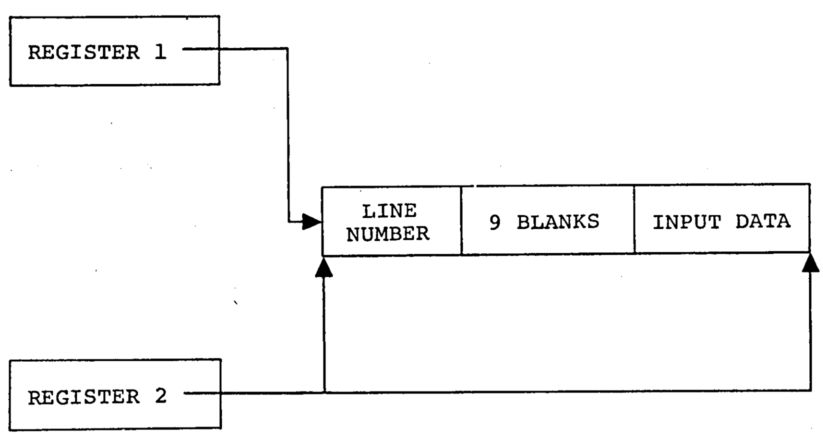


Figure 10A. 7770-3 Input data format for signon exit module interface

7770-3 SIGNON EXIT ROUTINE INCLUSION DURING SYSTEM DEFINITION

A usable signon routine is supplied with the system in IMS2.LOAD. This routine automatically signs the caller on for the INQUIRY LTERM whenever the 7770-3 answers a call and receives data. As supplied, this routine is transparent to the caller. If the supplied module is to be used, it is the user's responsibility to move the module from IMS2.LOAD to the user library specified in the IMSGEN statement before Stage 2 of system definition is executed.

7770-3 SIGNON EXIT ROUTINE PROGRAM LISTING

For further information on the IMS/360-supplied signon routine, see the _IMS/360 System Manual_, Volume I, (LYB0-0629) Description of Module DFSS7770. The source listing follows for reference.

```
STMT   SOURCE STATEMENT                                                  F15OCT70    3/27/72

     ************************************************************************ 00002000
     *                                                                      * 00003000
     *********************** 7770 AUTOMATIC INQUIRY ************************ 00004000
     ***********************     SIGN-ON MODULE     ************************ 00005000
     *                                                                      * 00006000
     *   THIS MODULE RECEIVES CONTROL FROM THE 7770 DEVICE DEPENDENT MODULE * 00007000
     *   WHEN A READ HAS COMPLETED BUT A LOGICAL CONNECTION HAS NOT BEEN    * 00008000
     *   ESTABLISHED.                                                       * 00009000
     *                                                                      * 00010000
     *   THIS MODULE SETS THE PROPER FLAGS AND FIELDS TO INDICATE THAT THE  * 00011000
     *   TERMINAL IS SIGNED ON FOR THE INQUIRY LOGICAL TERMINAL.            * 00012000
     *                                                                      * 00013000
     *   BLOCKS AND TABLES:                                                 * 00014000
     *                                                                      * 00015000
     *                    THIS MODULE USES THE CLB, CTB, AND THE CNT        * 00016000
     *                    THIS MODULE RECEIVES ACCESS TO THE INPUT DATA,    * 00017000
     *                    CTB, CTT, CLB, CNT, AND THE SCD.                  * 00018000
     *   SIZE OF MODULE:                                                    * 00019000
     *                    THIS MODULE CONTAINS APPROXIMATELY 54 BYTES OF    * 00020000
     *                    CODE.                                             * 00021000
     *   INTERFACE                                                          * 00022000
     *      REGISTERS                                                       * 00023000
     *        ON ENTRY:                                                     * 00024000
     *                    R1        ADDRESS OF INPUT DATA                   * 00025000
     *                    R2        LENGTH OF INPUT DATA                    * 00026000
     *                    R7        HAS DIAL CTB ADDRESS                    * 00027000
     *                    R8        HAS CTT ADDRESS                         * 00028000
     *                    R9        HAS CLB ADDRESS                         * 00029000
     *                    R11       HAS SCD ADDRESS                         * 00030000
     *                    R13-R15   STANDARD O.S. LINKAGE REGISTERS         * 00031000
     *                                                                      * 00032000
     *                    THE INPUT DATA AREA HAS THE FOLLOWING FORMAT:     * 00033000
     *                                                                      * 00034000
     *                    *-------------*                                   * 00035000
     *                    | REG 2 COUNT |----*----------------------------* * 00036000
     *                    *-------------*    |                            | * 00037000
     *                                       V     |        9            NV * 00038000
     *                                       *------+----------+---------* * 00039000
     *                                       | CTB  |          |  DATA   | * 00040000
     *                                    *->| LINE | 9 BLANKS |   IN    | * 00041000
     *                                    |  | NO.  |          | EBCDIC  | * 00042000
     *                                    |  *------+----------+---------* * 00043000
     *                    *-------------* |                                 * 00044000
     *                    | REG | ADDR. |-*                                 * 00045000
     *                    *-------------*                                   * 00046000
     *                                                                      * 00047000
     *                                                                      * 00048000
     *        ON EXIT:                                                      * 00049000
     *                    R2        LENGTH OF DATA IN BUFFER                * 00050000
     *                    R15       RETURN CODE                             * 00051000
     *                    ALL OTHER REGISTERS ARE RESTORED.                 * 00052000
     *                                                                      * 00053000
     *      RETURN CODES:                                                   * 00054000
     *                                                                      * 00055000
     *         0 - CONTINUE INPUT PROCESSING WITH CONTENTS OF THE BUFFER    * 00056000
```

```
STMT   SOURCE STATEMENT                                                   F15OCT70    3/27/72

  57 *           4 - SEND CONTENTS OF BUFFER TO CALLER FOLLOWED BY READ TO   * 00057000
  58 *               ALLOW RETRY.  OUTPUT MUST BE IN DRUM ADDRESS FORM.      * 00058000
  59 *           8 - SEND CONTENTS OF BUFFER TO CALLER FOLLOWED BY A DISABLE * 00059000
  60 *               TO DISCONNECT THE CALLER.                               * 00060000
  61 *                                                                       * 00061000
  62 *  ABENDS:                                                              * 00062000
  63 *                                                                       * 00063000
  64 *     NCT APPLICABLE                                                    * 00064000
  65 *                                                                       * 00065000
  66 ************************************************************************ 00066000
```

```
STMT   SOURCE STATEMENT                                                   F15OCT70    3/27/72

  68 DFSS777C CSECT                                                         00068000
  69 *                                                                      00069000
  70          USING *,R12                  BASE REGISTER                    00070000
  71          USING SCD,R11                --> SCD                          00071000
  72          USING CNT,R10                --> CNT (INQUIRY)                00072000
  73          USING IECTDECB,R9            --> CLB                          00073000
  74          USING CTT,R8                 --> CTT                          00074000
  75          USING CTB,R7                 --> CTB                          00075000
  76 *                                                                      00076000
  77          SAVE  (14,12),,S77A130       SAVE CALLER'S REGISTERS          00077000
  78+         B     12(0,15) BRANCH AROUND ID
  79+         DC    AL1(7) LENGTH OF IDENTIFIER
  80+         DC    CL7'S77A130' IDENTIFIER
  81+         STM   14,12,12(13) SAVE REGISTERS
  82 *                                                                      00078000
  83          LR    R12,R15                                                 00079000
  84          LH    R10,CTBCNTP            GET INQUIRY CNT OFFSET           00080000
  85          A     R10,SCDCNT             ADD BASE FOR CNT ADDRESS         00081000
  86          ST    R10,60(,R13)           SET CNT ADDRESS TO BE PASSED BAC 00082000
  87          NI    CTBFLAG1,NIMASK-CTB1PCNT-CTB1PSMB RESET PRESET FLAGS    00083000
  88          NI    CTBFLAG2,NIMASK-CTB2LOCK-CTB2TEST-CTB2EXCL  + OTHERS    00084000
  89          OI    CTBFLAG1,CTB1SIGN      DIAL CTB IS LOGICALLY CONNECTED  00085000
  90          OI    CNTFLAG1,CNT1SIGN      SIGN ON LTERM ONLY SPECIFICATION 00086000
  91          MVC   CLBCTBPS(2),CNTCTBP    POINT CLB TO CTB                 00087000
  92 *                                                                      00088000
  93          RETURN (14,12),RC=0          RESTORE AND RETURN TO DEVICE MOD 00089000
  94+         LM    14,12,12(13) RESTORE THE REGISTERS
  95+         LA    15,0(0,0) LCAD RETURN CODE
  96+         BR    14 RETURN
```

```
STMT   SOURCE STATEMENT                                                   F15OCT70    3/27/72

  98 *             *                                                        00091000
  99 *** EQUATES ***                                                        00092000
 100 *             *                                                        00093000

 102 NIMASK   EQU   255                    ALL BITS                         00095000
 103 R0       EQU   00                     R                                00096000
 104 R1       EQU   01                     E                                00097000
 105 R2       EQU   02                     G                                00098000
 106 R3       EQU   03                     I                                00099000
 107 R4       EQU   04                     S                                00100000
 108 R5       EQU   05                     T                                00101000
 109 R6       EQU   06                     E                                00102000
 110 R7       EQU   07                     R                                00103000
 111 R8       EQU   08                     S                                00104000
 112 R9       EQU   09                                                      00105000
 113 R10      EQU   10                     E                                00106000
 114 R11      EQU   11                     Q                                00107000
 115 R12      EQU   12                     U                                00108000
 116 R13      EQU   13                     A                                00109000
 117 R14      EQU   14                     T                                00110000
 118 R15      EQU   15                     E                                00111000
```

```
 120 *                      *                                                  00113000
 121 *** DUMMY SECTIONS ***                                                    00114000
 122 *                      *                                                  00115000

 124          ISCD   SCDBASE=0                                                 00117000
 125+SCD      DSECT
 126+*** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ***
 127+** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **
 128+**                                                                       **
 129+**              IMS SYSTEM CONTENTS DIRECTORY  (SCD)                     **
 130+**                                                                       **
 131+** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **
 132+*** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ***

 134+SSCDORG  DS     0D SECONDARY SCD ORIGIN
 135+SSCDSVC1 SVC    0 USER TYPE 1 SVC NUMBER
 136+         BR     14 RETURN ON REGISTER 14
 137+SSCDSVC2 SVC    0 USER TYPE 2 SVC NUMBER
 138+         BR     14 RETURN ON REGISTER 14
 139+SSCDSVC3 SR     15,15 CLEAR REGISTER
 140+         BR     14 RETURN ON REGISTER 14
 141+SSCDSVC4 SVC    255 USER TYPE 4 SVC NUMBER
 142+         BR     14 RETURN ON REGISTER 14
 143+SSCDOCEP DC     CL2'Z9' OSAM CHANNEL END APPENDAGE SUFFIX
 144+SSCDOSPS DC     X'10' OS/360 SYSTEM CONFIGURATION
 145+SSCDOSLV DC     AL1(18) OS/360 RELEASE LEVEL SPECIFIED
 146+SSCDIMSR DC     AL1(0) IMS RELEASE NUMBER.
 147+SSCDIMSL DC     AL1(0) IMS MOD LEVEL NUMBER.
 148+SSCDASPS DC     X'00' SYSTEM CONFIGURATION - ACTUAL
 149+SSCDASLV DC     AL1(0) OS/360 RELEASE LEVEL - ACTUAL
 150+SSCDDESC DC     AL1(0) MCS DESCRIPTOR CODE
 151+         DC     AL3(0) RESERVED FOR OS/360 EXPANSION
 152+SSCDROUT DC     AL2(0) MCS ROUTING CODES
 153+         DC     AL2(0) RESERVED FOR OS/360 EXPANSION
 154+DFSXWAIT DC     V(DFSIWAIT) IWAIT EP ADDRESS
 155+DFSXIOBA DC     V(DFSIOS40) OSAM IOB QCB
 156+DFSXOPEN DC     A(DFSXOPEN) OSAM OPEN EP
 157+SSCDMCID DC     AL1(0) CONSOLE ID - START IMS COMMAND.
 158+         DC     AL3(0) RESERVED
 159+         DC     F'0' RESERVED
```

```
162+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
163+*                                                                            *
164+*        R E C O R D E R    L O G    S E C T I O N :                         *
165+*                                                                            *
166+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

168+SCDREENT DC     V(DFSFLOIO) RECORDER ENTRY POINT
169+SCDREDCB DC     A(0) DCB AND WKAREA LIST ADDRESS
170+SCDREWRK DC     V(DFSILOGW) WORK AREA FOR THE LOGGER
171+SCDREPLN DC     H'28' LENGTH OF LOG PREFIX IN BYTES

173+*** LOG CONTROL BYTE ***
174+SCDRECTL DC     X'0000'
175+SCDLGOPN EQU    X'80' LOG IS OPEN
176+SCDLGTRM EQU    X'40' LOG SMBTASK DETACHED

178+SCDRELCT DC     H'500' CHECKPOINT LOG FREQUENCY CONTROL REFERENCE
179+SCDRECCT DC     H'500' CURRENT LOG FREQUENCY COUNTER, CPT AT ZERO
180+SCDRPENT DC     A(0) ENTRY TO LOG WRITER
181+SCDRTECB DC     F'0' LOG TASK TERMINATE ECB
182+SCDRTCB  DC     A(0) LOG TASK CONTROL BLOCK ADDRESS
183+SCDRETXR DC     A(0) RESERVED
184+SCDRPFX  DC     V(DFSIPREF) ENTRY TO LOG PREFIX BUILDER
185+SCDDBLNT DC     A(SCDDBLNT) ENTRY TO DATABASE CHANGE LOGGING ROUTINE
186+SCDCWRK  DC     F'0' CHECKPOINT/DB LOG WORK AREA ADDRESS
187+SCDCWRKL DC     H'512' LENGTH OF CHECKPOINT/DB LOG WORK AREA
188+SCDCPNO  DC     H'0' CHECKPOINT NUMBER
189+SCDDATE  DC     F'0' DATE FOR RESTART/DATABASE LOG
190+SCDTIME  DC     F'0' TIME FOR RESTART/DATABASE LOG
191+SCDSEQ   DC     H'0' DATABASE LOG SEQUENCE NUMBER
192+SCDRGTYP DC     AL1(0,0) REGION TYPE (0=CTL,3=DLI)
193+SCDDBDCB DC     A(0) ADDRESS OF DATABASE LOG DCB
194+         DC     4F'0' RESERVED FOR SYSTEM EXPANSION
```

```
196+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
197+*                                                                             *
198+*  D A T A    L A N G U A G E / I    &    O S A M    S E C T I O N :        *
199+*                                                                             *
200+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

202+SCDDLICT DC     A(SCDDLICT) DL/I ANALYZER ENTRY POINT (DFSDLA00)
203+SCDDLIMQ DC     A(0) RESERVED
204+SCDISAM  DC     A(SCDISAM) ISAM SIMULATOR ENTRY POINT
205+SCDDLARE EQU    22 RETURN POINT FROM ANALYZER ENTRY POINT
206+SCDDLIRE DC     A(SCDDLIRE) RETRIEVE MODULE ENTRY POINT
207+SCDDLIIN DC     A(SCDDLIIN) INSERT MODULE ENTRY POINT
208+SCDDLIDR DC     A(SCDDLIDR) DELETE/REPLACE MODULE ENTRY POINT
209+SCDDLDAE EQU    18 ALTERNATE ENTRY TO DELETE/REPLACE
210+SCDDLIHS DC     A(SCDDLIHS) HSAM ENTRY POINT
211+SCDDLIPS DC     V(DFSIDIR0) BEGINNING OF PSB DIRECTORY ENTRIES
212+SCDPSBFR DC     A(0) EP TP PSB FREE POOL SPACE ROUTINE
213+SCDDLILN DC     H'40' LENGTH OF PSB DIRECTORY ENTRY
214+SCDDLINC DC     H'0' NUMBER OF PSB DIRECTORY ENTRIES
215+SCDPSBSW DS     0XL1 0 - FREE PSB POOL SPACE
216+SCDDLIDM DC     A(0) BEGINNING OF DMB DIRECTORY ENTRIES
217+SCDDLIDL DC     H'36' LENGTH OF EACH DMB DIRECTORY ENTRY
218+SCDDLIDN DC     H'0' NUMBER OF DMB DIRECTORY ENTRIES
219+SCDDLIPA DC     V(DFSIPST) START OF PST BLOCKS
220+SCDDLIPL DC     H'424' LENGTH OF EACH PST
221+SCDDLIPN DC     H'0' NUMBER OF PST ENTRIES
222+SCDDLIBC DC     A(0)
223+SCDDMBSW DS     0XL1 0 - RELEASE DMB SPACE, 1 - RESERVED
224+SCDDMBFR DC     A(0) ENTRY TO DMB FREE POOL SPACE ROUTINE
225+SCDPSBMU DC     V(DFSIPSBQ) PSB MOST USED QCB ADDRESS
226+SCDDPDM  DC     V(DFSDPDM0) PSB/DMB POOL MANAGER
227+SCDACBDC DC     V(ACBDCB) ACBLIB DCB POINTER
228+SCDPSBPL DC     F'0'
229+SCDDMBPL DC     F'0'
230+SCDLLOGD DC     7F'0'
231+SCDDLIMV DC     V(DFSDBLM0) BLOCK MOVER ENTRY POINT
232+SCDDSST  DC     V(DFSDSST0) SEGMENT INTENT RESERVATION CSECT
233+SCDDLICL DC     A(SCDDLICL) DATA MANAGEMENT OPEN/CLOSE ENTRY POINT
234+SCDDLI07 DC     A(SCDDLIC7) ENTRY TO DL/I LOAD PROGRAM
235+SCDWQCB  DC     V(DFSPWQCB) DL/I MOVER QCB ADDRESS
236+SCDBPARM DC     A(0) DL/I REGION PARMS (PXPARMS)
237+SCDASE   DC     V(DFSIASE0) APPLICATION SCHEDULER TERMINATOR EP
238+SCDDBFPL DS     A LOCATION OF DL/I BUFFER POOL
239+SCDDDBHC DC     A(0) ENTRY POINT OF DL/I BUFFER HANDLER
240+SCDDHDSO DC     A(0) ENTRY POINT OF GET/FREE SPACE
241+SCDDXMTC DC     A(0) ENTRY POINT OF INDEX MAINTENANCE
242+SCDDBCVT DC     A(0) ENTRY POINT OF BYTE BLOCK CONVERSION
243+SCDOIOBS EQU    DFSXIOBA OSAM IOB QCB
244+SCDOOPEN EQU    DFSXOPEN OSAM OPEN EP
245+SCDIWAIT EQU    DFSXWAIT IWAIT ENTRY POINT ADDRESS
246+SCDSAVE  DC     V(DFSISAV0) ORIGIN OF SAVE AREA SET
247+SCDFLOS  DC     A(0) STAE EXIT ROUTINE ENTRY POINT
248+SCDFLOSR DC     A(0) STAE RETRY ROUTINE ENTRY POINT
249+SCDNAVIC DS     0F
250+SCDRLDTE DC     X'00'
```



```
251+SCDSYNCN DC     AL3(0)
252+SCDLOWID DC     F'0'
253+         DC     6F'0' RESERVED FOR SYSTEM EXPANSION
```

```
255+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
256+*                                                                         *
257+*      S T O R A G E   M A N A G E M E N T   S E C T I O N :              *
258+*                                                                         *
259+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

261+SCDSMMLC DC     A(0) ADDRESS OF START OF SMM CORE AREA
262+SCDSMML  DC     F'0' LENGTH OF STORAGE MANAGEMENT CORE AREA
263+SCDSMMHI DC     A(0) ADDRESS OF END OF SMM CORE AREA
264+SCDSMMGC DC     A(0)
265+SCDSMMFC DC     A(0)
266+SCDSMMCP DC     V(DFSICRET) ICREAT ENTRY POINT
267+SCDSMMDP DC     V(DFSIDEST) IDESTROY ENTRY POINT
268+SCDSMMFB DC     V(DFSIFBUF) IFREEBUF ENTRY POINT
269+SCDSMMGB DC     V(DFSIGBUF) IGETBUF ENTRY POINT
270+SCDILMSK DC     X'0001' PROGRAM LOAD MASK
271+         DS     XL2 NOT USED
272+         DS     3F PADDING



274+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
275+*                                                                         *
276+*        E N Q U E U E / D E Q U E U E   S E C T I O N :                  *
277+*                                                                         *
278+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

280+SCDIDEQ  DC     V(DFSIIDEQ) IDEQ ENTRY POINT
281+SCDIENQ  DC     V(DFSIIENQ) IENQ ENTRY POINT
282+         DS     2F
```



```
284+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
285+*                                                                         *
286+*        Q U E U E   M A N A G E M E N T   S E C T I O N :
287+*                                                                         *
288+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

290+SCDIRWQE DC     V(DFSQMGRO) IREAD/IWRITE QUEUE ENTRY POINT
291+SCDIQCCB DC     V(DFSIQDCB) ADDRESS OF INPUT QCR DCB
292+SCDSMDCB DC     V(DFSSMDCB) ADDRESS OF IMS2.SHMSG DCB
293+SCDLMDCB DC     V(DFSLMDCB) ADDRESS OF IMS2.LGMSG DCB
294+SCDSPDCB DC     V(DFSICDCB) ADDRESS OF SCRATCH PAD AREA (SPA) DCB
295+SCDQPOOL DC     A(0) POINTER TO QUEUE BLOCK POOL
296+SCDPRLN1 DC     H'40' LENGTH OF FULL MESSAGE PREFIX
297+SCDPRLN2 DC     H'16' LENGTH OF CONDENSED MESSAGE PREFIX
298+SCDQCRSZ DC     H'0' SIZE OF QCR BUFFER
299+SCDMSGSZ DC     H'0' SIZE OF MSG BUFFER
300+SCDQMREB DC     F'0' SAVE AREA PTR TO BUFFER ALLOC FOR REUSE
301+SCDCKCL  DC     F'0' COMM FOR INTERNAL CHECKPOINTS
302+SCDQMSRB DC     H'0' RECORDS RESERVED FOR SHUTDOWN/REUSE
303+         DC     5H'0' RESERVED FOR SYSTEM EXPANSION
```

```
305+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
306+*                                                                           *
307+*             C O M M U N I C A T I O N     S E C T I O N :                 *
308+*                                                                           *
309+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

311+SCDCTT     DC     V(DFSICTT) COMMUNICATION TRANSLATION TABLE BASE ADDR
312+SCDCTTL    DC     H'28' LENGTH OF EACH CTT
313+SCDCTTN    DC     H'0' NUMBER OF CTTS
314+SCDCLB     DC     V(DFSICLB) COMMUNICATION LINE BLOCK BASE ADDRESS
315+SCDCLBL    DC     H'100' LENGTH OF EACH CLB
316+SCDCLBN    DC     H'0' NUMBER OF CLBS
317+SCDCTB     DC     V(DFSICTB) COMMUNICATION TERMINAL BLOCK BASE ADDRESS
318+SCDCTBL    DC     H'32' LENGTH OF EACH CTB
319+SCDCTBN    DC     H'0' NUMBER OF CTBS
320+SCDCXB     DC     A(0) COMMUNICATION EXTENT BLOCK BASE ADDRESS
321+SCDCXBL    DC     H'20' LENGTH OF EACH CXB
322+SCDCXBN    DC     H'0' NUMBER OF CXBS
323+SCDCNT     DC     V(DFSICNT) COMMUNICATION NAME TABLE BASE ADDRESS
324+SCDCNTL    DC     H'44' LENGTH OF EACH CNT
325+SCDCNTN    DC     H'0' NUMBER OF CNTS
326+SCDCDB     DC     V(DFSICDB) COMMUNICATION DIRECTORY BLOCK BASE ADDR
327+SCDCDBL    DC     H'4' LENGTH OF A CDB
328+SCDCDBN    DC     H'0' NUMBER OF CDBS
329+SCDCVB     DC     V(DFSICVB) COMMUNICATION VERB BLOCK BASE ADDRESS
330+SCDCVBL    DC     H'12' LENGTH OF EACH CVB
331+SCDCVBN    DC     H'0' NUMBER OF CVBS
332+SCDCPT     DC     A(0) COMMUNICATION PRIORITY TABLE BASE ADDRESS
333+SCDCPTL    DC     H'0' LENGTH OF EACH CPT
334+SCDCPTN    DC     H'0' NUMBER OF CPTS
335+SCDCPM     DC     A(0) COMMUNICATION PRIORITY MATRIX BASE ADDR
336+SCDCPML    DC     H'0' LENGTH OF EACH CPM
337+SCDCPMN    DC     H'0' NUMBER OF CPMS
338+SCDCTM     DC     V(DFSICTM) COMMUNICATION TERMINAL MATRIX BASE ADDR
339+SCDCTML    DC     H'0' LENGTH OF EACH CTM
340+SCDCTMN    DC     H'0' NUMBER OF CTMS
341+SCDCCB     DC     A(0) CONVERSATIONAL CONTROL BLOCK BASE ADDRESS
342+SCDCCBL    DC     H'32' CCB LENGTH
343+SCDCCBN    DC     H'0' NUMBER OF CCBS
344+SCDCCBND   DC     AL2(0) OFFSET TO FIRST DISK CCB
345+SCDCCBNC   DC     AL2(0) OFFSET TO FIRST INCORE CCB
346+SCDCCBDL   DC     H'0' MAXIMUM LENGTH OF DISK SPA
347+SCDCCBCL   DC     H'0' MAXIMUM LENGTH OF INCORE SPA
348+SCDCAN01   DC     V(DFSCIOC1) COMMUNICATION ANALYZER ENTRY # 1
349+SCDCAN02   DC     V(DFSCIOC2) COMMUNICATION ANALYZER ENTRY # 2
350+SCDCAN03   DC     V(DFSCIOC3) COMMUNICATION ANALYZER ENTRY # 3
351+SCDCAN04   DC     V(DFSCIOC4) COMMUNICATION ANALYZER ENTRY # 4
352+SCDCAN05   DC     V(DFSCIOC5) COMMUNICATION ANALYZER ENTRY # 5
353+SCDCAN06   DC     V(DFSCIOC6) COMMUNICATION ANALYZER ENTRY # 6
354+SCDCAN07   DC     V(DFSCIOC7) COMMUNICATION ANALYZER ENTRY # 7
355+SCDCAN08   DC     V(DFSCIOC8) COMMUNICATION ANALYZER ENTRY # 8
356+SCDCAN09   DC     V(DFSCIOC9) COMMUNICATION ANALYZER ENTRY # 9
357+SCDCAN10   DC     V(DFSCIO10) COMMUNICATION ANALYZER ENTRY # 10
358+SCDCAN11   DC     V(DFSCIO11) COMMUNICATION ANALYZER ENTRY # 11
359+SCDFNDST   DC     V(DFSICLFO) FIND DESTINATION ROUTINE ENTRY POINT
```

```
 360+SCDMSGRT DC    V(DFSICLRO) MESSAGE ROUTER ROUTINE ENTRY POINT
 361+SCDMTRM  DC    V(DFSCTBMT) MASTER TERMINAL CTB ADDRESS
 362+SCDCLISO DC    V(DFSICLSO) SECURITY ROUTINE ENTRY POINT
 363+SCDTRANS DC    V(DFSICLTO) TRANSLATE ROUTINE ENTRY POINT
 364+SCDCONV  DC    A(0) CONVERSATION PROCESSOR ENTRY POINT
 365+SCDCOB   DC    A(0) COMMUNICATION OVERFLOW BLOCK BASE ADDRESS
 366+SCDCOBL  DC    H'32' LENGTH OF EACH COB
 367+SCDCOBN  DC    H'0' NUMBER OF COBS
 368+SCDCTCLB DC    V(DFSICLB) CONSOLE TYPEWRITER CLB
 369+SCDREPOL DC    V(DFSICLXO) RESET POLL MODULE ENTRY POINT
 370+SCDNTB   DC    V(DFSINTEO)
 371+SCDEDITC DC    A(0) USER EDIT ROUTINE FOR CNT DESTINATIONS
 372+SCDEDITS DC    V(DFSLIST0) USER EDIT ROUTINES FOR SMB DESTINATIONS
 373+SCDINCT  DC    H'0'
 374+SCDOUTCT DC    H'0'
 375+SCDCPCTL DC    X'00' CHECKPOINT POST SYNC BITS
 376+SCDPCP01 EQU   X'80'
 377+SCDPCP02 EQU   X'40'
 378+SCDPCP03 EQU   X'20'
 379+         DC    AL3(0) RESERVED FOR SYSTEM EXPANSION
 380+         DC    9F'0' RESERVED FOR SYSTEM EXPANSION
```



```
 382+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 383+*                                                                              *
 384+*               D I S P A T C H E R     S E C T I O N :                        *
 385+*                                                                              *
 386+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

 388+SCDEXTWQ DC    V(DFSEXTWQ) EXTERNAL WAIT QUEUE ADDRESS
 389+SCDINTWQ DC    V(DFSINTWQ) INTERNAL WAIT QUEUE ADDRESS
 390+SCDINTSQ DC    V(DFSINTSQ) INTERNAL SAVE QUEUE ENTRY POINT ADDRESS
 391+SCDTECBR DC    V(DFSITERM) TERMINATION ROUTINE ENTERED IF TECB POST
 392+SCDPARM  DC    A(0) ADDRESS OF PARM LIST FROM SYSTEM
 393+SCDTECB  DC    F'0' IMS TERMINATION ECB POSTED BY CHECKPOINT
 394+SCDSAVEL DC    H'508' LENGTH OF SAVE AREA SETS INCLUDING QE
 395+SCDSLEV  EQU   7 NUMBER OF SAVE AREAS IN A SET
 396+SCDSAVEN DC    H'0' NUMBER OF SAVE AREA SETS
 397+SCDSRQE  DC    V(DFSIRQEO) ORIGIN OF RQE LIST
 398+SCDSRQEL DC    H'8' LENGTH OF EACH RQE
 399+SCDSRQEN DC    H'0' NUMBER OF RQES
 400+SCDWAITQ DC    V(DFSIWATQ) ORIGIN OF WAIT QUEUE
 401+SCDEXTWL DC    H'4' LENGTH OF EACH ENTRY IN EXT WAIT QUEUE
 402+SCDEXTWN DC    H'0' NUMBER OF EXTERNAL EVENTS
 403+SCDSUBSV DC    A(0) ADDRESS OF LAST SUBTASK 1ST LEV SAVE AREA
 404+SCDCFAC  DC    X'00' CURRENT FIRST LEVEL FACILITY
 405+SCDCMOD  DC    X'00' CURRENT MODULE WITHIN FACILITY
 406+SCDTEST  DC    X'80' TEST MODE CONTROL INDICATOR
 407+SCDTLEV  DC    X'00' TEST MODE LEVEL INDICATOR
 408+SCDRES2  DC    F'0' PADDING
 409+         DC    4F'0' RESERVED FOR SYSTEM EXPANSION
```

```
411+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
412+*                                                                                     *
413+*    A P P L I C A T I O N    S C H E D U L E R    S E C T I O N :                    *
414+*                                                                                     *
415+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

417+SCDASINT DC     V(DFSASINT) APPLICATION SCHEDULER INITIATOR ENTRY
418+SCDASTER DC     V(DFSASTRM) APPLICATION SCHEDULER TERMINATOR ENTRY
419+SCDSMBEP DC     V(DFSISMB) SYSTEM MESSAGE BLOCKS BASE ADDRESS
420+SCDSMBL  DC     H'56' LENGTH OF EACH SMB
421+SCDSMBN  DC     H'0' NUMBER OF SMBS
422+SCDSMBDQ DC     V(DFSSMBDQ) SMB DEQUEUE ENTRY POINT ADDRESS
423+SCDSMBNQ DC     V(DFSSMBEQ) SMB ENQUEUE ENTRY POINT ADDRESS
424+SCDCHKXB DC     A(SCDCHKEB) ADDRESS OF ECB TO BE POSTED BY DFSIAS00
425+SCDTCTEP DC     V(DFSITCT) TRANSACTION CLASS TABLES BASE ADDRESS
426+SCDTCTL  DC     H'80' LENGTH OF EACH TCT
427+SCDTCTN  DC     H'0' NUMBER OF TCTS
428+SCDMRQEP DC     V(DFSIMRQ) ADDRESS OF MESSAGE REQUEST QUEUE
429+SCDPRQEP DC     V(DFSIPRQ) ADDRESS OF PARTITION REQUEST QUEUE
430+SCDBPRQE DC     V(DFSIBPRQ) ADDRESS OF BATCH PARTITION REQUEST QUEUE
431+SCDEXTSQ DC     V(DFSEXTSQ) EXTERNAL SAVE QUEUE ENTRY POINT ADDRESS
432+         DC     4F'0'
```

```
434+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
435+*                                                                       *
436+*   C H E C K P O I N T / R E S T A R T    S E C T I O N :              *
437+*                                                                       *
438+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

440+*** CHECKPOINT STATUS BYTE ***
441+SCDCKCTL DC     X'00'
442+*                        DESCRIPTION OF BIT SETTINGS
443+*                             0  NOT USED
444+*                             1  DATABASE RECOVERY
445+*                             2  PURGE REQUEST
446+*                             3  SYSTEM SHUTDOWN REQUEST
447+*                             4  DUMP QUEUE REQUEST
448+*                             5  MASTER TERMINAL CHECKPOINT REQUEST
449+*                             6 DATABASE DUMP REQUEST
450+*                             7  SYSTEM SCHEDULED CHECKPOINT (LOG)

452+*** RESTART STATUS BYTE ***
453+SCDRSCTL DC     X'00'
454+*                        DESCRIPTION OF BIT SETTINGS
455+*                             0  PASSWORD SECURITY REQUESTED
456+*                             1  QUEUE ONLY RESTART
457+*                             2  PSB/DMB CHANGED - REBUILD BLOCKS
458+*                             3  ALLOW COMMANDS EXCEPT RESTART
459+*                             4  TERMINAL SECURITY REQUESTED
460+*                             5  EMERGENCY RESTART
461+*                             6 BUILD QUEUES
462+*                             7  NORMAL RESTART

464+*** COMMON SYSTEM SHUTDOWN STATUS BYTE ***
465+SCDSTOP1 DC     X'0000'
466+*               BYTE 1   DESCRIPTION OF BIT SETTINGS - USER
467+*                         0  STOP INPUT                COMMUNICATIONS
468+*                         1  STOP OUTPUT               COMMUNICATIONS
469+*                         2  SEND ALL OUTPUT           COMMUNICATIONS
470+*                         3  PROCESS ALL MESSAGES      SCHEDULER
471+*                         4  QUEUE'S AVAILABLE         READ/WRITE Q
472+*                         5  FREE MESSAGE REGIONS      SCHEDULER
473+*                         6  TERMINATE ALL REGIONS     SCHEDULER
474+*                         7  STOP SMB INPUT            COMMUNICATIONS
475+*
476+*               BYTE 2   DESCRIPTION OF BIT SETTINGS - USER
477+*                         0  PROCESS AT SYSTEM PRIORITY SCHEDULER
478+*                         1  DATABASE STOP              CHECKPOINT
479+*                         2  FORCE EOV ON LOG FILE      CHECKPOINT
480+*                         3  ABEND AT TERMINATION       COMMUNICATIONS
481+*                         4  DISCONNECT ALL LINES       COMMUNICATIONS
482+*                         5  ALL OUTPUT SENT            COMMUNICATIONS
483+*                         6  SYSTEM CLOSE DOWN          CHECKPOINT
484+*                         7  FINAL ENTRY CONTROL        CHECKPOINT

486+SCDCKDBN DS     0CL1 # OF ENTERED DATABASE NAMES FOR RECOVERY
487+SCDCKDB  DC     A(0) POINTER TO POOL CONTAINING DATABASE NAMES
488+SCDLOGN  DS     0CL1 COUNT OF RESTART SERIAL NUMBERS
```

```
 489+SCDLOG    DC    A(0) POINTER TO POOL CONTAINING SERIAL NUMBERS
 490+SCDCLIPN  DC    H'0000' INPUT TERMINAL ACTIVE COUNT
 491+SCDCLOPN  DC    H'0000' OUTPUT TERMINAL ACTIVE COUNT
 492+SCDCPC01  DC    A(0) SCRATCH PAD AREA LOG ROUTINE ENTRY POINT
 493+SCDCHKEP  DS    0F
 494+SCDRSTEP  DC    V(DFSRSTC0) RESTART ROUTINE ENTRY POINT ADDRESS
 495+SCDCHKEB  DS    0F
 496+SCDRSTEB  DC    V(DFSRTECB) RESTART ECB ADDRESS
 497+SCDDBTBL  DC    A(0) RESTART DATABASE NAME TABLE
 498+          DC    4F'0'
```



```
 500+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 501+*                                                                       *
 502+*      S T A R T / S T O P    R E G I O N    S E C T I O N :            *
 503+*                                                                       *
 504+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

 506+SCDREGOP  DC    A(0) ECB TO OPEN REGION
 507+SCDREGCL  DC    A(0) ECB TO CLOSE REGION
 508+SCDCMDCT  DC    X'00' # OF START/STOP REGION COMMANDS ISSUED
 509+SCDREGST  DC    X'00' # OF PENDING STOP COMMANDS
 510+SCDREGCT  DC    H'0000' COUNT OF OPEN REGIONS
 511+SCDSRDR   DC    A(SCDPROC1) ADDRESS OF START READER COMMAND
 512+SCDSRMBR  DC    A(SCDPROC2) ADDRESS OF MEMBER FIELD IN START READER
 513+SCDSMMBR  DC    A(SCDPROC3) ADDRESS OF MSG REG PROC MEMBER NAME
 514+SCDSINIT  DC    A(SCDPROC4) ADDRESS OF START INITIATOR COMMAND
 515+SCDPINIT  DC    A(SCDPROC5) ADDRESS OF STOP INITIATOR COMMAND
 516+          DC    2F'0'
 517+SCDPSYS   EQU   SSCDOSPS PROGRAMMING SYSTEM OPTION BYTE FROM CVT
 518+SCDPSMVT  EQU   X'10' VALUE IF MVT
 519+SCDPSMFT  EQU   X'20' VALUE IF MFT-II
 520+SCDPSPCP  EQU   X'40' VALUE IF PCP
 521+SCDIMSR   EQU   SSCDIMSR IMS RELEASE NUMBER
 522+SCDIMSL   EQU   SSCDIMSL IMS MOD LEVEL
 523+          DC    2F'0' RESERVED FOR SYSTEM EXPANSION
```

```
525+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
526+*                                                                       *
527+*        C O M M A N D    S E C T I O N :                               *
528+*                                                                       *
529+*                   *** NOTE ***                                        *
530+*                           THE COMMAND SECTION MUST RESIDE LAST        *
531+*                           IN THE SYSTEM CONTENTS DIRECTORY SINCE      *
532+*                           THE COMMANDS ARE VARIABLE IN LENGTH.        *
533+*                                                                       *
534+*                           ANY ADDITION TO THE SCD                     *
535+*                           MUST PRECEDE THIS SECTION.                  *
536+*                                                                       *
537+* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

539+SCDPROC1 DS    0F
540+         DC    AL2(IHB0C05-*) MESSAGE LENGTH
541+         DC    B'0000000000000000' MCSFLAGS FIELD
542+         DC    C'S IMSRDR,MBR=' MESSAGE
543+IHB0005  EQU   *
544+SCDPROC2 DS    CL9 MEMBER NAME FIELD OF START READER COMMAND
545+SCDPROC3 DC    AL1(6) LENGTH IN BYTES OF PROC MEMBER NAME
546+         DC    C'IMSMSG' MSG PROC MEMBER NAME
547+SCDPROC4 DS    0F
548+         DC    AL2(IHB0C06-*) MESSAGE LENGTH
549+         DC    B'0000000000000000' MCSFLAGS FIELD
550+         DC    C'S INIT.CLASSB,,,(B,A) ' MESSAGE
551+IHB0006  EQU   *
552+SCDPROC5 DS    0F
553+         DC    AL2(IHB0C07-*) MESSAGE LENGTH
554+         DC    B'0000000000000000' MCSFLAGS FIELD
555+         DC    C'P CLASSB' MESSAGE
556+IHB0007  EQU   *
557+         DS    4F

559                *,*****************************************************
560                *,*              IMS/360 SYSTEM
561                *,*      VERSION 2   RELEASE 2   MOD LEVEL 0       *
562                *,*****************************************************
```

```
 565           ICLI   CLBBASE=C,CNTBASE=0,CTBBASE=0,CTTBASE=0                            00119000
 566+*********************************************************************************
 567+*
 568+*         CLB - COMMUNICATION LINE BLOCKS DSECT. DSECT NAME IS IECTDECB.
 569+*
 570+*********************************************************************************
 571+IECTDECB DSECT  DECB DUMMY SECTION
 572+*         -----------------------------------------
 573+*         +                                       +
 574+*      0  +       STANDARD EVENT CONTROL BLOCK    +
 575+*         +                                       +
 576+*         +---------------------------------------+
 577+*         +                   +                   +
 578+*      4  +  OPERATION TYPE   +    AREA LENGTH    +
 579+*         +                   +                   +
 580+*         +---------------------------------------+
 581+*         + ON-LINE +                             +
 582+*      8  +TERMINAL +        ADDRESS OF DCB       +
 583+*         +  TEST   +                             +
 584+*         +---------------------------------------+
 585+*         +         +                             +
 586+*     12  +RESERVED +        ADDRESS OF AREA      +
 587+*         +         +                             +
 588+*         +---------------------------------------+
 589+*         +                   +                   +
 590+*     16  +    SENSE BYTES    +   RESIDUAL COUNT  +
 591+*         +     1    &    2   +                   +
 592+*         +---------------------------------------+
 593+*         +         +                             +
 594+*     20  + COMMAND |   ADDRESS OF TERMINAL LIST  +
 595+*         +   CODE  +                             +
 596+*         +---------------------------------------+
 597+*         +         + RELATIVE+         +         +
 598+*     24  +  STATUS +   LINE  + ADDRESS + VRC/LRC +
 599+*         +  FLAGS  +  NUMBER + RESPONSE+ RESPONSE +
 600+*         +---------------------------------------+
 601+*         +         +         +                   +
 602+*     28  +  TP-OP  +  ERROR  +     CSW STATUS    +
 603+*         +   CODE  +  STATUS +                   +
 604+*         +---------------------------------------+
 605+*         +         +                             +
 606+*     32  +RESERVED +      ADDRESS OF CURRENT     +
 607+*         +         +       ADDRESSING ENTRY      +
 608+*         +---------------------------------------+
 609+*         +         +                             +
 610+*     36  +RESERVED +      ADDRESS OF CURRENT     +
 611+*         +         +        POLLING ENTRY        +
 612+*         +---------------------------------------+
 613+*         +         +         +                   +
 614+*     40  +RESERVED +RESERVED +  WRITE AREA LENGTH +
 615+*         +         +         +                   +
 616+*         +---------------------------------------+
 617+*         +         +                             +
 618+*     44  +RESERVED +   ADDRESS OF WRITE AREA     +
 619+*         +         +                             +
```



```
 620+*         -----------------------------------------
```

```
622+DECSDECB DS     1F STATUS FLAG + ADDRESS OF THE TCB

624+DECTYPE  DS     1H OPERATION TYPE

626+DECLNGTH DS     1H AREA LENGTH

628+DECONLTT DS     0CL1 RESERVED FOR ON-LINE TERMINAL TEST
629+DECDCBAD DS     1F ADDRESS OF DCB

631+DECAREA  DS     1F ADDRESS OF AREA

633+DECSENS0 DS     1C 1ST SENSE BYTE

635+DECSENS1 DS     1C 2ND SENSE BYTE

637+DECCOUNT DS     1H RESIDUAL COUNT

639+DECCMCOD DS     0CL1 COMMAND CODE
640+DECENTRY DS     1F ADDRESS OF TERMINAL LIST


643+DECFLAGS DS     1C STATUS FLAGS

645+DECRLN   DS     1C RELATIVE LINE NUMBER

647+DECRESPN DS     1H RESPONSE FIELDS

649+DECTPCOD DS     1C TP-OP CODE

651+DECERRST DS     1C ERROR STATUS

653+DECCSWST DS     1H CSW STATUS

655+DECADRPT DS     1F ADDRESS OF CURRENT ADDRESSING ENTRY

657+DECPOLPT DS     1F ADDRESS OF CURRENT POLLING ENTRY

659+         DS     2C RESERVED

661+DECWLNG  DS     1H WRITE AREA LENGTH

663+DECWAREA DS     1F ADDRESS OF WRITE AREA

665+CLBDECB  EQU    DECSDECB
666+CLBRCNTR DS     1C RETRY COUNTER                                       $

668+CLBFLAG1 DS     C .             BIT USAGE
669+CLB1COB  EQU    X'80' .     0 - COB REQUIRED FOR CTB'S ON THIS LINE
670+CLB1TEST EQU    X'40' .     1 - TEST RECYCLE REQUEST
671+CLB1IBUF EQU    X'20' .     2 - INPUT BUFFER ALLOCATED
672+CLB1OBUF EQU    X'10' .     3 - OUTPUT BUFFER ALLOCATED
673+*        EQU    X'08' .     4 - RESERVED FOR FUTURE USE
674+CLB1QCRI EQU    X'04' .     5 - QCR ALLOCATED  (INPUT)             $
675+CLB1RESP EQU    X'02' .     6 - RESPONSE WAITING
676+CLB1QCRO EQU    X'01' .     7 - QCR ALLOCATED   (OUTPUT)           $
```

```
678+CLBFLAG2 DS    C
679+CLB2NOIN EQU   X'80' .    0 - NO INPUT ALLOWED
680+CLB2NOOU EQU   X'40' .    1 - NO OUTPUT ALLOWED
681+CLB2NOQU EQU   X'20' .    2 - DO NOT QUEUE ON THIS LINE
682+*        EQU   X'10' .    3 - RESERVED FOR FUTURE USE
683+CLB2IDLE EQU   X'08' .    4 - IDLE LINE
684+CLB2INP  EQU   X'04' .    5 - INPUT IN CONTROL
685+CLB2OUTP EQU   X'02' .    6 - OUTPUT IN CONTROL
686+CLB2AUTO EQU   X'01' .    7 - AUTOPOLL

688+CLBFLAG3 DS    C .
689+CLB3INP  EQU   X'80' .    0 - 0=OUTPUT NEXT, 1=INPUT NEXT ON THIS LINE
690+CLB3CSUB EQU   X'40' .    1 - COMMAND ENTERED AS NON-FIRST SEGMENT
691+CLB3DOPN EQU   X'20' .    2 - LINE IS OPENED DYNAMICALLY
692+CLB3STOP EQU   X'10' .    3 - LINE HAS BEEN IDLED FOR SHUTDOWN
693+CLB3CBUF EQU   X'08' 4 - CONDENSED BUFFER ALLOCATED
694+CLB3ICMD EQU   X'04' 5 - INTERNAL COMMAND IN PROGRESS
695+*        EQU   X'02' .    6 - RESERVED FOR FUTURE USE
696+*        EQU   X'01' .    7 - RESERVED FOR FUTURE USE

698+CLBCNTPT DS    F . POINTER TO CNT FOR RESPONSE                        $
699+CLBTEMP1 DS    F .
700+CLBTEMP4 DS    F .              USED FOR TEMPORARY STORAGE
701+CLBTEMP5 DS    F .              USED FOR TEMPORARY STORAGE

703+CLBCSID  DS    OC . FIND DEST FIELD                                    $
704+*                       0      DL/1 USE
705+*                       1      RESTART IN PROGRESS
706+CLBCSCVB EQU   X'20' .  2       CVB FOUND AS DESTINATION
707+*                       3      ALWAYS 0 FOR CLB
708+CLBCSFND EQU   X'08' .  4      CNT/SMB NOT FOUND
709+CLBCSSMB EQU   X'04' .  5 = 0   CNT FOUND
710+*                         = 1 SMB FOUND
711+CLBCSDST EQU   X'02' .  6       FNDDST ENTERED (CLBCNTP POINTS TO DEST)
712+CLBCSACT EQU   X'01' .  7      ACTIVE POLLING LIST

714+CLBPOLAD DS    F .  POLLING LIST ADDRESS
715+CLBCTBPT DS    H .  FIRST CTB ON LINE
716+CLBCRCTB DS    H .  CURRENT CTB
717+CLBCNTQB DS    2F . QCB FOR CNTS FOR OUTPUT
718+CLBSMBPT DS    F .  POINTER TO BLOCK FOR QUEUEING CNT/SMB/CVB          $
719+CLBCTBPS DS    H .
720+CLBDPOLC DS    C
721+CLBDCTL  DS    C .              CLB DEVICE CONTROL BYTE (USED BY DDM)
722+CLBINBUF DS    F .         INPUT BUFFER POINTER
723+CLBOUTBF DS    F .         OUTPUT BUFFER POINTER
724+*
725+* IF ($) FALLS IN COLUMN 71 THIS FIELD WILL BE SAVED IN A COB IF REQ
```

```
727+***********************************************************************
728+*
729+*        CTT - COMMUNICATION TRANSLATE TABLES DSECT.
730+*
731+***********************************************************************
732+CTT      DSECT
733+CTTSEND  DS    F .  LOCATION OF OUTPUT TRANSLATE TABLE
734+CTTREC   DS    F .  LOCATION OF INPUT TRANSLATE TABLE
735+CTTDD    DS    F .         POINTER TO DEVICE DEPENDENT ROUTINE
736+CTTEDIT  DS    F .ENTRY FOR USER EDIT OUTPUT ROUTINE
737+CTTINLNG DS    H .         INPUT LINE BUFFER LENGTH
738+CTTBUFSZ DS    H .         MAXIMUM OUTPUT BUFFER SIZE

740+CTTOPT1  DS    C .
741+CTT1MULT EQU   X'80' .   0  THIS DEVICE HAS MULTISEG INPUT BUFFERS
742+CTT1BACK EQU   X'40' .   1  BACKSPACE ELIMINATION NOT REQUIRED
743+CTT1RESP EQU   X'20' .   2  TREAT RESPONSE TYPE AS NORMAL MESSAGE
744+CTT1BSC  EQU   X'10' .   3  BSC
745+CTT1DIAL EQU   X'08' .   4  SWITCHED
746+CTT1EBCD EQU   X'04' .   5  INPUT IS IN EBCDIC
747+CTT1NOTP EQU   X'02' .   6  NON TP DEVICE
748+CTT1SEBC EQU   X'01' .   7  OUTPUT IS IN EBCDIC

750+CTTOPT2  DS    C .
751+CTT2UCAS EQU   X'80' INPUT IS IN UPPER CASE
752+CTT2SYNC EQU   X'40' . 1      DEVICE REQUIRES SYNCRONIZATION
753+CTT2NORP EQU   X'20' RESET POLL NOT APPLICABLE
754+CTT2FRSP EQU   X'10' FORCE RESPONSE MODE
755+CTT2INQ  EQU   X'08' INQUIRY ONLY TERMINAL

757+CTTNTRYL DS    C .         LENGTH OF POLLING LIST ENTRY
758+CTTTABI  DS    C .         # OF IDLES FOR TAB CHAR
759+CTTNLI   DS    C .         # OF IDLES FOR NEW LINE CHAR
760+CTTLFI   DS    C .         # OF IDLES FOR LINE FEED CHAR
761+         DS    H .         RESERVED
```

```
763+*********************************************************************
764+*
765+*        CTB - COMMUNICATION TERMINAL BLOCKS DSECT.
766+*
767+*********************************************************************
768+CTB      DSECT
769+CTBTYPE  DS    C .       CTT NUMBER
770+CTBLINE  DS    C .       LINE NUMBER
771+CTBTERM  DS    3C .      TERMINAL ADDRESS

773+CTBFLAG1 DS    C .
774+CTB1CONV EQU   X'80' .  0 THIS CTB IN CONVERSATION
775+CTB1MAST EQU   X'40' .  1 MASTER TERMINAL
776+CTB1SUBP EQU   X'20' .  2 SUBPOOL CTB
777+CTB1DIAL EQU   X'10' .  3 DIAL CTB IS PHYSICALLY CONNECTED
778+CTB1SIGN EQU   X'08' .  4 DIAL CTB IS LOGICALLY CONNECTED
779+CTB1HELD EQU   X'04' .  5 CONVERSATION IN PROGRESS WAS HELD
780+CTB1PCNT EQU   X'02' .  6 PRESET CNT
781+CTB1PSMB EQU   X'01' .  7 PRESET SMB

783+CTBFLAG2 DS    C .
784+CTB2NOIN EQU   X'80' .  0 NO INPUT
785+CTB2NOOU EQU   X'40' .  1 NO OUTPUT
786+CTB2NOQU EQU   X'20' .  2 NO QUEUEING
787+CTB2LOCK EQU   X'10' .  3 LOCKED
788+CTB2TEST EQU   X'08' .  4 TEST MODE
789+CTB2EXCL EQU   X'04' .  5 EXCLUSIVE MODE
790+CTB2INOP EQU   X'02' .  6 INOPERABLE
791+CTB2EDIT EQU   X'01' .  7 USER OUTPUT EDIT REQUESTED

793+CTBFLAG3 DS    C .
794+CTB3READ EQU   X'80' .  0 INPUT ONLY DEVICE
795+CTB3QSYS EQU   X'40' .  1 QUEUE ALL SYSTEM MESSAGES
796+CTB3LOOP EQU   X'20' .  2 LOOP TEST PENDING
797+CTB3CHE  EQU   X'10' .  3 LINE DISCONNECT REQUESTED
798+CTB3QERR EQU   X'08' .  4 ERROR ON GET NEXT FOR OUTPUT
799+CTB3QMSG EQU   X'04' .  5 QUEUE CAUSE OF ERROR WITH SYSMSG
800+CTB3LAST EQU   X'02' .  6 LAST CTB ON THIS LINE
801+CTB3SEG1 EQU   X'01' .  7 FIRST SEGMENT

803+CTBFLAG4 DS    C .
804+CTB4CNCL EQU   X'80' .  0  DEQUEUE MESSAGE IN PROCESS REQUESTED
805+CTB4OUTP EQU   X'40' .   OUTPUT ONLY TERMINAL
806+*        EQU   X'20' .  2  NOT USED
807+*        EQU   X'10' .  3  NOT USED
808+*        EQU   X'08' .  4  NOT USED
809+*        EQU   X'04' .  5  NOT USED
810+*        EQU   X'02' .  6  NOT USED
811+*        EQU   X'01' .  7  NOT USED

813+CTBACTL  DS    C .       FLAGS TO BE USED BY ANALYZER
814+CTBAINP  EQU   X'80' .  0  0=OUTPUT NEXT, 1= INPUT NEXT
815+CTBAEOM  EQU   X'40' .  1  1=END OF OUTPUT MSG
816+CTBASHUT EQU   X'20' .  2  1=SYSTEM SHUTDOWN
817+CTBAINC  EQU   X'10' .  3  1=INCORE SYSTEM MESSAGE
```

```
 818+CTBAMULT EQU    X'08' .  4    REJECT IF NOT MULTIPLE SEGMENT MESSAGE
 819+CTBAERR  EQU    X'04' .  5    ERROR FOUND ON LAST SEGMENT PASSED

 821+CTBDCTL  DS     C .         FLAGS USED BY DD ROUTINES
 822+CTBDCTL2 DS     C .         FLAGS USED BY DD ROUTINES

 824+CTBFEAT  DS     C .         FEATURES FLAGS
 825+CTBFSYN1 EQU    X'08' ASYNCHRONOUS DEVICE,  SYNC FLAG 1
 826+CTBFSYN2 EQU    X'04' ASYNCHRONOUS DEVICE,  SYNC FLAG 2
 827+CTBFPAGE EQU    X'02' PACING IN PROCESS
 828+CTBFNAPD EQU    X'01' AUTOMATIC PAGE DELETION NOT REQUESTED

 830+CTBCOMP  DS     C .         COMPONENTS TWO BITS FOR EACH
 831+CTBC1NA  EQU    X'80' COMPONENT 1 IS NOT ATTACHED                3088
 832+CTBC1IP  EQU    X'40' COMPONENT 1 IS INOPERABLE                  3088
 833+CTBC2NA  EQU    X'20' COMPONENT 2 IS NOT ATTACHED                3088
 834+CTBC2IP  EQU    X'10' COMPONENT 2 IS INOPERABLE                  3088
 835+CTBC3NA  EQU    X'08' COMPONENT 3 IS NOT ATTACHED                3088
 836+CTBC3IP  EQU    X'04' COMPONENT 3 IS INOPERABLE                  3088
 837+CTBC4NA  EQU    X'02' COMPONENT 4 IS NOT ATTACHED                3088
 838+CTBC4IP  EQU    X'01' COMPONENT 4 IS INOPERABLE                  3088

 840+CTBINCT  DS     H .         INPUT MSG COUNT
 841+CTBOUTCT DS     H .         OUTPUT MSG COUNT
 842+CTBCPMP  DS     H .         PASSWORD MATRIX POINTER
 843+CTBCNTP  DS     H .         LOGICAL TERMINAL POINTER
 844+CTBPREST DS     H .         PRESET DESTINATION OFFSET
 845+CTBCOB   DS     H .         OFFSET TO COMMUNICATION OVERFLOW BLOCK
 846+CTBCCB   DS     H .         OFFSET TO CONVERSATIONAL CONTROL BLOCK
 847+CTBCXB   DS     H .         COMMUNICATION EXTENT BLOCK POINTER
 848+CTBPGNO  DS     H .         CURRENT PAGE BEING VIEWED
```

```
850+*********************************************************************
851+*
852+*         CNT - COMMUNICATION NAME TABLES DSECT.
853+*
854+*********************************************************************
855+CNT      DSECT
856+CNTQE    DS     2F .      QE FOR CNT Q OFF CLB
857+CNTQCBDQ DS     A DRRN OF NEXT MESSAGE TO READ
858+CNTQCBEQ DS     A DRRN OF LAST MESSAGE WRITTEN

860+CNTQFLG1 DS     C CNT QUEUE FLAG 1
861+*        EQU    X'80'        QUEUE READ IN PROCESS
862+*        EQU    X'40'        QUEUE 1 HAS MESSAGE ENQUEUED
863+*        EQU    X'20'        QUEUE 2 HAS MESSAGE ENQUEUED
864+*        EQU    X'10'        QUEUE 3 HAS MESSAGE ENQUEUED
865+*        EQU    X'08'        QUEUE 4 HAS MESSAGE ENQUEUED
866+*        EQU    X'04'        MESSAGE EXISTS IN BACKUP QUEUE
867+*        EQU    X'02'        QCBDQ POINTS TO QUEUE BLOCK RECORD

869+CNTQFLG2 DS     C CNT QUEUE FLAG 2
870+*        EQU    X'80'        THIS DESTINATION IS PERMANENT
871+*        EQU    X'40'        AVERAGE LENGTH IS AVAILABLE
872+*        EQU    X'20'        ENQ/DEQ COUNT IS AVAILABLE
873+*        EQU    X'10'        NAME FIELD EXISTS
874+*        EQU    X'0F'        DESTINATION TYPE 0 THRU 15

876+CNTQAVGL DS     H AVERAGE MESSAGE LENGTH FOR DESTINATION
877+CNTDQCT  DS     H NUMBER OF MESSAGES DEQUEUED
878+CNTNQCT  DS     H NUMBER OF MESSAGES ENQUEUED
879+CNTNAME  DC     D'0' LOGICAL TERMINAL NAME

881+CNTFLAG1 DS     C CNT FLAG 1
882+CNT1EDIT EQU    X'80' USER EDIT ROUTINE
883+CNT1MAST EQU    X'40' MASTER TERMINAL CNT
884+CNT1NOQU EQU    X'20' DO NOT QUEUE ON THIS CNT
885+CNT1NOSE EQU    X'10' DO NOT SEND TO THIS CNT
886+CNT1LOCK EQU    X'08' CNT IS LOCKED
887+CNT1SIGN EQU    X'04' SIGN ON SPECIFIED LTERM ONLY
888+CNT1QERR EQU    X'02' I/O ERROR OCCURED ON QUEUE
889+CNT1CNT  EQU    X'01' ALWAYS ZERO, INDICATES CNT

891+CNTFLAG2 DS     C CNT FLAG 2
892+CNT2UPP  EQU    X'80' UPPER CASE TRANSLATION REQUESTED
893+CNT2PAGE EQU    X'40' PAGED MESSAGE IN PROCESS
894+CNT2AERR EQU    X'20' SYSTEM ERROR MSG PLACED IN Q BY DFSCLMR0
895+*        EQU    X'10'        RESERVED FOR FUTURE USE
896+*        EQU    X'08'        RESERVED FOR FUTURE USE
897+*        EQU    X'04'        RESERVED FOR FUTURE USE
898+CNT2SUB  EQU    X'02' SUB POOL CNT
899+CNT2INQ  EQU    X'01' INQUIRY ONLY CNT

901+CNTCTBP  DS     H CTB POINTER
902+CNTCNTP  DS     H POINTER TO NEXT CNT ON SAME CTB
903+CNTCPMP  DS     H PASSWORD MATRIX POINTER
904+CNTCMPNT DS     C COMPONENT POINTER
```

STMT   SOURCE STATEMENT   F15OCT70   3/27/72

```
905+              DS     CL3 RESERVED FOR FUTURE USE
```

STMT   SOURCE STATEMENT   F15OCT70   3/27/72

```
907               END                                                          00120000
```

7770-3 INPUT EDIT ROUTINE - DFSI7770

For the 7770-3, a user input edit exit has been implemented at the line level (from device module DFSDS030). This exit is primarily provided for a user edit routine to operate conversationally with the line (caller) to do basic (no data base reference) validity checking of input fields (the 7770-3 has limited error detection) and the building of a transaction, field by field, until enough data has been received and validity checked that the message (transaction) can be scheduled into the IMS/360 system. Message text has been translated to EBCDIC before the user routine is invoked.

Note: IMS/360 checkpoint/restart and recovery capabilities are not effective until the message has been scheduled into the system (see return codes 0 and 4 below).

In conjunction with the above concept of input editing, several additional entries and actions have been provided for the user input edit routine to allow the user edit to be continually aware of the line status from operation to operation.

INTERFACE - 7770-3 USER INPUT EDIT ROUTINE

REGISTERS

On Entry:

R0	-	Entry Vector Value
R1	-	Address of input data/buffer area
R2	-	Length of input data/buffer area
R7	-	CTB address
R8	-	CTT address
R9	-	CLB address
R10	-	CNT address
R11	-	SCD address
R13	-	Save area address. The first three words in the save area may not be changed.
R14-R15	-	Standard OS linkage registers
Note:		The save area chain fields for the save area pointed to by R13 must not be modified.

Entry Vector Value

00 - Entry is for normal segment read completion from the line (caller).

04 - Reentry for next segment of message after input edit has indicated that it has more segments to send to the analyzer. R2 has count of usable buffer length of buffer pointed to by R1.

08 - The calling party on the line has hung up.

12 - The line is being stopped or the system is shutting down.

On Exit:

All registers must be restored except R0, R2, and R15. The contents of R0 and R1, if any, are ignored by the device-dependent routine.

R2 - Must contain the count of data now in the input buffer area that was pointed to by R1 on entry.

R15 - Contains one of the following return codes:

00 - The message segment in the input buffer is to be sent to the analyzer and is the last segment of the message.

04 - The message segment in the input buffer is to be sent to the analyzer and is not the last segment of the message. The next time the device module is entered for a read, it will enter the edit module with R1 pointing to a buffer area and R2 containing the amount of available area contained in the buffer. R0 will contain the value of 04.

08 - The message in the input buffer is to be sent to the caller followed by a read. R2 must contain the count for the message to be sent to the caller and the message must be in drum address form.

12 - Repeat the last output message for the caller.

16 - The contents of the input buffer should be sent to the caller with a reset to hang the caller up.

ERROR CONDITIONS

The line will be stopped and a message generated to the master terminal for any one of the following input edit module error conditions:

1. The return code from the input edit module exceeds 16.

2. The count value returned in R2 is greater than the available space in the buffer (buffer overrun).

3. The input-edit module sent a single segment message to the analyzer after the caller has hung up and indicated that it had more segments to send to the analyzer.

4. The return code from the routine exceeds 8 after entered for disconnect indication.

SPECIAL CONDITIONS

After the edit module has been entered with the 08 entry vector value indicating that the caller has hung up, the edit routine may use return codes 00 and 04 to continue sending data to the analyzer before the analyzer is notified of the line drop condition. During this mode of processing, return code 00 will indicate the end of input edit control and that the message should be enqueued for processing. Alternatively, a return code of 08 during this mode will cause the message to be cancelled and terminate input edit control for this sequence. If the edit module was entered with input Vector 12, no external action (device module action) may be specified. Returned parameters, if any, are not used, as the entry with entry Vector 12 is an information-only entry. The return code value of 12 or 16 can only be returned after the user routine was entered for a normal read completion.

DATA SPECIAL CHARACTERS

The input data may contain one or more of the following special characters:

X'00'	For Invalid Input Line Codes
X'16'	For 2721 Cancel Key
X'26'	For EOB (on 2721 also '000' key and '#' Key as EOIs)
X'B0'	For 2721 Verify Key
X'B1'	For 2721 Repeat Key
X'B2'	For 2721 Function 1 (F1) Key
X'B3'	For 2721 Function 2 (F2) Key
X'B4'	For 2721 Function 3 (F3) Key
X'B5'	For 2721 Function 4 (F4) Key
X'B6'	For 2721 Function 5 (F5) Key
X'B7'	For 2721 ID X'19' Code
X'B8'	For 2721 ID X'59' Code
X'B9'	For 2721 ID X'21' Code
X'BA'	For 2721 ID X'61' Code
X'FA'	For 2721 00 Key and for TOUCH-TONE+ (or equivalent) Phone '*' Key when working on the ABB' Code Line Interface

+ Registered Trademark of American Telephone & Telegraph Co.

7770-3 USER INPUT EDIT ROUTINE INCLUSION DURING SYSTEM DEFINITION

IMS/360 supplies a basic input edit routine for the 7770-3 as module DFSI7770 in IMS2.LOAD. If the user wishes to use the supplied module, it is his responsibility to move the supplied module from IMS2.LOAD to the user library specified in the IMSGEN statement. If the user has written his own input edit routine, that module must be placed into the user library specified in the IMSGEN statement prior to system definition. The module must be named and have an entry point with the name DFSI7770.

7770-3 SUPPLIED INPUT EDIT ROUTINE PROGRAM LISTING

For more information on the IMS/360-supplied input edit routine, see the Description of Module DFSI7770 in the *IMS/360 System Manual*, Volume I, Form LY20-0629. The source listing of the IMS/360-supplied module follows for reference.

```
STMT    SOURCE STATEMENT                                                  F15OCT70    3/27/72

  2 DFSI777C CSECT                                                                 00002000
  3 ************************************************************************ 00003000
  4 *                                                                      * 00004000
  5 *     7770 USER INPUT EDIT MODULE SUPPLIED BY IMS                      * 00005000
  6 *                                                                      * 00006000
  7 *    . THIS MODULE ASSUMES NO RESPONSIBILITY FOR TRANSMISSION ERROR    * 00007000
  8 *      DETECTION OR CORRECTION.                                        * 00008000
  9 *                                                                      * 00009000
 10 *    . A MESSAGE IS ASSUMED COMPLETE AND NO ATTEMPT WILL BE MADE TO    * 00010000
 11 *      SEGMENTIZE INPUT DATA                                           * 00011000
 12 *                                                                      * 00012000
 13 *    . THE FIRST TWO CHARACTERS OF THE DATA IS ASSUMED TO CONTAIN A    * 00013000
 14 *      DEFINED TRANSACTION CODE OR LOGICAL TERMINAL NAME               * 00014000
 15 *                                                                      * 00015000
 16 *    . INPUT PASSED BY THIS MODULE WILL BE 1 BYTE LONGER THAN THE DATA * 00016000
 17 *      INPUT FROM THE TERMINAL WITH A BLANK INSERTED AFTER THE SECOND  * 00017000
 18 *      CHARACTER                                                       * 00018000
 19 *                                                                      * 00019000
 20 *    . EOI ONLY INPUT WILL BE SENT TO THE SYSTEM AS A NO TEXT MESSAGE  * 00020000
 21 *                                                                      * 00021000
 22 *    . ANY CHARACTER FOLLOWED BY EOI WILL BE SENT AS A REPEAT REQUEST  * 00022000
 23 *                                                                      * 00023000
 24 *    . AN INPUT OF 99+EOI WILL BE USED AS NORMAL SIGN/OFF; THE EDIT    * 00024000
 25 *      ROUTIN E WILL RETURN TO THE DDM WITH A DISCONNECT REQUEST.      * 00024010
 26 *                                                                      * 00025000
 27 ************************************************************************ 00026000

STMT    SOURCE STATEMENT                                                  F15OCT70    3/27/72

 29            SAVE  (14,12),,I779090                                              00028000
 30+           B     12(0,15) BRANCH AROUND ID
 31+           DC    AL1(7) LENGTH OF IDENTIFIER
 32+           DC    CL7'I779C90' IDENTIFIER
 33+           STM   14,12,12(13) SAVE REGISTERS
 34            USING DFSI7770,R12                                                  00029000
 35            LR    R12,R15          SET BASE REGISTER                            00030000
 36            CH    R0,TWLVE         VALIDITY CHECK ENTRY VECTOR                  00031000
 37            BH    BADVECT          BRANCH IF TOO HIGH                           00032000
 38            LR    R15,R0           COPY THE ENTY VECTOR                         00033000
 39            B     ENTRY(R15)       GO TO PROPER ROUTINE                         00034000
 40 ENTRY      EQU   *                                                             00035000
 41            B     ENTRY1           00 READ COMPLETION FROM LINE                 00036000
 42            B     BADVECT          04 GET NEXT SHOULD NOT OCCUR FOR THIS        00037000
 43            B     ENTRY2           08 LINE DISCONNECT ENTRY                     00038000
 44            B     RETURN           12 NO ACTION ON LINE STOP OR SHUTDOWN        00039000
```

```
STMT    SOURCE STATEMENT                                                        F15OCT70    3/27/72

  46 ENTRY1   EQU   *                                                                  00041000
  47          CH    R2,TWLVE          CHECK NO. DATA CHARS REC'D                       00042000
  48          BNH   SPECIAL                LESS THAN 3 CHAR IS FUNCTION REQUEST        00043000
  49          CH    R2,THIRTEEN            TWO DATA CHAR + EOI ?                       00043010
  50          BNE   MOVER                  BR IF NOT                                   00043020
  51          CLC   10(2,R1),=C'99'        IS IT 99 + EOI ?                            00043030
  52          BE    SIGNOFF                BR IF YES                                   00043040
  53 MOVER    EQU   *                                                                  00043050
  54          MVC   0(2,R1),10(R1)         SET TRANSACTION CODE                        00044000
  55          SH    R2,TWLVE          REMOVE OVERHEAD COUNT                            00045000
  56          EX    R2,MOVTXT              MOVE REMAINDER OF DATA TEXT                 00046000
  57          MVI   2(R1),X'40'            TRANSACTION SEPERATOR                       00047000
  58          AH    R2,THREE               SET DATA LENGTH                             00048000
  59          SR    R15,R15           SCHEDULE SEGMENT WITH EOT R.C.                   00049000
  60          B     RETURN                 RETURN MESSAGE TO ANALYZER                  00050000
  61 MOVTXT   MVC   3(1,R1),12(R1)                                                     00051000
  62 SIGNOFF  EQU   *                                                                  00051010
  63          SR    R2,R2                  NO MESSAGE FOR CALLER                       00051020
  64          LA    R15,16                 SET DISCONNECT REQUEST RC                   00051030
  65          B     RETURN                 AND GO HANG UP THE LINE                     00051040

STMT    SOURCE STATEMENT                                                        F15OCT70    3/27/72

  67 ENTRY2   EQU   *                                                                  00053000
  68          LA    R15,8                  CANCEL ANY MESSAGE IN PROCESS               00054000
  69 RETURN   EQU   *                                                                  00055000
  70          L     R14,12(,R13)      GET RETURN ADDRESS                               00056000
  71          LM    R3,R12,32(R13)    R0,R1 NOT RESTORED. R15,R2 PRESET                00057000
  72          BR    R14               RETURN TO DEVICE MODULE                          00058000
  73 *                                                                                 00059000
  74 SPECIAL  EQU   *   THIS SECTION DEPENDENT ON COMPARE IN ENTRY1 CODE..             00060000
  75          LA    R15,12                      SET REPEAT VECTOR                      00061000
  76          BE    RETURN                      AND DO REPEAT IF 2 CHARS REC'D         00062000
  77          MVI   0(R1),EOT                   SET EOT ONLY FOR NO TEXT MESG          00063000
  78          LA    R2,1                        AND SET DATA COUNT                     00064000
  79          LA    R15,0                       AND SET FOR EOT RETURN                 00065000
  80          B     RETURN                                                             00066000
```

```
 82 BADVECT  EQU    *                                                         00068000
 83          SR     R15,R15        IF BAD INPUT VECTOR SET EOT R.C.           00069000
 84          B      RETURN                     AND TRY TO CONTINUE            00070000
 85 *               CONSTANTS AND DSECTS FOR INPUT EDIT                       00071000


 87 THIRTEEN DC     H'13'                                                     00072010
 88 TWLVE    DC     H'12'                                                     00073000
 89 THREE    DC     H'3'                                                      00074000
 90 EOT      EQU    055                                                       00075000
 91          REQUATE                                                          00076000


 93+*********************************************************************
 94+*                                                                    *
 95+*        EQUATE REGISTERS                                            *
 96+*                                                                    *
 97+*********************************************************************


 99+R0       EQU    0
100+R1       EQU    1
101+R2       EQU    2
102+R3       EQU    3
103+R4       EQU    4
104+R5       EQU    5
105+R6       EQU    6
106+R7       EQU    7
107+R8       EQU    8
108+R9       EQU    9
109+R10      EQU    10
110+R11      EQU    11
111+R12      EQU    12
112+R13      EQU    13
113+R14      EQU    14
114+R15      EQU    15
```



```
116          END                                                              00077000
117                 =C'99'
```

7770-3 OUTPUT EDIT ROUTINE - DFSO7770

The user has the ability to install a 7770-3 with an installation-tailored vocabulary. IMS/360 support cannot, of course, predict this vocabulary. For this reason, an output edit exit was implemented to allow a user-written module to inspect system messages and terminal-to-terminal message switch messages and convert them at the user's discretion to a message that is compatible with his vocabulary.

DATA FORMAT

See Figure 10A in this manual.

INTERFACE - DFSO7770

REGISTERS

On Entry:

R1	-	Address of output message segment
R2	-	Count of output message segment length
R7	-	CTB address
R8	-	CTT address
R9	-	CLB address
R10	-	CNT address
R11	-	SCD address
R13	-	Save area address. The first three words in the save area may not be changed.
R14-R15	-	Standard OS linkage registers
Note:		The save area chain fields for the save area pointed to by R13 must not be modified.

The output edit module will receive control on incore messages, system messages, and message switches. It will not receive control for a message from an application program that is a response to an input transaction.

On Exit:

All registers must be restored except R0, R1, R2, and R15. The contents of R0 and R1, if any, are ignored by the device-dependent routine.

R2 - Must contain the count of data now in the output buffer area that was pointed to by R1 on entry.

R15 - Contains one of the following return codes:

00 - No action taken by the output edit. The device module should continue sending the message and any further segments without routing control to the output edit module.

04 - The current contents of the buffer are to be sent to the line, and the output edit module

desires to gain control for any further segments of this message.

08 - The contents of the buffer have been changed. What is now in the buffer should be sent, and any further segments of the message should be ignored (dequeued and not sent).

ERROR CONDITIONS

The line will be stopped and a message generated to the master terminal for any one of the following output edit module error conditions.

1. The return code from the output edit module exceeds 8.

2. The count returned in R2 is negative or zero.

3. The count returned in R2 is greater than the available buffer space (buffer overrun).

SPECIAL CONDITIONS

The supplied output edit module makes the following assumptions:

1. The vocabulary of the 7770-3 contains the phonetic equivalents for the numbers 0 through 9 and that the translate table supplied by the user will convert the EBCDIC numbers to their vocabulary equivalents.

2. The prefix phrase (in drum address form) to be sent for system messages follows the user translate table and the orientation phrase and has the form nppp, where n is a single byte containing the count of the number of drum address bytes (p) following. The orientation phrase has the format nppp.

3. Because of the variable nature of the 7770-3 vocabulary, the system definition utility requires that the user supply the output translate table for the 7770-3. It is also the user's responsibility to provide the required orientation phrase to be used for system message conversion.

DATA FORMAT

The output message will be edited into the output buffer until the end of message is reached or the buffer is full before control is given to the output edit module. The buffer contains output message data only and is in EBCDIC.

7770-3 OUTPUT EDIT ROUTINE INCLUSION DURING SYSTEM DEFINITION

If the IMS/360-provided output edit routine is to be used, it is the user's responsibility to move the module, DFS07770, from IMS2.LOAD to the user library specified in the IMSGEN statement prior to system definition.

If the user is providing his own output edit routine, the module must be placed into IMS2.RESLIB prior to system definition.

7770-3 OUTPUT EDIT ROUTINE PROGRAM LISTING

For more information on the IMS/360-supplied output edit module, see the Description of Module DFS07770 in the IMS/360 System Manual, Volume I, Form LY20-0629.

The edit routine program listing follows for reference.

```
STMT   SOURCE STATEMENT                                               F15OCT70   3/27/72

   2 DFS07770 CSECT                                                             00002000
   3 ***********************************************************************  00003000
   4 *                                                                     *  00004000
   5 *        7770  SYSTEM MESSAGE EDIT ROUTINE SUPPLIED BY IMS            *  00005000
   6 *                                                                     *  00006000
   7 *    . ANY MESSAGE SWITCHED TO THIS TERMINAL IS SENT AS IS WITH NO    *  00007000
   8 *      MODIFICATION BY THIS PROGRAM                                   *  00008000
   9 *                                                                     *  00009000
  10 *    . SYSTEM 'COMMAND COMPLETED' MESSAGES ARE CONVERTED TO THE USER  *  00010000
  11 *      SUPPLIED ORIENTATION PHRASE                                    *  00011000
  12 *                                                                     *  00012000
  13 *    . SYSTEM ERROR MESSAGES ARE REPLACED BY THE USER SUPPLIED ERROR  *  00013000
  14 *      PHRASE PLUS THE IMS ERROR MESSAGE NUMBER                       *  00014000
  15 *                                                                     *  00015000
  16 ***********************************************************************  00016000
```

```
STMT    SOURCE STATEMENT                                                F15OCT70    3/27/72

  18             SAVE   (14,12),,0779090                                            00018000
  19+            B      12(0,15) BRANCH AROUND ID
  20+            DC     AL1(7) LENGTH OF IDENTIFIER
  21+            DC     CL7'0779090' IDENTIFIER
  22+            STM    14,12,12(13) SAVE REGISTERS
  23             USING  DFS07770,R12                                                00019000
  24             LR     R12,R15                                                     00020000
  25             CH     R2,SEVEN         TOO SHORT FOR SYSTEM USE                   00021000
  26             BL     MSGSW            YES                                        00022000
  27             CLC    1(3,R1),DFS      IS IT A SYSTEM MSG?                        00023000
  28             BNE    MSGSW                                                       00024000
  29             TM     4(R1),X'F0'                                                 00025000
  30             BNO    MSGSW            NO                                         00026000
  31             TM     5(R1),X'F0'                                                 00027000
  32             BNO    MSGSW            NO                                         00028000
  33             TM     6(R1),X'F0'                                                 00029000
  34             BNO    MSGSW            AND NO                                     00030000
  35             USING  CTT,R8                                                      00031000
  36             L      R3,CTTSEND                                                  00032000
  37             LA     R3,256(R3)       GET ACK PHRASE                             00033000
  38             SR     R4,R4                                                       00034000
  39             IC     R4,0(R3)         LENGTH OF PHRASE                           00035000
  40             CLC    4(3,R1),C59      COMMAND COMPLETE PHRASE                    00036000
  41             BH     ERRMSG           NO - ERROR MSG                             00037000
  42             EX     R4,MOVFRAZE                                                 00038000
  43             LR     R2,R4            SET NEW TEXT LENGTH                        00039000
  44             LA     R15,8            SET SKIP REST RETURN CODE                  00040000
  45 RETURN      L      R14,12(13)                                                  00041000
  46             LM     3,12,32(13)                                                 00042000
  47             BR     R14                                                         00043000

STMT    SOURCE STATEMENT                                                F15OCT70    3/27/72

  49 ERRMSG      EQU    *                                                           00045000
  50             LA     R3,1(R3,R4)      POINT TO ERROR PHRASE                      00046000
  51             IC     R4,0(R3)         GET LENGTH                                 00047000
  52             LA     R5,7(R1,R4)      STEP PAST POSSIBLE SELF DESTRUCTION        00048000
  53             MVC    0(3,R5),4(R1)    SAVE ERROR NUMBER OF MESSAGE               00049000
  54             EX     R4,MOVFRAZE      MOVE USER ERROR PHRASE                     00050000
  55             LA     R3,0(R4,R1)                                                 00051000
  56             MVC    0(3,R3),0(R5)    SET ERROR NUMBER                           00052000
  57             LA     R2,3(R4)         SET NEW LENGTH                             00053000
  58             LA     R15,8            SET SKIP REST RETURN CODE                  00054000
  59             B      RETURN                                                      00055000
  60 *                                                                              00056000

STMT    SOURCE STATEMENT                                                F15OCT70    3/27/72

  62 MSGSW       EQU    *                                                           00058000
  63             SR     R15,R15                                                     00059000
  64             B      RETURN                                                      00060000
```

```
  66 *             CONSTANTS  AND DSECTS USED BY DFS07770                                       00062000
  68 SEVEN     DC     H'7'                                                                      00064000
  69 DFS       DC     C'DFS'                                                                    00065000
  70 C59       DC     C'059'                                                                    00066000
  71 MOVFRAZE MVC     0(1,R1),1(R3)                                                             00067000
  72               REQUATE                                                                      00068000
  74+*********************************************************************
  75+*                                                                   *
  76+*          EQUATE REGISTERS                                         *
  77+*                                                                   *
  78+*********************************************************************
  80+R0        EQU    0
  81+R1        EQU    1
  82+R2        EQU    2
  83+R3        EQU    3
  84+R4        EQU    4
  85+R5        EQU    5
  86+R6        EQU    6
  87+R7        EQU    7
  88+R8        EQU    8
  89+R9        EQU    9
  90+R10       EQU    10
  91+R11       EQU    11
  92+R12       EQU    12
  93+R13       EQU    13
  94+R14       EQU    14
  95+R15       EQU    15
  97           ICLI   CTTBASE=C                                                                 00069000
  98+*********************************************************************
  99+*
 100+*          CTT - COMMUNICATION TRANSLATE TABLES DSECT.
 101+*
 102+*********************************************************************
 103+CTT       DSECT
 104+CTTSEND   DS     F .  LOCATION OF OUTPUT TRANSLATE TABLE
 105+CTTREC    DS     F .  LOCATION OF INPUT TRANSLATE TABLE
 106+CTTDD     DS     F .        POINTER TO DEVICE DEPENDENT ROUTINE
 107+CTTEDIT   DS     F .ENTRY FOR USER EDIT OUTPUT ROUTINE
 108+CTTINLNG DS      H .        INPUT LINE BUFFER LENGTH
 109+CTTBUFSZ DS      H .        MAXIMUM OUTPUT BUFFER SIZE
 111+CTTOPT1   DS     C .
 112+CTT1MULT EQU     X'80' .   0  THIS DEVICE HAS MULTISEG INPUT BUFFERS
 113+CTT1BACK EQU     X'40' .   1  BACKSPACE ELIMINATION NOT REQUIRED
 114+CTT1RESP EQU     X'20' .   2  TREAT RESPONSE TYPE AS NORMAL MESSAGE
 115+CTT1BSC  EQU     X'10' .   3  BSC
 116+CTT1DIAL EQU     X'08' .   4  SWITCHED
 117+CTT1EBCD EQU     X'04' .   5  INPUT IS IN EBCDIC
 118+CTT1NOTP EQU     X'02' .   6  NON TP DEVICE
 119+CTT1SEBC EQU     X'01' .   7  OUTPUT IS IN EBCDIC
 121+CTTOPT2   DS     C .
 122+CTT2UCAS EQU     X'80' INPUT IS IN UPPER CASE
 123+CTT2SYNC EQU     X'40' . 1      DEVICE REQUIRES SYNCRONIZATION
 124+CTT2NORP EQU     X'20' RESET POLL NOT APPLICABLE
 125+CTT2FRSP EQU     X'10' FORCE RESPONSE MODE
 126+CTT2INQ  EQU     X'08' INQUIRY ONLY TERMINAL
 128+CTTNTRYL DS      C .        LENGTH OF POLLING LIST ENTRY
 129+CTTTABI   DS     C .        # OF IDLES FOR TAB CHAR
 130+CTTNLI    DS     C .        # OF IDLES FOR NEW LINE CHAR
 131+CTTLFI    DS     C .        # OF IDLES FOR LINE FEED CHAR
 132+          DS     H .        RESERVED
```



```
 134           END                                                                              00070000
```

7770-3 USER OUTPUT TRANSLATE TABLE SAMPLE LISTING

The following is an example of a listing which might be produced for a user-supplied output translate table. See also User Input Edit Routine and User Output Edit Routine in this chapter.

```
                                                                                                PAGE    1

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  F15OCT70

000000                                1 OUT7770  CSECT                                                   00001000
                                      2 ********************************************************************** 00002000
                                      3 *                                                                    * 00003000
                                      4 * 7770 OUTPUT TRANSLATE TABLE                                        * 00004000
                                      5 *                                                                    * 00005000
                                      6 *          THIS TABLE IS DEPENDENT UPON THE VOCABULARY PRESENT       * 00006000
                                      7 *          ON THE 7770 DRUM TRACKS.                                  * 00007000
                                      8 *                                                                    * 00008000
                                      9 ********************************************************************** 00009000

                                     11 *                 0 1 2 3 4 5 6 7 8 9 A B C D E F                      00011000
000000 0001020304050607              12          DC    X'0001C2030405060708090A0B0C0D0E0F' 0 PRE              00012000
000010 1011121314151617              13          DC    X'101112131415161718191A1B1C1D1E1F' 1  FORMATTED       00013000
000020 2021222324252627              14          DC    X'202122232425262728292A2B2C2D2E2F' 2    MESSAGES      00014000
000030 3031323334353637              15          DC    X'303132333435363738393A3B3C3D3E3F' 3                  00015000
000040 0000000000000000              16          DC    X'00000000000000000000000000000000' 4                  00016000
000050 0000000000000000              17          DC    X'00000000000000000000000000000000' 5                  00017000
000060 0000000000000000              18          DC    X'00000000000000000000000000000000' 6                  00018000
000070 0000000000000000              19          DC    X'00000000000000000000000000000000' 7                  00019000
000080 0001020304050607              20          DC    X'00010203040506070809000000000000' 8  LOWER           00020000
000090 0011121314151617              21          DC    X'00111213141516171819000000000000' 9   CASE           00021000
0000A0 0000222324252627              22          DC    X'00002223242526272829000000000000' A    ALPHA         00022000
0000B0 0000000000000000              23          DC    X'00000000000000000000000000000000' B                  00023000
0000C0 0001020304050607              24          DC    X'00010203040506070809000000000000' C  UPPER           00024000
0000D0 0011121314151617              25          DC    X'00111213141516171819000000000000' D   CASE           00025000
0000E0 0000222324252627              26          DC    X'00002223242526272829000000000000' E    ALPHA         00026000
0000F0 1631323334353637              27          DC    X'16313233343536373839000000000000' F  NUMERIC         00027000
                                     28 *                 0 1 2 3 4 5 6 7 8 9 A B C D E F                      00028000

                                     30 ****************************************                           00030000
                                     31 *                                      *                           00031000
                                     32 *   7770-3 IMS/360 ORIENTATION PHRASE  *                           00032000
                                     33 *                                      *                           00033000
                                     34 ****************************************                           00034000

000100 02                            36          DC    AL1(ORIEND-*-1)          PHRASE LENGTH              00036000
000101 2B0E                          37          DC    X'2B0E'                  PHRASE IS 'DIAL RELEASED'  00037000
000103                               38 ORIEND   EQU   *                                                   00038000

                                     40 ****************************************                           00040000
                                     41 *                                      *                           00041000
                                     42 * 7770-3  IMS/360 OUTPUT PREFIX PHRASE *                           00042000
                                     43 *                                      *                           00043000
                                     44 ****************************************                           00044000

000103 06                            46          DC    AL1(OPREND-*-1)          PHRASE LENGTH              00046000
000104 051919161900                  47          DC    X'051919161900'          PHRASE IS 'E R R O R '     00047000
00010A                               48 OPREND   EQU   *                                                   00048000
                                     49          END                                                       00049000
```

7770-3 USER OUTPUT TRANSLATE TABLE

Please refer to the paragraph Special Conditions in the section of this chapter titled "7770-3 Output Edit Routine - DFSO7770" for a description of the requirements for the user output translate table. Refer also to the user output translate table listing that follows in this chapter.

The orientation phrase is used by the device-dependent module. Before and after each read, the phrase is sent to the terminal operator to indicate that a read is pending on the line, and that he can now enter his data.

The prefix phrase is optional. It is used only by the supplied Output Edit Routine - DFSO7770. See the description of module DFSO7770 functions in this chapter.

7770-3 USER OUTPUT TRANSLATE TABLE INCLUSION

Before executing Stage 2 of IMS/360 system definition, the user-supplied translate table must be placed in the user library specified in the IMSGEN statement. The table must be a load module with the name specified in the LINEGRP statement.

IMS/360 DATA BASE BUFFER POOL

The default size of this pool is 7,000 bytes. At the beginning of the data base buffer pool, there exists a work area used by IMS/360 to retain statistics on the activity in the buffer pool. These statistics should be of value to the IMS/360 user in determining the most appropriate buffer pool size for a given application program. The following DSECT describes this work area. There is a fullword aligned address in the IMS/360 System Contents Directory (SCD) control block named SCDDBFPL which points to the following DSECT. For IMS/360 Version 2.3, the SCDDBFPL fullword address is offset from the beginning of the SCD by 276 bytes (decimal). It is suggested that any program which wishes to reference this area use the DSECTS for the SCD and the buffer pool prefix which are in the macro library for the IMS/360 system. The macro for the SCD DSECT is ISCD SCDBASE=0 and the macro for the buffer pool prefix is BFPL. The SCD is the major block within the IMS/360 system. The address of the SCD can be obtained by an application program through a GET SCD call. The format of this call is:

```
CALL    'CBLTDLI'

USING   GSCD, any valid PCB-addr, IOAREA
```

where

GSCD

is a four-character constant for the data base function, and IOAREA is an eight-byte area. The first word of this eight-byte area contains the SCD address upon return from a GET SCD call. The second word contains the address of the PST.

WARNING: When running a MSG or BMP region type, using either the VS2 Operating System or the VS1 Operating System with fetch protect specified, the GSCD call will function normally. The operating system, however, will not permit a program in one region (the MSG or BMP region) to access data in another region (the CTL region), and, therefore, the addresses returned on the GSCD call cannot be used in either a MSG or

BMP region type. An 0C4 System abend will result if they are used in the above situation. Since the SCD and buffer pool are in the same Operating System Region as the application program when running in a DLI or DBB region type, these addresses may be used by a DLI or DBB region.

The buffer pool work area format is as follows:

```
BFPLID     DS  OF
           DC  CL4' ' .     ID OF POOL
BFPLSZE    DC  F'0' .       SIZE OF POOL LESS OHEAD
BFPLFWDT   DC  A(0) .       ADDRESS OF LEAST RECENTLY USED BUFFER
BFPLBWDB   DC  A(0) .       ADDRESS OF LEAST RECENTLY USED BUFFER
BFPLQCB    DC  2F'0' .      QCB FOR USERS WAITNIG FOR OTHER TRMNATIONS
BFPLEQP    DC  XL20'0' .    WORK AREA FOR BUFFER NQ AND DQ MACROES
BFPLRQCT   DC  F'0' .       NUMBER OF BLOCK REQUESTS RECEIVED
BFPLINPL   DC  F'0'         NUMBER OF REQ SATISFIED FROM POOL
BFPLRDCT   DC  F'0'.        NUMBER OF READ REQUESTS ISSUED
BFPLALTR   DC  F'0' .       NUMBER OF BUFFER ALTERATIONS RECEIVED
BFPLOSWT   DC  F'0' .       NUMBER OF OSAM WRITES ISSUED
BFPLBKWT   DC  F'0' .       NUMBER OF BLOCKS WRITTEN
BFPLNWBK   DC  F'0' .       NEW BLOCKS CREATED IN POOL
BFPLCHWT   DC  F'0' .       NUMBER OF CHAINED WRITES ISSUED
BFPLCHBK   DC  F'0' .       NUMBER OF BLOCKS WRITTEN IN WRITE CHAINS
BFPLTLCP   DC  F'0' .       NUMBER OF POOL COMPACTIONS PERFORMED
BFPLBCMB   DC  F'0' .       NUMBER OF BUFFS COMBINED
BFPLBMVD   DC  F'0' .       NUMBER OF BUFFERS MOVED
BFPLISTL   DC  F'0'         NO OF RETRIEVE BY KEY CALLS REC
BFPLIGET   DC  F'0'         NO OF GN CALLS REC
BFPLGTBF   DC  F'0'         NO OF BISAM READS OR QISAM SETLS
BFPLWERR   DC  X'0' .       NUMBER OF PRM WRT ERR BUFFS NOW IN POOL
BFPLWERT   DC  X'0'.        LARGEST NUMBER OF PRM ERR BUFFS EVER IN PL
BFPLSW1    DC  X'0'.        SWITCHES
BFPLCPCT   DC  X'0' .       COUNT OF CURRENT NUMBER OF COMPACTORS
BFPLFBUF   DS  0D .         LOCATION OF FIRST BUFF IN POOL
```

The most important entries in the DSECT above are those starting at BFPLEQP and down through BFPLGTBF.

BFPLRQCT

equals the number of requests for logical records, segments, or physical blocks made to the data base buffer handler by higher level Data Language/I modules

BFPLINPL

equals the number of requests defined under BFPLRQCT which are satisfied from data already in the data base buffer pool without a physical I/O operation required. This number should be a fraction of BFPLRQCT. This value may be increased by increasing the data base buffer pool size.

BFPLRDCT

equals the number of physical I/O read requests performed by the buffer handler. This number should be a fraction of BFPLRQCT. This value will normally be increased by a reduction in the data base buffer pool size.

BFPLALTR

equals the number of data base buffers altered because of deletes, replaces, or inserts to data bases

BFPLOSWT

equals the number of physical I/O writes performed on an OSAM data set or on an ISAM data set using OSAM to write back data. This number should be a fraction of BFPLRQCT.

BFPLBKWT

equals the number of physical I/O write operations performed

BFPLNWBK

equals the number of new data base blocks created in the data base pool which are subsequently written to data base storage. This value is a counter of new physical blocks in all data bases used.

BFPLCHWT

is the number of OSAM write operations performed where multiple blocks are written with one channel program.

BFPLCHBK

is the number of blocks written in the mode described by the parameter BFPLCHWT

BFPLTLCP

is the number of times the data base pool needed to be compacted because of fragmented use of space. This value should be small relative to BFPLRQCT.

BFPLBCMB

is the number of fragmented buffer pool areas combined to obtain space in the buffer pool for new data from the data base

BFPLBMVD

is the number of buffers containing data which must be moved around in a fragmented pool. Both BFPLBCMB and BFPLBMVD should be small values. The ability to ensure these values are small is achieved in two ways:

1) Common block sizes for all data bases used

2) Large data base buffer pool

BFPLISTL

is the number of retrievals by key field value made by DL/I modules to the buffer handler.

BFPLIGET

is the number of QISAM GET NEXT or BISAM sequential READ calls received

BFPLGTBF

is the number of BISAM READs or QISAM SETL operations issued to satisfy retrieval requests. This number should be smaller than BFPLTSTL.

BFPLWERR

is the number of permanent WRITE errors currently in buffer pool

BFPLWERT

is the largest number of permanent WRITE errors ever encountered in the buffer pool during the IMS/360 control or batch region execution.

Just prior to the termination of an application program, it may be beneficial to access the contents of the data base pool work area. Analysis of the information should assist the user in determining the size of the buffer pool for subsequent executions.

HDAM RANDOMIZING MODULES

The Data Language/I access method called HDAM requires the IMS/360 user to supply a module for root segment placement in, or retrieval from, an HDAM data base. One or more modules, called randomizing modules, may be utilized within the IMS/360 system. A particular data base has only one randomizing module associated with it. A generalized module, which uses DBD generation-supplied parameters to perform randomizing for a particular data base, may be written to service several data bases.

The purpose of a randomizing module is to convert a segment search argument key field value supplied by an application program for root segment placement in, or retrieval from, an HDAM data base into a relative block number and anchor point number.

After a randomizing module has been compiled and tested, and before its use by the IMS/360 system, it must be placed into the IMS2.RESLIB data set. Each randomizing module must have a unique name and the name must not conflict with the already existent members of the IMS2.RESLIB data set. Alternative locations for randomizing module storage are SYS1.LINKLIB or any operating system partitioned data set to which access is provided with a JOBLIB or STEPLIB job control language statement.

The name given to the load module used for randomizing functions with a specific data base should also appear in the DBD generation associated with the data base. The load module name must be the value of the "mod" parameter of the RMNAME= operand on the DBD statement in the HDAM DBD generation.

The necessary randomizing module associated with a specific data base is brought into core storage in either the IMS/360 online control program region or batch processing region at the time the associated data base is opened. If a single randomizing module is utilized for more than one HDAM data base, it must be written, compiled, and link edited as reentrant (RENT). It may also be placed in LINKPACK. This will allow one copy of the module to service several data bases concurrently open.

When an HDAM data base is to be utilized in either the IMS/360 online control region or a Data Language/I batch processing region and the randomizing module does not exist in OS LINKPACK, space must be provided for it. Space must be provided in the IMS/360 control region to accommodate all randomizing modules which may be employed for online HDAM data bases.

All randomizing modules are loaded from their resident library by the IMS/360 open module, DFSDLOC0. The IMS/360 open module obtains the name of the randomizing module from the control block entitled RDMVTAB. This block is constructed by the utility block builder program and placed in IMS2.ACBLIB from parameters specified in the associated DBD. If the IMS2.ACBLIB data set is not being used, the block is constructed in core and passed to the IMS/360 open module. The OS LOAD macro instruction is employed.

When an application program issues a Get Unique, Get Next with qualification, or Insert call which operates on a root segment of an HDAM data base, the user-supplied randomizing module is invoked. The segment search argument and the segment I/O work area in the data base call which relates to the sequence field of a root segment provide the primary input parameter to the randomizing module. The following illustrates the format of a segment search argument.

ROOT SEGMENT NAME (SEQUENCE FIELD NAME-OPERATOR-value)

The root segment and sequence field names are eight-character alphameric values. The operator is a two-character arithmetic value. A description is provided in the _IMS/360 Application Programming Reference Manual_ (SH20-0912). Other operators at the root level will give unpredictable results. The value parameter is a term whose length equals the length of a root segment sequence field in the data base and whose content defines an already existent root segment to be retrieved. If the data base call consists of a root segment insert, the segment search argument consists only of the segment name. In this case, the field value is obtained from the segment I/O area provided in the insert call.

This field value parameter is supplied to the randomizing module for conversion to a relative block number and anchor pointer number within the data base. In addition to the field value parameter supplied by an application program, parameters from the DBD generation associated with the data base being used are available to the randomizing module.

When a randomizing module is invoked for the purposes of conversion, control is passed from the IMS/360 data base logical,retrieve function module, DFSDLR00.

The parameters from DBD generation are available to a randomizing module in a CSECT named RDMVTAB. The address of this CSECT is passed to the module each time a conversion is requested.

The following DSECT defines the format of this CSECT:

```
DMBDACS    DSECT
DMBDANME    DS  CL8     NAME OF ADDR ALGORITHM LOAD MODULE
DMBDAKL     DS  OCL1    EXECUTABLE KEY LENGTH OF ROOT
DMBDAEP     DS  A       EP OF ADDR LOAD MODULE
DMBDASZE    DS  H       SIZE OF THIS CSECT
DMBDARAP    DS  H       NUMBER OF ROOT ANCHOR POINTS/BLOCK
DMBDABLK    DS  F       NUMBER OF HIGHEST BLOCK DIRECTLY ADDRSD
DMBDABYM    DS  F       MAX NUMBER OF BYTES BEFORE OFLOW TO 2NDARY
DMBDABYC    DS  F       CUR NUM OF BYTES INSERTED UNDER ROOT
DMBDACP     DS  F       RESULT OF LAST ADDRESS CONVERSION
```

RANDOMIZING MODULE INTERFACES

Upon entry to any randomizing module, registers must be saved. Upon return to IMS/360, registers must be restored. A save area address is provided in Register 13 upon entry for the purpose of register save.

The following registers upon entry to a randomizing module have the indicated meanings:

Register	Meaning or Content
0	Data Management Block address (DMB)
1	DMBDACS CSECT address
7	Partition Specification Table address (PST)
9	Address of first byte of key field value supplied by an application program
13	Save area address. The first three words in the save area must not be changed.
14	Return to IMS/360 address
15	Entry point address of randomizing module

Note 1: If an HDAM data base does not have a sequence field defined, the values supplied to the randomizing module are as follows:

a. The executable key length field in the CSECT named RDMVTAB is not initialized and should not be used.

b. The value in Register 9 at entry to the randomizing module contains the address of the first byte of the user I/O area.

Note 2: If an HDAM data base does not have a sequence field defined at the root level, the randomizing module will only be given control on an insert call. All retrieval-type calls will result in a scan mode operation to satisfy the root level qualification. On GU type calls, the scan is begun at the beginning of the data base. On GN type calls, the scan is begun at the current root level position within the data base.

Internal IMS/360 control blocks which are of value to a randomizing routine are: the Partition Specification Table (PST), the Data Management Block (DMB), Physical Segment Description Block (PSDB) for the root segment, and the first Field Description Block (FDB), which is the root segment key field format description. DSECTs of these blocks are provided in the examples which follow later in this section.

The result of a randomizing module conversion must be in the form

BBBR

where:

R

is a one-byte binary number of the appropriate anchor point within a relative block within an OSAM data set of the data base.

BBB

is a three-byte binary number of the block into which a root segment is to be inserted or from which it is to be retrieved.

This result must be placed in the CSECT addressed by Register 1 in the four-byte fixed name DMBDACP. If the result exceeds the content of the field DMBDABLK, the result is changed to the highest block and last anchor point of that block.

HDAM RANDOMIZING MODULE EXAMPLES

Three randomizing module examples are provided as guidance to the IMS/360 system user. Each of the examples uses one of the following techniques:

1. Modulo or division method
2. Binary halving method
3. Hashing method

The intent of a randomizing module is to convert a root segment key field value to a relative block number and anchor point number in an HDAM data base. The relative block number may range from 1 to $2^{24}-1$. The anchor point number may range from 1 to 255.

MODULO OR DIVISION METHOD EXAMPLE

This module uses the principle that the remainder of a divide can only range from zero to the divisor minus one. Thus, any number divided by four can only yield a remainder of 0, 1, 2, or 3. To determine the base location for a root segment, multiply the number of blocks in the root segment addressable area by the number of anchor points per block. This is effectively the number of base locations for root segments in the root segment addressable area. Then, divide the root segment key field value by the result of the multiplication. The remainder indicates the appropriate base location.

To convert the base location to relative block and anchor point numbers, divide the base location by the number of anchor points per block. This last division leaves the relative block number as the quotient and the anchor point number as the remainder. Since both numbers are relative to zero, both must be incremented by one to yield the correct block and anchor point.

Example:

Assume a) root segment addressable area is 50 blocks

b) 2 anchor points per block

c) root segment key value is 23

Result a) number of base locations = 50 x 2 = 100

b) appropriate base location = 23/100 = 23 remainder

c) appropriate block = 23/2 = 11 (the quotient), appropriate anchor point = 1 (the remainder)

d) adjust both numbers by one; thus, relative block = 12 and anchor point = 2

Notice that external keys 123, 223, 323, etc. will be synonyms. As the number of base locations is increased, the distance between root segments increases. This may waste direct access space. However, the number of synonyms decreases as the number of base locations approaches or exceeds the largest key value. If the root segment key field value is numeric and the number of base locations equals or exceeds largest key value, no synonyms will be produced.

```
HDCNVRT1 CSECT                                                          00001000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00004000
*                                                                     * 00005000
*                S A M P L E   C O N V E R S I O N   P R O G R A M   * 00006000
*                                                                     * 00007000
*              THIS CSECT CONVERTS AN EBCDIC NUMERIC KEY TO A RELATIVE* 00008000
*         BLOCK AND ROOT ANCHOR POINT. THIS RESULT IS OBTAINED AS     * 00009000
*         FOLLOWS  RECNO= MOD(KEY,DMBDABLK*DMBDARAP)                  * 00010000
*                  BLOCK= RECNO/DMBDARAP+1                            * 00011000
*                  RAP   = MOD(RECNO,DMBDARAP)+1                      * 00012000
*              THE CSECT ASSUMES THAT THE EXTERNAL KEY IS 10 BYTES OR * 00013000
*              THE CSECT ASSUMES THAT THE EXTERNAL KEY IS 15 BYTES OR * 00014000
*         LESS.  NON-NUMERIC CHARACTERS ARE VALID, HOWEVER ONLY THE   * 00015000
*         FOUR LOW ORDER BITS WILL BE USED.                           * 00016000
*                                                                     * 00017000
*         CALLING SEQUENCE                                            * 00018000
*              R0 - DMB                                               * 00019000
*              R1 - DMBDACS                                           * 00020000
*              R7 - PST                                               * 00021000
*              R9 - KEY ADDRESS                                       * 00022000
*         ON RETURN                                                   * 00023000
*              DMBDACP - BBBR                                         * 00024000
*                                                                     * 00025000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00026000
         STM   14,12,12(13)        SAVE                                 00027000
         USING PST,R7                                                   00028000
         USING DMBDACS,R1                                               00029000
         USING HDCNVRT1,R15                                             00030000
         XC    PSTDECB(8),PSTDECB INIT FOR CVB                          00031000
         IC    R5,DMBDAKL          GET EXECUTABLE KEY FLD LENGTH        00032000
         EX    R5,PACK                                                  00033000
         SR    R4,R4                                                    00034000
         OI    PSTDECB+7,X'0F'     FORCE SIGN                           00035000
         CVB   R5,PSTDECB                                               00036000
         L     R6,DMBDABLK         HIGEST BLOCK NUMBER DIRECTLY ADDR    00037000
         MH    R6,DMBDARAP         HIGHEST RECORD NUMBER                00038000
         DR    R4,R6                                                    00039000
         LR    R5,R4               RECNUM                               00040000
         SR    R4,R4                                                    00041000
         LH    R6,DMBDARAP                                              00042000
         DR    R4,R6                                                    00043000
         LA    R4,1(,R4)           ROOT ANCHOR POINT                    00044000
         LA    R5,1(,R5)           BLOCK                                00045000
         SLL   R5,8                                                     00046000
         OR    R4,R5               BBBR                                 00047000
         ST    R4,DMBDACP          RESULT                               00048000
         LM    14,12,12(13)        RESTORE                              00049000
         BR    R14                 RETURN                               00050000
PACK     PACK  PSTDECB(8),0(0,R9)                                       00051000
         REQUATE                                                        00052000

+***********************************************************************
+*                                                                     *
```

```
   55+*         EQUATE REGISTERS                                                          *
   56+*                                                                                   *
   57+*************************************************************************************

   59+R0        EQU   0
   60+R1        EQU   1
   61+R2        EQU   2
   62+R3        EQU   3
   63+R4        EQU   4
   64+R5        EQU   5
   65+R6        EQU   6
   66+R7        EQU   7
   67+R8        EQU   8
   68+R9        EQU   9
   69+R10       EQU   10
   70+R11       EQU   11
   71+R12       EQU   12
   72+R13       EQU   13
   73+R14       EQU   14
   74+R15       EQU   15
```

```
 76           IDLI   PSTBASE=C,DMBBASE=0                                          00053000
 77+*                                  * * *    I D L I     18DEC69     * * *
 78+* CHANGES TO IDLI SINCE LAST UPDATED

 80+*...P A R T I T I O N   S P E C I F I C A T I O N   T A B L E :

 82+PST        DSECT
 83+PSTDECB  DS     8F DECB OF THIS PST ENTRY
 84+PSTDOPE  DS     12F DOPE VECTORS FOR I/O TERMINAL PCB
 85+PSTSYMBC DS     CL8 TERMINAL SYMBOLIC
 86+PSTCODE  DS     CL1 CODE
 87+PSTLOSEO EQU    X'80' LOSE ALL OUTPUT MESSAGES (RESTART)
 88+*        EQU    X'40' RESERVED
 89+PSTISPCB EQU    X'20' THIS IS A PCB
 90+PSTPCBIN EQU    X'10' PCB IS IN PST
 91+PSTSMBDE EQU    X'04' DESTINATION IS SMB (0 IF CNT)
 92+*        EQU    X'02' RESERVED
 93+*        EQU    X'01' USED IN ALTERNATE TERMINAL PCB ONLY
 94+PSTRFFU  DS     CL1 CODES
 95+PSTRGU   EQU    X'80' THIS IS A GU CALL, NOT A GN CALL
 96+PSTSPAR  EQU    X'40' SCRATCH PAD HAS BEEN INSERTED
 97+PSTSPAA  EQU    X'20' SCRATCH PAD HAS BEEN INSERTED TO ALTERNATE PCB
 98+PSTRESP  EQU    X'10' RESPONSE HAS BEEN SENT TO ALTERNATE PCB
 99+PSTSPAP  EQU    X'08' 29C8
100+PSTSTAT  DS     CL2 STATUS CODES
101+PSTPRE1  DS     CL4 PREFIX DATA TO BE SCANNED BY USER
102+PSTPRE2  DS     CL4 PREFIX DATA TO BE SCANNED BY USER
103+PSTPRE3  DS     CL4 PREFIX DATA TO BE SCANNED BY USER
104+PSTOQPCB DS     0F OUTPUT PORTION OF I/O PCB TO INTERFACE W/QMGR
105+PSTOPQCR DS     F DRRN OF O/P MSG IF ISRT DONE
106+         DS     2F QMGR CURRENT BFFR & USER CHAIN
107+PSTCNT   DS     A O/P DESTINATION (CNT)
108+         DS     2H QMGR OFFST & MSGLNG
109+PSTIQPCB DS     0F INPUT PORTION OF I/O PCB TO INTERFACE W/QMGR
110+PSTIPQCR DS     F DRRN OF I/P MSG IF SUCCESSFUL GU MADE
111+         DS     2F QMGR CURRENT BFFR & USER CHAIN
112+PSTSMB   DS     F SMB ADDR
113+         DS     2H QMGR OFFST & MSGLNG
114+PSTPREF  DS     F SAVED INPUT PREFIX TERMTYP-COMPADD
115+PSTPREFO DS     H PREFIX OUTPUT USE COUNT (QMGR UPDATES)
116+PSTPREFI DS     H PREFIX INPUT USE COUNT (QMGR UPDATES)
117+PSTIQE   DS     0CL20 SPS IQE IS 5 WORDS
118+PSTIQLNK DS     A SPS IQE LINK FIELD
119+PSTIQPRM DS     A SPS IQE PARM FIELD
120+PSTIQIRB DS     A SPS IQE INTERRUPT RQ BLOCK ADDR
121+PSTPROT  DS     0CL1 PARTITION STORE PROTECT KEY
122+PSTIQTCB DS     A SPS IQE PARTITION TCB ADDR
123+PSTIQECB DS     A SPS IQE PARTITION ECB ADDR
124+PSTSEG   DS     A SEGMENT ADDR
125+PSTSEGL  DS     F SEGMENT LENGTH
126+PSTMI    DS     0C
127+PSTDVIND EQU    X'80' CURRENT PCB IS TERMINAL PCB
128+PSTLAST  EQU    X'40' LAST OF MULTI-BUFF SEGMENT
129+PSTMULTI EQU    X'20' .        MULTI BUFFER SEGMENT
130+PSTUSER  DS     A ADDR OF USERS I/O AREA
```

```
131+PSTPSB    DS    A DIRECTCRY ENTRY OF PSB SCHEDULED
132+PSTCLASS DS    CL4 MESSAGE CLASS SCHEDULING USE
133+PSTQE     DS    2F QE OF PST ENTRY
134+PSTCOUNT DS    F COUNT CF I/P MSGS SUCCESSFULLY DEQUEUED
135+PSTNJOB   DC    CL8' ' CURRENT JOB NAME
136+PSTNSTEP DC    CL8' ' CURRENT STEP NAME
137+PSTACCT   DS    OF BEGINNING OF ACCOUNTING DATA AREA
138+PSTDGU    DC    A(0) GU    DATA BASE CALLS ISSUED
139+PSTDGN    DC    A(0) GN    DATA BASE CALLS ISSUED
140+PSTDGNP   DC    A(0) GNP   DATA BASE CALLS ISSUED
141+PSTDGHU   DC    A(0) GHU   DATA BASE CALLS ISSUED
142+PSTDGHN   DC    A(0) GHN   DATA BASE CALLS ISSUED
143+PSTDGHNP DC    A(0) GHNP DATA BASE CALLS ISSUED
144+PSTDISRT DC    A(0) ISRT DATA BASE CALLS ISSUED
145+PSTDDLET DC    A(0) DLET DATA BASE CALLS ISSUED
146+PSTDREPL DC    A(0) REPL DATA BASE CALLS ISSUED
147+PSTMMOVE DS    A(0) MOVE CALLS ISSUED FROM MSG REGION
148+PSTMGU    DC    A(0) GU    MESSAGE ICAL
149+PSTMGN    DC    A(0) GN    MESSAGE ICAL
150+PSTMISRT DC    A(0) ISRT MESSAGE ICAL
151+PSTMDLET DC    A(0) DLET MESSAGE ICAL
152+PSTACCTL EQU   *-PSTACCT LENGTH OF ACCOUNTING AREA
153+PSTTERM   DC    AL1(0) SCHEDULER FLAG BYTE                          2728
154+PSTMSG    EQU   128 PST REPRESENTS A MESSAGE REGION         2728
155+PSTBMP    EQU   64 PST REPRESENTS BATCH MESSAGE REGION   2728
156+PSTCAN    EQU   32 THIS REGION TO BE STOPPED                2728
157+PSTINT    EQU   16 PST INTERCEPTED FLAG                   CS
158+PSTREP    EQU   8 A REPLY IS OUTSTANDING                  CS
159+PSTBTCH   EQU   4 EXTERNALLY SCHEDUALED PST
160+PSTMVR    EQU   1 REGION CANCELLED - MOVER WAITING      CS
161+PSTPRTY   DS    X INITIATOR CUTOFF PRIORITY               CS
162+          DS    2X RESERVED
163+PSTTPPCB DS    A ADDRESS OF CURRENT OR LAST USED TERMINAL PCB
164+PSTDBPCB DS    A ADDRESS OF CURRENT OR LAST USED DATA BASE PCB
165+PSTTSKID DS    F
166+PSTCODE1 DS    OCL1 BIT MEANINGS IF SET TO 1
167+PSTINTNT EQU   X'40' CANNOT SCHEDULE - INTENT NOT SATISFIED
168+PSTWDMB   EQU   X'20' PST ON DMB WAIT QUEUE
169+PSTSCHED EQU   X'10' OK TO COMPLETE SCHEDULE FUNCTION
170+PSTP3RGN EQU   X'08' THIS IS A TYPE 3 BATCH REGION
171+PSTWPSB   EQU   X'04' PST ON PSB WAIT QUEUE
172+PSTBLKBL EQU   X'02' BLCCK BUILDER CALLED - REPLY NECESSARY
173+PSTBLDDB EQU   X'01' CALL TO BLOCK BUILDER FOR A SPECIAL DBD
174+*               DDIR ADDRESS IN PSTDBPCB
175+PSTSCDAD DS    A ADDRESS OF SCD

177+*               BUFFER HANDLER SECTION OF PST


180+*               BUFFER HANDLER FUNCTION CODES

182+PSTFNCTN DC    XL1'0' CALLERS FUNCTION
183+PSTBKLCT EQU   X'01' LOCATE BLOCK, BLOCK # AT PSTBLKNM, RET=PSTDATA
184+PSTBYLCT EQU   X'02' LOCATE REL BYTE OR REL RECORD
185+*                    REL BYTE OR RECORD AT PSTBYTNM, RET = PSTDATA
```

```
186+PSTISRCH EQU    X'03' CHECK FOR DUPLICATE ISAM BLOCK, ADDR IN PSTBFUSE
187+PSTFCWRT EQU    X'04' WRITE BUFFER AT PSTBFUSE
188+PSTPGUSR EQU    X'05' WRITE ALL BLOCKS MODIFIED BY THIS USER
189+PSTBFALT EQU    X'06' MARK BUFFER ALTERED, ADDR IN PSTBFUSE
190+PSTSRCH  EQU    X'07' SEARCH POOL FOR ISAM BLOCK, KEY ADDRESS AT
191+*                     PSTISAMW, IF FOUND, PSTBYTNM HAS ISAM RLAD,
192+*                     PSTDATA HAS ADDRESS OF ISAM WORK AREA, PSTRTCDE
193+*                     IS 0, IF NOT FOUND, SPACE ADDRESS IS PSTDATA,
194+*                     AND PSTRTCDE IS PSTNOTFD.
195+PSTGBSPC EQU    X'08' GET SPACE IN BUFFER POOL, # OF BYTES IN PSTBYTNM
196+*               PSTDMBNM & PSTDCBNM USED, ID RETURNED IN PSTBLKNM
197+PSTFBSPC EQU    X'09' FREE SPACE IN BUFFER POOL, PSTDMBNM &
198+*               PSTDCBNM & PSTBLKNM HAVE ID OF SPACE
199+PSTDMGNX EQU    X'0A' PUT BLOCK AT BOTTOM OF USE CHAIN & DO PSTGETNX
200+PSTBYALT EQU    X'0B' LOCATE BLOCK AND MARK ALTERED, REL BYTE=PSTBYTNM
201+PSTBFMPT EQU    X'0C' MARK ALL BUFFERS EMPTY. USED FOR A GIVEN DMB
202+*                     OR A GIVEN DCB OR A GIVEN BLOCK.
203+*                     PARAMETERS - PSTDMBNM,PSTDCBNM AND PSTBLKNM.
204+PSTSTLBG EQU    X'F0' RETRIEVE BY KEY    BEGIN AT START OF DATA BASE
205+PSTSTLGT EQU    X'F1' RETRIEVE BY KEY    GT KEY ADDR AT PSTBYTNM
206+PSTSTLEQ EQU    X'F2' RETRIEVE BY KEY    EQ OR GT KEY ADDR AT PSTBYTNM
207+PSTSTLIS EQU    X'F4' RETRIEVE BY KEY    REC TO CHAIN THIS ONE FROM
208+PSTGETNX EQU    X'F8' GET NXT SEQUENTIAL ROOT (KEY)

210+*               SPACE MANAGEMENT FUNCTION CODES

212+PSTGTSPC EQU    X'01' GET SPACE FOR SEGMENT, R5 POINTS TO PSDB
213+PSTFRSPC EQU    X'02' FREE SPACE FOR SEGMENT, R5 POINTS TO PSDB
214+*PSTBTMPFEQU    X'03' TURN OFF BIT FOR PSTBLKNM (ALSO RETURN CODE)
215+PSTGTRAP EQU    X'04' GET SPACE CLOSE TO ROOT ANCHOR POINT IN PSTBYTNM

217+*               OPEN CLOSE FUNCTION CODES
218+PSTOCDSG EQU    X'40' OPEN THE DSG FOUND IN PSTDSGA        V610

220+PSTOCLD  EQU    X'20' OPEN FOR LOAD
221+PSTOCDCB EQU    X'10' DSG ADDRESS IN PSTDSGA - OPEN/CLOSE ONLY THE DCB
222+*                     IN PSTDCBNM
223+PSTOCOPN EQU    X'08' THIS IS AN OPEN CALL
224+PSTOCCLS EQU    X'00' THIS IS A CLOSE CALL (BIT 4=0)
225+PSTOCALL EQU    X'04' CLOSE ALL DMB'S IN THE SYSTEM
226+PSTOCPCB EQU    X'02' CLOSE PCB, ADDRESS OF PCB IN R2
227+PSTOCDMB EQU    X'01' CLOSE DMB, ADDRESS OF DMB IN R2
228+PSTOCBAD EQU    X'80' OPEN NOT SUCCESSFUL

230+*               INDEX MAINTAINANCE FUNCTION CODES                    V911

232+PSTXMDLT EQU    X'A0' PERFORM INDEX MAINTENANCE FOR SEG TO BE DLET
233+PSTXMRPL EQU    X'A1' ''        ''        ''          ''         REPL  V911
234+PSTXMISR EQU    X'A2' ''        ''        ''          ''         ISRT  V911
235+PSTXMUNL EQU    X'A3' ''        ''        ''          ''         UNLD  V911

237+*               BLOCK LOADER FUNCTION CODES

239+PSTDMBRD EQU    1 READ DMB FROM ACBLIB
240+PSTPSBRD EQU    2 READ PSB FROM ACBLIB
```

```
 241+PSTINTRD EQU   3 READ INTENT & DMB NAME LISTS FROM ACBLIB

 243+*              BUFFER HANDLER RETURN CODES

 245+PSTRTCDE DC    1XL1'00' STATUS OF CALL
 246+PSTCLOK  EQU   000 EVERYTHING COPASETIC
 247+PSTGTDS  EQU   004 RBN BEYOND DATA SET
 248+PSTRDERR EQU   008 PERMANENT READ ERROR
 249+PSTNOSPC EQU   012 NO MORE SPACE IN DATA SET
 250+PSTBDCAL EQU   016 ILLEGAL CALL
 251+PSTENDDA EQU   020 END OF DATA SET ENCOUNTERED, NO RECORD RETURNED
 252+PSTNOTFD EQU   024 REQUESTED RECORD CANNOT BE FOUND
 253+PSTNWBLK EQU   028 NEW BLOCK CREATED IN BUFFER POOL
 254+PSTNPLSP EQU   032 INSUFFICIENT SPACE IN POOL
 255+PSTTRMNT EQU   036 USER MUST TERMINATE, NO SPACE IN POOL

 257+*              ISAM SIMULATOR RETURN CODES

 259+PSTKEYHI EQU   012 ON SETL FOR ISRT ISAM KEY WAS HI, RETURN
 260+*                   CODE IN R15 WAS ZERO

 262+*              SPACE MANAGEMENT RETURN CODES

 264+PSTBTMPF EQU   003 TURN OFF BIT FOR PSTBLKNM (ALSO FUNCTION CODE)

 266+PSTOFFST DC    H'0' OFFSET OF SEGMENT OR RECORD FROM PSTDATA
 267+PSTDSGA  DC    F'0' ADDRESS OF DATA SET GROUP PORTION OF JCB
 268+PSTBLKNM DC    F'0' RELATIVE BLOCK NUMBER
 269+PSTDMBNM DC    H'0' DMB NUMBER
 270+PSTDCBNM DC    XL1'00' DCB NUMBER
 271+         DC    XL1'00' RESERVED FOR ALIGNMENT
 272+PSTBYTNM DC    F'0' RELATIVE BYTE ADDR OR RELATIVE RECORD NUMBER
 273+PSTDATA  DC    F'0' CORE ADDRESS OF REQUESTED DATA (RECORD OR SEGMENT)
 274+PSTBUFFA DC    0F'0' CORE ADDR OF BUFFER HEADER
 275+PSTOBSZE DC    F'0' SPACE OBTAINED BY USE OF EXISTING BUFF(S)
 276+PSTRQSZE DC    F'0' SPACE REQUIRED TO HOLD BLOCK + OP CHNL PGM
 277+PSTDSBKQ DC    F'0' DQ PTR FOR REQUESTED BLOCK
 278+PSTBFPDQ DC    F'0' DQ PTR FOR BUFFER POOL
 279+PSTBFUSE DC    A(0) ADDR OF BUFF TO BE USED OR COMPACTED,2NDTM
 280+PSTLFMST DC    A(0) COMPACTED BUFFER WITH LOWEST CORE ADDR
 281+PSTRTMST DC    A(0) COMPACTED BUFFER WITH HGHEST CORE ADDR

 283+*              DATA BASE LOG FUNCTION CODES
 284+*              FUNCTION IN HI BYTE OF PSTWRK1

 286+DBLNDXC  EQU   X'80' INDEX MAINTENANCE CALL
 287+DBLCMC   EQU   X'00' BITS 1-3 = 0 CHAIN MAINTENANCE CALL
 288+DBLNCTR  EQU   X'70' COUNTER MAINTENANCE
 289+DBLLGDLT EQU   X'60' LOGICAL DELETE
 290+DBLPHYI  EQU   X'40' PHYSICAL INSERT
 291+DBLPHYD  EQU   X'20' PHYSICAL DELETE
 292+DBLPHYR  EQU   X'10' PHYSICAL REPLACE
 293+DBLLASTC EQU   X'08' LAST CHANGE FOR THIS USER CALL
 294+DBLFSE1  EQU   X'00' BIT 5 = 0 ONE FSE (IF '40' OR '20' ON)
 295+DBLFSE2  EQU   X'04' TWO FSE'S          (IF '40' OR '20' ON)
```

```
296+DBLPHYRO EQU   X'02' OLD COPY OF A PHYSICAL REPLACE
297+DBLNEWBL EQU   X'01' NEW BLOCK LOG CALL
298+DBLOOPS  EQU   X'0A' NO DATA - END OF USER CALL
299+DBLBUFWT EQU   X'F0' BUFFER WRITE CALL

301+*              PSTWRK1,2,3,4 USAGE BY DB LOG

303+*PSTWRK1       PHYSICAL SDB ADDRESS/COUNT IF NEW BLOCK CALL(LO 2 BYTES)
304+*PSTWRK2,3,4   LOGICAL DELETE - OLD COPY OF PHYSICAL CODE AND DELETE
305+*              BYTE (2 BYTES),  CHAIN MAINTENANCE - OLD COPY OF CHAIN
306+*              POINTER (4 BYTES),  PHYSICAL INSERT AND DELETE - OFFSETS
307+*              AND NEW FSE'S (6 OR 12 BYTES)
308+PSTWRK1  DC    F'0'
309+*                             NEXT CHECKED - COMPACT
310+*                             TRMNATE SIZE - FIND SPACE
311+PSTWRK2  DC    F'0'
312+*                             SPACE COLLECTED SO FAR (HGH 2BYTES - CPS)
313+*                             PST MASK WORK AREA IF WRT ERR - BFFRWRT
314+PSTWRK3  DC    F'0'
315+*                             DO PTR FOR BUFF(S) BEING WRITTEN - BFFRWRT
316+*                             TEMP SAVE OF BLOCK NUMBER -SPC1AA
317+PSTWRK4  DC    F'0'
318+*                             DCB LAST BLOCK WRITTEN BEFORE WRITE -BFWRT
319+*                             TEMP SAVE OF DQ PTR - SPC1AA
320+PSTNUMRD DC    XL1'0' NUMBER OF BLOCKS READ ON THIS CALL
321+PSTNUMCP DC    XL1'0' NUMBER OF POOL COMPACTS ON THIS CALL
322+PSTNUMWT DC    XL1'0' NUMBER OF WRITES ISSUED ON THIS CALL
323+PSTCLRWT DC    X'00' INDICATORS                              V878
324+PSTIWAIT EQU   X'80' IWAIT ISSUED ON THIS CALL               V878
325+PSTLOGQ  DC    A(0) ADDRESS OF REUSE QUEUE QCB IN POOL
326+PSTLOGWA DC    A(0) ADDRESS OF WORK AREA FOR LOG O/P
327+PSTISAMW DC    F'0' WORK AREA USED BY ISAM SIMULATOR
328+PSTWRKT1 DC    F'0' WORK SPACE PRESERVED ACCROSS CALLS TO BUFF HANDLER
329+PSTWRKT2 DC    F'0' WORK SPACE PRESERVED ACCROSS CALLS TO BUFF HANDLER
330+PSTWRKT3 DC    F'0' WORK SPACE PRESERVED ACCROSS CALLS TO BUFF HANDLER
331+PSTWRKT4 DC    F'0' WORK SPACE PRESERVED ACCROSS CALLS TO BUFF HANDLER
332+PSTWRKT5 DC    F'0' WORK SPACE PRESERVED ACCROSS CALLS TO BUFF HANLLER
333+PSTWRKD1 DC    F'0' WORK SPACE FOR USE BY DELETE/REPLACE
334+PSTWRKD2 DC    F'0' WORK SPACE FOR USE BY DELETE/REPLACE
335+PSTWRKD3 DC    F'0' WORK SPACE FOR USE BY DELETE/REPLACE
336+PSTWRKC  DC    3F'0' 1SED BY CONVERSION ROUTINES
337+PSTLEN   EQU   *-PST

339+* COMMUNICATION AREA FOR BLOCK BUILDER UTILITY - DFSUACB0
340+         ORG   PSTDSGA
341+PSTBFLAG DS    XL1 .                  BUILDER INTERNAL FLAGS
342+PRECOMP  EQU   X'80' .                PRE-COMPRESS REQUESTED
343+POSTCOMP EQU   X'40' .                POST COMPRESS REQUESTED
344+PSBALL   EQU   X'20' .                PSB=ALL REQUESTED
345+PSTDATE  DS    PL3 .                  DATE FOR THIS RUN
346+PSTTIME  DS    F .                    TIME AT START OF THIS RUN
347+PSTBLDL  DS    A .                    A(LCB) FOR DBD & PSB BUILD LIST
348+PSTMSGR  DS    A .                    ENTRY POINT TO MESSAGE ROUTINE
349+PSTLBLMO DS    A .                    ENTRY POINT TO BLK BUILDER MSG ROUT
350+PSTDCBAC DS    A .                    ADDRESS OF DCB FOR IMSACB OUTPUT
```

```
 351+PSTDCBSC DS    A .                   ADDRESS OF DCB FOR IMSACB INPUT
 352+PSTPNAME DS    CL8 .                 PSB NAME                    *MBRNAME *
 353+PSTPTTR  DS    XL3 .                 PSBMEMBER TTR               *MBRTTR  *
 354+PSTPMBRC DS    X .                   DIRECTORY FLAG BYTE         *MBRC    *
 355+PSTPTTRI DS    XL3 .                 INTENT&DBD NAME LIST TTR    *MBRTTRI *
 356+PSTPTTN1 DS    X .                   RESERVED--(ALWAYS ZERO)     *MBRTTRN1*
 357+PSTPFLAG DS    X .                   BIT 7 ON= THIS IS A DMB     *MBRFLAG *
 358+PSTPDATE DS    PL3 .                 JULIAN RUN DATE             *MBRDATE *
 359+PSTPTIME DS    F .                   BINARY RUN TIME             *MBRTIME *
 360+PSTPSIZE DS    H .                   # OF DBLWORDS IM MEMBER     *MBRSIZE *
 361+PSTPBFR  DS    H .                   CORE SIZE IN DBLWORDS       *MBRBFR  *
 362+PSTPDMOP DS    F .                   DATA MGMT OPTIONS           *MBRDMOPT*
 363+PSTPXWA  DS    H .                   SIZE OF INDEX WORK AREA
 364+PSTPEND  EQU   * .                   END OF ACBLIB PSB DIR. ENTRY
 365+*
 366+* ACBLIB DIRECTORY ENTRY FOR DMB-- FIELDS COMPARABLE TO PSB ENTRY
 367+*
 368+         DS    OF
 369+PSTDNAME DS    CL8 .                 DMB NAME
 370+PSTDTTR  DS    XL3
 371+PSTDMBRC DS    X
 372+PSTDTTRI DS    XL3 .                 TTR OF DMB SEQ LIST - ZERO IF NONE
 373+PSTDTTN1 DS    X .                   ALWAYS ZERO
 374+PSTDFLAG DS    X .                   BIT 7 ON FOR DMB
 375+PSTDDATE DS    PL3
 376+PSTDTIME DS    F
 377+PSTDSIZE DS    H
 378+PSTDBFR  DS    H
 379+PSTDLSIZ DS    H .                   SIZE OF DMB SEQ LIST
 380+         DS    2H .                  UNUSED
 381+PSTDEND  EQU   *
 382+         ORG   PSTBLKNM
 383+PSTMFLGS DS    XL1 .                 LOADER FLAG BYTE
 384+PSTMLDMB EQU   X'01' .               'LOADDMB' MUST BE CALLED
 385+PSTMSSTC EQU   X'02' .           RESOURCES RESERVED FOR AT LEAST ONE DMB


 388+*...D A T A    M A N A G E M E N T    B L O C K :

 390+DMB      DSECT
 391+DMBSIZE  DS    H DMB SIZE
 392+DMBLENTB DS    H OFFSET FROM DMB TO FIRST PSDB (DMBPSDB)
 393+DMBSECTB DS    H OFFSET FROM DMB TO FIRST SECONDARY LIST (DMBSEC)
 394+DMBORG   DS    XL1 DMB CRGANIZATION

 396+DMBISAM1 EQU   001 ISAM CASE 1
 397+DMBISAM2 EQU   002 ISAM CASE 2 (MULT DSG'S)
 398+DMBSSAM  EQU   003 SSAM (SEQ WITH ONE SEG TYPE)
 399+DMBHSAM  EQU   004 HSAM (SEQ WITH MULT SEG TYPES)
 400+DMBHD    EQU   005 HD DIRECT
 401+DMBHI    EQU   006 HD INDEXED
 402+DMBNDEX  EQU   007 INDEX DATA BASE

 404+DMBLDDCB DS    0XL1 DCB PREFIX NO FOR FELATIONSHIPS AND NDEX ON LOAD
 405+DMBRES1  DS    XL1 RESERVED FOR ALIGNMENT
```

```
 406+DMBNREF  DS     OXL1 NUMBER OF ENTRIES IN EXTERNAL REFERENCE TABLE
 407+DMBDALGR DS     F ADDR OF DIRECT ALGORYTHM CSECT
 408+DMBPPRND EQU    * END OF PRE PREFIX SECTION OF DMB
 409+DMBPPRLN EQU    DMBPPRND-DMB LENGTH OF PRE PREFIX SECTION OF DMB
 410+DMBDCBOF DS     H OFFSET FROM * TO DCB
 411+DMBDCBDL DS     XL1 DELTA CYLINDERS TO SCAN
 412+DMBDCBAP DS     XL1 NUMBER OF ROOT ANCHOR POINTS IN EACH BLOCK
 413+DMBDCBMX DS     H LENGTH OF LARGEST  SEGMENT STORED IN DATA SET
 414+DMBDCBMN DS     H LENGTH OF SMALLEST SEGMENT STORED IN DATA SET
 415+DMBDCBLN EQU    *-DMBPPRND LENGTH OF EACH DCB SECTION OF DMB
 416+DMBPSCB  DSECT
 417+DMBSC    DS     XL1 SEGMENT CODE
 418+DMBPSC   DS     XL1 PARENT
 419+DMBLEV   DS     XL1 LEVEL
 420+DMBXNULL DS     XL1 DO NOT CREATE INDEX ENTRY IF FIELD = THIS BYTE
 421+DMBPPFD  DS     XL1 PTR NO IN PARENT TO FIRST
 422+DMBPPBK  DS     XL1 PTR NO IN PARENT TO LAST
 423+DMBDCB   DS     XL1 DCB NO
 424+DMBPTR   DS     XL1 PREFIX FLAGS

 426+DMBCTR   EQU    X'80' CTR PRESENT
 427+DMBPTFD  EQU    X'40' PT FWD
 428+DMBPTBK  EQU    X'20' PT BKD
 429+DMBPP    EQU    X'10' PP
 430+DMBLTFD  EQU    X'08' LT FWD
 431+DMBLTBK  EQU    X'04' LT BKD
 432+DMBLP    EQU    X'02' LP
 433+DMBHIER  EQU    X'01' HIER

 435+DMBPRSZ  DS     H PREFIX SIZE
 436+DMBDL    DS     H DATA LENGTH OF SEGMENT AS USER SEES IT
 437+DMBISRT  DS     XL1 INSERT RULES

 439+DMBXNLP  EQU    X'80' INSERT RULE FOR INDEX, NULL BYTE PRESENT
 440+DMBIHERE EQU    X'30' FOR NON KEYED SEGMENT, INSERT AT CURRENT POSITION
 441+DMBILST  EQU    X'20' FOR NON KEYED SEGMENT, INSERT AFTER  EXISTING SEG
 442+DMBIFST  EQU    X'10' FOR NON KEYED SEGMENT, INSERT BEFORE EXISTING SEG
 443+DMBIRL   EQU    X'03' INSERT RULE IS LOGICAL
 444+DMBIRP   EQU    X'02' INSERT RULE IS PHYSICAL
 445+DMBIRV   EQU    X'01' INSERT RULE IS VIRTUAL

 447+DMBDLT   DS     XL1 DELETE REPLACE RULES

 449+DMBDRLC  EQU    X'30' INDUCED DELETE RULE IS LOGICAL
 450+DMBDRPC  EQU    X'20' INDUCED DELETE RULE IS PHYSICAL
 451+DMBDRVC  EQU    X'10' INDUCED DELTTE RULE IS VIRTUAL
 452+DMBRRL   EQU    X'0C' REPLACE RULE IS LOGICAL
 453+DMBRRP   EQU    X'08' REPLACE RULE IS PHYSICAL
 454+DMBRRV   EQU    X'04' REPLACE RULE IS VIRTUAL
 455+DMBDRL   EQU    X'03' DELETE RULE FOR SEG IS LOGICAL
 456+DMBDRP   EQU    X'02' DELETE RULE FOR THIS SEG IS PHYSICAL
 457+DMBDRV   EQU    X'01' DELETE RULE FOR SEG IS VIRTUAL

 459+DMBCKL   DS     H CONCATENATED KEY LENGTH TO SEG
 460+DMBUSE   DS     OCL1
```

```
 462+DMBEX      EQU    X'80' THIS PSDB IN USE EXCLUSIVELY
 463+DMBUP      EQU    X'40' THIS PSDB IN USE FOR UPDATE
 464+*                 BITS 2-7 CONTAIN A COUNT OF READ ONLY USERS

 466+DMBFDBA    DS     F ADDRESS OF FDB'S FOR THIS SEGMENT
 467+DMBFSDB    DS     A FIRST SDB FOR THIS SEG
 468+DMBFLAG    DS     0XL1 SECCNDARY LIST FLAG

 470+DMBPAIR    EQU    X'48' A PHYSICAL PAIR EXISTS
 471+DMBLPEX    EQU    X'40' A LOGICAL PARENT EXISTS
 472+DMBLCEX    EQU    X'20' ONE OR MORE LOGICAL CHILDREN EXIST
 473+DMBNXEX    EQU    X'10' ONE OR MORE INDEXES EXIST
 474+DMBXDEX    EQU    X'04' AN INDEXED SEGMENT EXISTS

 476+DMBLST     DS     F ADDRESS OF SECONDARY LIST FOR THIS SEGMENT
 477+DMBPSDBN   EQU    * END OF ONE SEGMENT ENTRY
 478+DMBPLEN    EQU    DMBPSDBN-DMBSC LENGTH FO EACH SEGMENT DESCRIPTION IN DMB
 479+DMBSEC     DSECT
 480+DMBSCDE    DS     XL1 CODE BYTE

 482+DMBSLP     EQU    X'01' SEC LIST DESCRIBES A LOGICAL PARENT
 483+DMBSLC     EQU    X'02' SEC LIST DESCRIBES A LOGICAL CHILD
 484+DMBSRCH    EQU    X'04' SEC LIST DESCRIBES INDEX SEARCH FIELD(S)
 485+DMBSLCF    EQU    X'08' SEC LIST DESCRIBES LT SEQUENCE FIELD
 486+DMBSLCPR   EQU    X'11' SEC LIST DESCRIBES PHY PAIR OF LOGICAL CHILD
 487+DMBSOURC   EQU    X'20' SEC LIST DESCRIBES INDEX SOURCE FIELD(S)
 488+DMBSUBSQ   EQU    X'24' SEC LIST DESCRIBES INDEX SUBSEQ FIELD(S)
 489+DMBEXTRN   EQU    X'40' SEC LIST DESCRIBES USER INDEX EXIT ROUTINE
 490+DMBINDXD   EQU    X'44' SEC LIST DESCRIBES INDEXED SEGMENT
 491+DMBNXISS   EQU    X'60' SEC LIST DESCRIBES INDEX RELATIONSHIP
 492+*                       AS SEEN FROM INDEX SOURCE SEGMENT (ISS)
 493+DMBNXXDS   EQU    X'64' SEC LIST DESCRIBES INDEX RELATIONSHIP AS SEEN
 494+*                       FROM INDEXED SEGMENT (XDS)  THIS LIST NOT
 495+*                       PRESENT IF ISS = XDS
 496+DMBSND     EQU    X'80' LAST IN SECONDARY LIST


 498+DMBSFLDS   EQU    * FOLLOWING FIELDS LISTED BY CODE TYPE

 500+*                 ****  C C D E   0 1  ****

 502+DMBSFLG    DS     XL1
 503+DMBVKY     EQU    C'V' KEY OF LP IS VIRTUAL
 504+DMBPKY     EQU    C'P' KEY OF LP IS PHYSICAL
 505+DMBSFD     DS     H LP KEY LENGTH
 506+DMBSECSC   DS     0CL1 SEGMENT CODE OF REFERENCED SEGMENT
 507+DMBSECDB   DS     A DDIR ACDR OF REFERENCED DATA BASE
 508+DMBSECNM   DS     CL8 SEGMENT NAME OF EXTERNAL SEGMENT
 509+           ORG    DMBSFLDS

 511+*                 ****  C C D E   0 2  ****

 513+           DS     XL1 NOT USED
 514+DMBSLCFL   DS     H NO OF FIRST AND LAST LC PTRS IN LP PREFIX
 515+*          REMAINDER SAME AS CODE 1
```

```
 516+           ORG    DMBSFLDS

 518+*                 ****  C C D E   0 4  ****

 520+DMBFDFLG DS       5XL1 5 ONE BYTE FLAGS ASSOCIATED WITH THE
 521+*                      THE FOLLOWING 5 HALFWORD,FDB OFFSETS

 523+DMBSYM1  EQU      X'08' FIRST PART OF SYMBOLIC POINTER
 524+DMBSYMN1 EQU      X'04' NOT FIRST PART OF SYMBOLIC PTR (MIDDLE OR LAST)
 525+DMBSYSFD EQU      X'02' THIS SLOT FOR SYSTEM RELATED FIELD
 526+DMBFDUSE EQU      X'01' THIS SLOT IN USE
 527+DMBFDOFF DS       5H OFFSET TO FDB FROM 1ST FDB OF ISS IF
 528+*                      THIS SLOT IN USE, OTHERWISE ZERO
 529+           ORG    DMBSFLDS

 531+*                 ****  C C D E   0 8  ****

 533+           DS     XL1 NOT USED
 534+DMBSFNAM DS       CL8 FDB FIELD NAME
 535+DMBSFOFF DS       H OFFSET TO FIELD START
 536+DMBSFCEN DS       XL1 CODE BYTE (SAME AS FDBDCENF)
 537+DMBSFLEN DS       XL1 EXECUTABLE FIELD LENGTH
 538+DMBXSOFF DS       H OFFSET OF FIELD IN INDEXED SEGMENT
 539+           ORG    DMBSFLDS

 541+*                 ****  C C D E   1 1  ****

 543+*          SAME   AS CODE 1

 545+*                 ****  C C D E   2 0  ****

 547+*          SAME   AS CODE 4

 549+*                 ****  C C D E   2 4  ****

 551+*          SAME   AS CODE4

 553+*                 ****  C C D E   4 0  ****

 555+DMBSFLG1 DS       XL1 FLAG BYTE

 557+DMBSNULL EQU      X'01' NULL FIELD PRESENT
 558+DMBEXIT  EQU      X'02' EXIT ROUTINE PRESENT
 559+DMBNLXIT EQU      X'03' BOTH NULL FIELD AND EXIT ROUTINE PRESENT
 560+DMBEXLOD EQU      X'04' EXIT ROUTINE HAS BEEN LOADED

 562+           DS     H NOT USED
 563+DMBNBYTE DS       OCL1 IF INDEX FIELD = THIS BYTE BYPASS INDEXING
 564+DMBXITAD DS       A ENTRY POINT OF EXIT ROUTINE
 565+DMBXITNM DS       CL8 NAME OF INDEX EXIT ROUTINE
 566+           ORG    DMBSFLDS

 568+*                 ****  C C D E   4 4  ****

 570+DMBSKYLN DS       XL1 EXECUTABLE LENGTH OF KEY
```

```
 571+DMBSOFF  DS     H OFFSET TO CODE 60 OR 64 FROM START OF SECONDARY
 572+*                     LIST OF INDEXED SEGM
 573+DMBXDSSC DS     0XL1 SEGMENT CODE OF INDEXED SEGMENT
 574+DMBXDSDB DS     A DDIR ADDR OF INDEXED SEGMENT
 575+DMBXDSC  DS     0XL1 SEGMENT CODE OF INDEXED SEGMENT
 576+DMBXPSDB DS     A PSDB ADDR OF INDEXED SEGMENT
 577+DMBXDFLG DS     XL1 CODE BYTE FROM ASSOCIATED FDB

 579+DMBXDLST EQU    X'80' IN FDB LIST LAST FDB
 580+DMBXDSYM EQU    X'40' INDEX POINTER IS SYMBOLIC
 581+DMBXDSSS EQU    X'20' PTR CONTAINED IN SOURCE/SUBSEQ DATA
 582+DMBXDSPC EQU    X'10' SPECIAL FDB
 583+DMBXDCON EQU    X'08' CONSTANT PRESENT
 584+DMBXDSSQ EQU    X'04' SUBSEQ PRESENT
 585+DMBXDSOR EQU    X'02' RESERVED
 586+DMBXDEQ  EQU    X'01' XDS = ISS

 588+DMBXDPAD DS     XL1 PADDING CONSTANT
 589+DMBSYMOF DS     H OFFSET TO SYMBOLIC POINTER INDEXING SEGMENT
 590+         ORG    DMBSFLDS

 592+*               ****  C O D E   6 0  ****

 594+         DS     XL3 SAME AS CODE 44
 595+DMBXNSSC DS     0XL1 SEGMENT CODE OF INDEXING SEGMENT
 596+DMBXNSDB DS     A DDIR ADDR OF INDEX
 597+*        REMAINDER SAME AS CODE 44
 598+         ORG    DMBSFLDS

 600+*               ****  C O D E   6 4  ****

 602+         DS     XL1 SAME AS CODE 44
 603+DMBISSOF DS     H OFFSET TO CODE 60 FROM START OF ISS SEC LIST
 604+         DS     A SAME AS CODE 60
 605+DMBISSSC DS     0XL1 SEGMENT CODE OF INDEX SOURCE SEGMENT
 606+DMBIPSDB DS     A PSDB ADDR OF INDEX SOURCE SEGMENT
 607+         DS     F SAME AS CODE 44
 608+DMBSECND EQU    * END OF EACH SECONDARY LIST ENTRY
 609+DMBSECLN EQU    DMBSECND-DMBSEC LENGTH OF EACH SECONDARY LIST

 611+DMBDACS  DSECT
 612+DMBDANME DS     CL8 NAME OF ADDR ALGORYTHM LOAD MODULE
 613+DMBDAKL  DS     0CL1 EXEC ROOT KEY FLD LNGTH
 614+DMBDAEP  DS     A E P OF ADDR LOAD MODULE
 615+DMBDASZE DS     H SIZE OF THIS CSECT
 616+DMBDARAP DS     H NUMBER OF ROOT ANCHOR POINTS/BLOCK
 617+DMBDABLK DS     F NUMBER OF HIGHEST BLOCK DIRECTLY ADDRSD
 618+DMBDABYM DS     F MAX NUM OF BYTES BEFORE OFLOW TO 2NDARY
 619+DMBDABYC DS     F CUR NUM OF BYTES INSERTED UNDER ROOT
 620+DMBDACP  DS     F RESULT OF LAST ADDR CONVERSION
 621          END                                                          00055000
```

BINARY HALVING METHOD EXAMPLE

This module attempts to distribute root segments across the root segment addressable area according to the bit pattern of a root segment key field value after it has been converted to a binary value. This distribution is performed as follows: A result register is set to zero. After a key field value has been converted to binary, the number of base locations (number of blocks in the root segment addressable area times number of anchor points per block) is computed and divided by two. The low-order bit of the converted key field value (ckey) is tested for one. If equal to one, the current number of base locations is added to the result register. If the low-order bit is zero, no addition to the result register is performed.

The number of remaining base locations is again divided by two and the quotient tested for zero. If nonzero, the next higher bit position in the ckey is tested for a one or zero and the appropriate action taken. This process continues until the number of remaining base locations divided by two yields a quotient of zero. At this point, the appropriate base location is in the result register. In order to produce the proper relative block number and anchor point number, divide by the number of anchor points per block. The division yields a quotient of relative block number and remainder of anchor point number. As in the previous module, the results are both relative to zero and must be incremented by one to yield the appropriate values.

Example:

Assume

a) 10 blocks in root segment addressable area

b) 2 anchor points per block

c) root segment key field value of 29

After initialization:

CKEY	No. of remaining base locations	Result Register
1 1 1 0 1	(10x2)/2 = 10	0

After bit tested

. . . . x	10	10
. . . x .	5	10
. . x . .	2	12
. x . . .	1	13

At this point, the number of remaining base locations is reduced to zero. Hence the appropriate base location is 13. To get the actual relative block number and root anchor point, divide 13 by 2 and add 1 to both the quotient and the remainder to yield a relative block number of 7 and an anchor point number of 2.

Notice that the number of base locations determines when testing ceases. Hence, in this example, all key field values ending in the same four bits will be synonyms. Additional bits of the key are tested when the number of base locations exceeds another power of two. If the number of base locations is not a power of two, some of the base locations will never be used.

The major advantage of this method is that the relative order of root segment placement is disturbed very little when the number of base locations is changed.

```
  3 *                                                                            00002000
  4 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00003000
  5 *                                                                          * 00004000
  6 *                   B I N A R Y   H A L V I N G   C O N V E R T            * 00005000
  7 *                                                                          * 00006000
  8 *                THIS CSECT DETERMINES THE RELATIVE BLOCK AND ROOT         * 00007000
  9 *           ANCHOR POINT BY A BINARY HALVING TECHNIQUE. THIS APPROACH      * 00008000
 10 *           IS SLOWER THAN THE MODULO SCHEMES, BUT IT DOES TEND TO KEEP    * 00009000
 11 *           THE SAME PHYSICAL SEQUENCE WHEN THE NUMBER OF ADDRESSABLE      * 00010000
 12 *           BLOCKS IS CHANGED. SINCE THE ROUTINE USES SHIFTS ON INTEGER    * 00011000
 13 *           NUMBERS, SOME RECORD NUMBERS WILL BE INACCESSABLE IF THE       * 00012000
 14 *           TOTAL NUMBER OF DIRECTLY ADDRESSABLE RECORDS (BLCCKS*ROOT      * 00013000
 15 *           ANCHOR POINTS) IS NOT A POWER OF 2                             * 00014000
 16 *                                                                          * 00015000
 17 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00016000
 18          STM   14,12,12(13)                                                  00017000
 19          USING PST,R7                                                        00018000
 20          USING DMBDACS,R1                                                    00019000
 21          USING DFSHDC20,R15                                                  00020000
 22          XC    PSTDECB(8),PSTDECB  INIT FOR CVB                              00021000
 23          IC    R5,DMBDAKL         GET EX KEY LENGTH                          00022000
 24          EX    R5,PACK                                                       00023000
 25          OI    PSTDECB+7,X'0F'    FORCE VALID SIGN                           00024000
 26          CVB   R2,PSTDECB                                                    00025000
 27          L     R4,DMBDABLK                                                   00026000
 28          MH    R4,DMBDARAP        HIGHEST RECORD IN RANGE                    00027000
 29          SR    R5,R5              CLEAR RESULT REG                           00028000
 30 CVTLP    SRL   R4,1               CUT RANGE IN HALF                          00029000
 31          LTR   R4,R4              RANGE EXHAUSTED                            00030000
 32          BZ    XIT                YES                                        00031000
 33          SR    R3,R3              NO                                         00032000
 34          SRDL  R2,1               TEST MASK FOR 1                            00033000
 35          LTR   R3,R3                                                         00034000
 36          BZ    CVTLP              NO ONE                                     00035000
 37          BXH   R5,R4,CVTLP        ONE - ADD IN RANGE                         00036000
 38 XIT      DS    0H                                                            00037000
 39          LH    R6,DMBDARAP                                                   00038000
 40          DR    R4,R6                                                         00039000
 41          LA    R4,1(,R4)          ROOT ANCHOR POINT                          00040000
 42          LA    R5,1(,R5)          BLOCK                                      00041000
 43          SLL   R5,8                                                          00042000
 44          OR    R4,R5                                                         00043000
 45          ST    R4,DMBDACP         RESULT                                     00044000
 46          LM    14,12,12(13)                                                  00045000
 47          BR    R14                                                           00046000
 48 PACK     PACK  PSTDECB(8),0(0,R9)                                            00047000
 49          PRINT NOGEN                                                         00048000
 50          IDLI  PSTBASE=0,DMBBASE=0                                           00049000
595          REQUATE                                                             00050000
620          END                                                                 00051000
```

HASHING METHOD EXAMPLE

This module uses a shift and add technique to develop a 31-bit binary number which should have a fairly even distribution from 0 to 2^{31}. The number is developed as follows: The result register is initialized to zero. The first character of a key field value is added to the result register and the register is shifted left three hexadecimal digits. The bits of the register shifted left and off the register are then added back to the register containing the previous shift result. This partial result is tested to be odd or even. If odd, the contents of the register are complemented. The original character is then added to the register. This process is repeated for each character in the key field value. Instead of starting off with a zero content in the result register, the result of the previous content is used. When the key field value characters are exhausted, the result is adjusted to guarantee a 31-bit positive result.

Example:

Assume a) Key field value = ABCD

Key Character	Result Register	
A	0C100000	After test for complement
	0C10C100	After completion of A
B	1C20C10C	After test for complement
	1CE1C20C	After completion of B
C	2CF1CE1C	After test for complement
	EDF2CF1C	After completion of C
D	FE0EDF2C	After test for complement
	FF0FE0ED	After completion of D
	7F0FE0ED	Positive number

The result can then be used as input to the modulo or binary halving technique. The latter technique is used in this example.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00001000
*                  S A M P L E    H A S H I N G    T E C H N I Q U E* 00002000
*                                                                    * 00003000
*               THIS CSECT IS A ONE METHOD OF HASHING AN EXTERNAL KEY  * 00004000
*         INTO A 31 BIT BINARY NUMBER WHICH CAN THEN BE USED AS INPUT  * 00005000
*         TO THE BINARY HALVING ADDRESSES RESOLUTION OR A MODULO SCHEME* 00006000
*         TO DETERMINE THE BLOCK AND ROOT ANCHOR POINT.                * 00007000
*               THIS ROUTINE PLACES FEW RESTRICTIONS ON THE EXTERNAL   * 00008000
*         KEY E.G. IT CAN BE 156 BYTES LONG, IT CAN CONTAIN ANY BIT    * 00009000
*         PATTERN. THE KEY SHOULD BE LONGER THAN 3 CHARACTERS TO INSURE* 00010000
*         SOME SPREADING, HOWEVER IT WILL WORK ON SHORTER KEYS.        * 00011000
*                                                                      * 00012000
*         CALLING SEQUENCE                                             * 00013000
*               R0 - DMB                                               * 00014000
*                1 - DMBDACS                                           * 00015000
*                7 - PST                                               * 00016000
*                9 - KEY ADDRESS                                       * 00017000
*         ON RETURN                                                    * 00018000
*               DMBDACP - BBBR                                         * 00019000
*                                                                      * 00020000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00021000
DFSHDC30 CSECT                                                          00022000
         STM   R14,R12,12(R13)                                          00023000
         USING DFSHDC30,R15                                             00024000
         USING DMBDACS,R1                                               00025000
         SR    R12,R12                                                  00026000
         BCTR  R12,0                  SET TO ALL FF S                   00027000
         SR    R11,R11                                                  00028000
         LA    R9,0(,R9)              CLEAR ANY HIGH ORDER BITS         00029000
         SR    R7,R7                  INIT                              00030000
         IC    R7,DMBDAKL                 FOR                           00031000
         AR    R7,R9                         LATER                      00032000
         LA    R6,1                                BXLE                 00033000
         SR    R2,R2                                                    00034000
LOOP     DS    0H                                                       00035000
         IC    R11,0(,R9)             GET GROUP OF 8 BITS               00036000
         ALR   R2,R11                 ADD TO HASH                       00037000
         SR    R3,R3                                                    00038000
         SRDL  R2,12                  BREAK UP CHAR PATTERNS            00039000
         OR    R2,R3                  ADD INTO HIGH PORTION             00040000
         STC   R2,DMBDACP             COMPLEMENT                        00041000
         TM    DMBDACP,X'01'                    ON                      00042000
         BZ    PASS                               MODERATELY            00043000
         XR    R2,R12                             CHANGING              00044000
PASS     SR    R3,R3                            BIT                     00045000
         ALR   R2,R11                 DO SECOND PASS                    00046000
         SRDL  R2,12                  WITHOUT                           00047000
         OR    R2,R3                  COMPLIMENT                        00048000
         BXLE  R9,R6,LOOP             EXHAUST KEY                       00049000
         N     R2,NOSIGN              FORCE POSITIVE 31 BIT RESULT      00050000
*        USE R2 AS INPUT TO HALVING OR MODULO SCHEME - HALVING SHOWN    00051000
         L     R4,DMBDABLK                                              00052000
         MH    R4,DMBDARAP            HIGHEST RECORD IN RANGE           00053000
         SR    R5,R5                  RESULT REG                        00054000
CVTLP    SRL   R4,1                   CUT RANGE IN HALF                 00055000
```

```
STMT    SOURCE STATEMENT                                                  F15OCT70    3/27/72

  57            LTR   R4,R4              RANGE EXHAUSTED                           00056000
  58            BZ    XIT                YES                                       00057000
  59            SR    R3,R3              NO                                        00058000
  60            SRDL  R2,1               TEST MASK FOR ONE                         00059000
  61            LTR   R3,R3                                                        00060000
  62            BZ    CVTLP              NO ONE                                    00061000
  63            BXH   R5,R4,CVTLP        ONE - ADD IN RANGE                        00062000
  64 XIT        LH    R6,DMBDARAP                                                  00063000
  65            DR    R4,R6                                                        00064000
  66            LA    R4,1(,R4)          ROOT ANCHOR POINT                         00065000
  67            LA    R5,1(,R5)          BLOCK                                     00066000
  68            SLL   R5,8                                                         00067000
  69            OR    R4,R5                                                        00068000
  70            ST    R4,DMBDACP         RESULT                                    00069000
  71            LM    R14,R12,12(R13)                                              00070000
  72            BR    R14                RETURN                                    00071000
  73            DS    0F                                                           00072000
  74 NOSIGN     DC    X'7FFFFFFF'                                                  00073000
  75            PRINT NOGEN                                                        00074000
  76            IDLI  DMBBASE=C                                                    00075000
 313            REQUATE                                                            00076000
 338            END                                                                00077000
```

DATA BASE LOG TAPE RECORD FORMAT

The following DSECT provides an image of the log tape record format for all data base modifications. This log tape record format is provided to facilitate the writing of any user-written statistics, recovery analysis, or batch checkpoint/restart programs.

```
DBLOG       DSECT
DLENGTH     DS      H         LENGTH OF LOG RECORD
DSPACE      DS      H         ZEROS
DLOGCODE    DS      CL1       LOG RECORD I.D.
DLOGFLG1    DS      CL1
*                         BITS 0-3 =  REGION PROTECT KEY
*                         BITS 4-7 =  COUNT OF FSE'S IN LOG RECORD

DLOGFLG2    DS      CL1
DNDXC       EQU     X'80'     INDEX MAINTENANCE RECORD
DCMC        EQU     X'00'     BITS 1-3 =  000 CHAIN MAINTENANCE RECORD
DPHYI       EQU     X'40'     PHYSICAL INSERT
DPHYD       EQU     X'20'     PHYSICAL DELETE
DNCTR       EQU     X'70'     COUNTER MAINTENANCE CALL

DPHYR       EQU     X'10'     PHYSICAL REPLACE
DLASTREC    EQU     X'08'     LAST RECORD FOR THIS USER CALL
DOSAM       EQU     X'00'     BIT 5=0 OSAM DATA SET
DISAM       EQU     X'04'     BIT 5=1 ISAM DATA SET
DHS         EQU     X'00'     BIT 6=0 HS ORGANIZATION
DHD         EQU     X'02'     BIT 6=1 HD ORGANIZATION
DNEWBLK     EQU     X'01'      NEW BLOCK CALL

DLOGFLG3    DS      CL1
DRCALL      EQU     X'80'     REPL CALL
DDCALL      EQU     X'40'     DLET CALL
DICALL      EQU     X'20'     ISRT CALL
DREG0       EQU     X'00'     BITS 3-4 = 00 MOD BY TYPE 0 REGION
DREG3       EQU     X'10'     MOD BY TYPE 3 REGION
DREG12      EQU     X'08'     MOD BY TYPE 1/2 REGION
DRSV2       EQU     X'04'     UNUSED
DFIRSTSG    EQU     X'02'     FIRST LOG RECORD OF A SEGMENT
DLASTSEG    EQU     X'01'     LAST LOG RECORD OF A SEGMENT

DIDLN       DS      CL2       LENGTH OF DDATAID FIELD
DOFFSET     DS      CL2       DATA OFFSET FROM BEGINNING OF A BLOCK
DDATALN     DS      CL2       LENGTH OF DDATA FIELD
DCCODE      DS      CL2       DL/I COMPLETION CODE
DDBDNAME    DS      CL8       DATA BASE NAME
DDPGMNAM    DS      CL8       PROGRAM NAME
DDSID       DS      CL1       DATA SET I.D.
DDATE       DS      CL3       DATE
DTIME       DS      CL4       TIME
DSEQ        DS      CL2       SEQUENCE NUMBER
DDATAID     DS      0CL1      ISAM PRIME KEY OR OSAM RBN
DDATA       DS      0CL1      SEGMENT DATA
DFSEOFF     DS      0CL2      FREE SPACE ELEMENT OFFSET
DFSE        DS      0CL4      FREE SPACE ELEMENT
```

2972/2980 INPUT EDIT ROUTINE

A 2980 input edit routine is required by IMS/360 2972/2980 device support to perform terminal-related functions inherent in the design of the 2972/2980 General Banking Terminal system. Usage and value of these functional characteristics are installation oriented and are therefore not performed by normal IMS/360 procedures. The entry (CSECT) name of this routine must be DFS29800 and must be link edited with the IMS/360 control region nucleus, since it will be called directly by the IMS/360 2972/2980 device-dependent module (DFSDN110). Control is passed to the 2972/2980 Input Edit routine to process each entered message segment after that message segment has been translated by IMS/360.

The 2972/2980 Input Edit routine must perform the following functions:

1. Determine the IMS/360 destination (SMB or CNT) of messages entered from a 2980 teller or administrative station.

2. Determine end-of-message of multisegment messages (by setting DECCSWST bit 7 to indicate EOM).

3. Reposition the entered data to the beginning of the input buffer for IMS/360 processing (the entered segment must be in standard IMS/360 input message format after edit processing).

In addition to performing the above required functions, the 2972/2980 Input Edit routine may add inputting terminal status information to the entered segment, such as the presence or absence of a passbook or auditor key on the inputting terminal. The Input Edit routine may initiate retransmission of the last successfully transmitted message to a 2980 logical terminal through a return code to the calling routine.

If the INTERACTIVE Query Facility (IQF) is incorporated into the IMS/360 system and will receive input from the 2980, the following additional steps must be taken by the input edit routine:

1. The inputting terminal status information must be separated from IQF elements by at least one blank.

2. If the inputting terminal status information is appended to the end of a segment, any preceding carriage return must be removed (replaced with a blank).

3. The inputting terminal status information must be defined to IQF as a null word.

4. In the edited segment, the inputting terminal status information must not be the initial characters of the segment.

Familiarization with IMS/360 terminal handling procedures and control blocks is required for a user to write an input edit routine which must interface with IMS/360 routines in the IMS/360 control region. Examination of these control blocks may be required; however, modification of IMS/360 control blocks by a user-written routine seriously endangers the integrity of the entire system. A sample 2972/2980 Input Edit routine appears later in this chapter. The IMS/360 interfaces to the 2972/2980 Input Edit routine are as follows:

R0	Input buffer length
R1	Start of input area. The first nine bytes of this area are blanks (X'40') followed by the terminal address character, followed by the entered text. (The first byte of text if the data was entered from a 2980 Model 4 is the teller identification character.)
R2	Input data length. (The length of the area pointed to in Register 1.)
R7	CTB Address
R9	CLB Address
R11	SCD Base
R13	Caller's save area address in the IMS/360 prechained save area set.
R14	Return Address
R15	The 2972/2980 Input Edit routine base address

Registers at return to the calling program must be as follows:

R2	Data length after edit (a zero length signifies a no data segment)
R10	The inputting CNT address if a retransmission of the last successfully outputted message is required.
R15	Return Code: 0 - Process the entered segment 4 - Resend the last message to the CNT in Register 10

IMS/360 provides a sample 2972/2980 Input Edit routine which is described in this chapter.

The format of the buffer pointed to in Register 1 at entry to the 2972/2980 input edit routine is as follows:

```
|-----------------|----|--------------------------------
    9 Blanks                   Entered Text

                   T
                   E
                   R    The first character of the entered
                   M    text if entry is from a 2980 Model 4 is
                   I    the teller identification character.
                   N
                   A
                   L

                   A
                   D
                   D
                   R
                   E
                   S
                   S
```

STMT SOURCE STATEMENT F15OCT70 1/14/72

```
EDIT2980 CSECT                                                          00002000
*********************************************************************** 00003000
*                                                                     * 00004000
*         THIS IS A SAMPLE OF THE 2980 INPUT EDIT ROUTINE REQUIRED BY * 00005000
* IMS/360 2972/2980 DEVICE SUPPORT.  THE INPUT EDIT ROUTINE MUST PER- * 00006000
* FORM THE FOLLOWING FUNCTIONS:                                       * 00007000
*                                                                     * 00008000
*      1. DETERMINE THE IMS/360 DESTINATION (SMB OR CNT) OF MESSAGES  * 00009000
*         ENTERED FROM A 2980 TELLER OR ADMINISTRATIVE STATION.       * 00010000
*                                                                     * 00011000
*      2. DETERMINE END-OF-MESSAGE OF MULTI-SEGMENT MESSAGES AND SET  * 00012000
*         DECCSWST BIT 7 AT END-OF-MESSAGE.                           * 00013000
*                                                                     * 00014000
*      3. REPOSITION THE ENTERED DATA TO THE BEGINING OF THE INPUT    * 00015000
*         BUFFER FOR IMS/360 PROCESSING.  THE ADDRESS OF THE INPUT    * 00016000
*         BUFFER IS PASSED TO THE EDIT ROUTINE IN REGISTER 1.         * 00017000
*                                                                     * 00018000
*         IN ADDITION TO PERFORMING THE ABOVE FUNCTIONS THIS SAMPLE   * 00019000
* ROUTINE ALSO DOES THE FOLLOWING:                                    * 00020000
*                                                                     * 00021000
*      1. DETERMINES THE INPUTING LOGICAL TERMINAL (CNT) FOR MESSAGES * 00022000
*         ENTERED FROM A 2980-4 TO BE USED FOR SECURITY VALIDATION AND * 00023000
*         AS THE I/O PCB FOR THE APPLICATION PROGRAM.                 * 00024000
*                                                                     * 00025000
*      2. INITIATES RE-TRANSMISSION OF THE LAST SUCCESSFULLY OUTPUTED * 00026000
*         MESSAGE TO ANY PHYSICAL TERMINAL.                           * 00027000
*                                                                     * 00028000
*         DETERMINATION OF INPUT DESTINATION IS NOT PERFORMED ON DATA * 00029000
* ENTERED FROM A 2980-2 ADMINISTRATIVE STATION AS THIS TERMINAL CAN   * 00030000
* READILY USE THE STANDARD IMS/360 MESSAGE FORMAT.  DATA ENTRY FROM A * 00031000
* 2980-1 OR 2980-4 TELLER STATION REQUIRE THE ENTRY OF A TRANSACTION  * 00032000
* CODE SEQUENCE IN THE FIRST SEGMENT OF ALL ENTERED MESSAGES (IMS/360 * 00033000
* COMMANDS MUST BE ENTERED IN STANDARD IMS/360 FORMAT).  THE TRANSACT-* 00034000
* ION SEQUENCE MAY OCCUR ANYWHERE IN THE FIRST SEGMENT AND CONSIST OF * 00035000
* A DESIGNATED BEGIN CHARACTER, FOLLOWED BY A VALID IMS/360 TRANSACT- * 00036000
* ION CODE TERMINATED BY ANY CHARACTER WHICH WHEN TRANSLATED BY IMS   * 00037000
* HAS A HEXADECIMAL VALUE LESS THAN X'C1', OR END OF MESSAGE SEGMENT. * 00038000
* IF A SCAN OF THE FIRST MESSAGE SEGMENT DOES NOT ENCOUNTER A VALID   * 00039000
* TRANSACTION SEQUENCE (IE: A BEGIN CHARACTER FOLLOWED BY NO MORE     * 00040000
* THAN EIGHT (8) CHARACTERS BEFORE THE TERMINATION CHARACTER),  THIS  * 00041000
* ROUTINE ASSUMES THE MESSAGE WAS ENTERED IN STANDARD IMS/360 INPUT   * 00042000
* MESSAGE FORMAT AND BYPASSES THE DESTINATION EDIT FUNCTION.  THE     * 00043000
* DESIGNATED BEGIN CHARACTERS SCANNED FOR ARE:                        * 00044000
*                                                                     * 00045000
*         X'41' NUMERIC ENTRY OF KEY 0 (MSGACK) FROM A 2980-1.        * 00046000
*         X'59' NUMERIC ENTRY OF KEY 15 (CODE) FROM A 2980-4.         * 00047000
*                                                                     * 00048000
*         END-OF-MESSAGE IS DETERMINED BY THE ENTRY OF A PERIOD(.) AS * 00049000
* THE LAST CHARACTER OF THE LAST SEGMENT OF A MULTI-SEGMENT MESSAGE,  * 00050000
* OR AS THE LAST CHARACTER OF A SINGLE SEGMENT MESSAGE.               * 00051000
*                                                                     * 00052000
*         INPUTING TERMINAL STATUS INFORMATION IS APPENDED TO EACH MSG * 00053000
* SEGMENT IN THE FOLLOWING FORMAT:                                    * 00054000
*                                                                     * 00055000
```

```
*                                                                     * 00057000
*        AABC                                                         * 00058000
*                                                                     * 00059000
*        WHERE:   AA- IS A TWO (2) BYTE HEXADECIMAL FIELD CONTAINING  * 00060000
*                     TWO NINES (X'F9F9')                             * 00061000
*                  B- IS A 'P' (X'D7') TO INDICATE A PASSBOOK WAS     * 00062000
*                     PRESENT AT SEGMENT ENTRY (OR THE AUDITOR'S      * 00063000
*                     KEY WAS INSERTED ON A 2980-2);  OTHERWISE       * 00064000
*                     THIS CHARACTER IS AN 'N' (X'D5').               * 00065000
*                  C- IS THE TELLER IDENTIFICATION CHARACTER FOR A    * 00066000
*                     2980-4.                                         * 00067000
*                      A - TELLER A WITHOUT SUPERVISOR KEY            * 00068000
*                      B - TELLER B WITHOUT SUPERVISOR KEY            * 00069000
*                      J - TELLER A WITH SUPERVISOR KEY               * 00070000
*                      K - TELLER B WITH SUPERVISOR KEY               * 00071000
*                     IF ENTRY WAS NOT FROM A 2980-4 THIS CHARACTER   * 00072000
*                     IS BLANK (X'40').  THE TELLER IDENTIFICATION    * 00073000
*                     CHARACTER IS REMOVED FROM THE INPUT TEXT.       * 00074000
*                                                                     * 00075000
*        DETERMINATION OF THE INPUTING LOGICAL TERMINAL (CNT) IS MADE * 00076000
* BY EXAMINATION OF THE NAMES OF THE CNTS ASSIGNED TO THE INPUTING    * 00077000
* PHYSICAL TERMINAL.  EACH CNT IS EXAMINED TO FIND ONE WITH A NAME    * 00078000
* WHOSE FIRST CHARACTER MATCHES THE TELLER IDENTIFICATION CHARACTER;  * 00079000
* IF ONE IS FOUND THE CNT CHAIN IS ALTERED TO MAKE THAT CNT THE FIRST * 00080000
* CNT IN THE CHAIN OF CNTS.  THE CNT CHAIN REMAINS UNALTERED IF NO CNT* 00081000
* IS FOUND.                                                           * 00082000
*                                                                     * 00083000
*        ENTRY OF THE CHARACTERS '&RESEND' AS THE ONLY CHARACTERS OF  * 00084000
* A MESSAGE WILL CAUSE THE LAST SUCCESSFULLY OUTPUTED MESSAGE TO BE   * 00085000
* RE-TRANSMITTED TO THE INPUTING TERMINAL.                            * 00086000
*                                                                     * 00087000
*   REGISTERS AT ENTRY:                                               * 00088000
*                                                                     * 00089000
*        R0     INPUT BUFFER LENGTH                                   * 00090000
*        R1     POINTS TO THE INPUT MESSAGE SEGMENT;  PREFIXED BY     * 00091000
*               NINE BLANKS, THE TERMINAL ADDRESS CHARACTER, THE      * 00092000
*               TELLER IDENTIFICATION CHARACTER(IF ENTERED FROM       * 00093000
*               A 2980-4), AND THE ENTERED TEXT.                      * 00094000
*        R2     DATA LENGTH                                           * 00095000
*        R7     CTB BASE                                              * 00096000
*        R9     CLB BASE                                              * 00097000
*        R11    SCD BASE                                              * 00098000
*        R13    CALLER'S SAVE AREA (MY SAVE AREA IS PRE-CHAINED)      * 00099000
*        R14    RETURN ADDRESS                                        * 00100000
*        R15    ENTRY POINT ADDRESS                                   * 00101000
*                                                                     * 00102000
*   RETURN REGISTERS:                                                 * 00103000
*                                                                     * 00104000
*        R2     DATA LENGTH AFTER EDIT                                * 00105000
*        R10    CNT BASE                                              * 00106000
*        R15    RETURN CODE                                           * 00107000
*                                                                     * 00108000
*********************************************************************** 00109000
```

```
STMT    SOURCE STATEMENT                                          F15OCT70   1/14/72
 111           REQUATE                                                      00111000

 113+**********************************************************************
 114+*                                                                    *
 115+*         EQUATE REGISTERS                                           *
 116+*                                                                    *
 117+**********************************************************************

 119+R0        EQU   0
 120+R1        EQU   1
 121+R2        EQU   2
 122+R3        EQU   3
 123+R4        EQU   4
 124+R5        EQU   5
 125+R6        EQU   6
 126+R7        EQU   7
 127+R8        EQU   8
 128+R9        EQU   9
 129+R10       EQU   10
 130+R11       EQU   11
 131+R12       EQU   12
 132+R13       EQU   13
 133+R14       EQU   14
 134+R15       EQU   15
```

```
STMT   SOURCE STATEMENT                                          F15OCT70   1/14/72

 136            USING CTB,R7                                             00112000
 137            USING IECTDECB,R9                                        00113000
 138            USING CNT,R10                                            00114000
 139            USING SCD,R11                                            00115000
 140            USING EDIT2980,R12                                       00116000
 141            SAVE  (14,12),,EDT2980     SAVE INPUT REGISTERS          00117000
 142+           B     12(0,15) BRANCH AROUND ID
 143+           DC    AL1(7) LENGTH OF IDENTIFIER
 144+           DC    CL7'EDT2980' IDENTIFIER
 145+           STM   14,12,12(13) SAVE REGISTERS
 146            LR    R12,R15              SET PROGRAM BASE              00118000
 147            L     R13,8(,R13)          STEP TO NEXT SAVE AREA        00119000
 148            SR    R15,R15              CLEAR RETURN CODE             00120000
 149            LR    R5,R1                SAVE MESSAGE POINTER          00121000
 150            SH    R2,=H'10'            REMOVE BLANKS FROM LENGTH     00122000
 151            LTR   R6,R2                SET LENGTH REG                00123000
 152            BNP   ZEROLNG              BRANCH IF NO DATA             00124000
 153            MVI   DESTLNG,0            SET DESTINATION LENGTH TO ZERO 00125000
 154            MVI   TELLERID,C' '        CLEAR TELLER ID               00126000
 155            LA    R14,10(,R5)          SET BEGIN OF TEXT             00127000
 156            TM    CTBFEAT,CTBFMOD4     2980-4?                       00128000
 157            BZ    CKRESEND             NO                            00129000
 158            MVC   TELLERID(1),0(R14)   SAVE TELLER ID                00130000
 159            LA    R14,1(,R14)          STEP TO TEXT                  00131000
 160            BCTR  R6,0                 DECREMENT DATA LENGTH         00132000
 161            LTR   R6,R6                NO DATA?                      00133000
 162            BNP   ZEROLNG              YES                           00134000
 163 CKRESEND   EQU   *                                                  00135000
 164            CLI   0(R14),C'&&'         POSSIBLE RESEND REQUEST?      00136000
 165            BE    RESEND               YES                           00137000
 166 SETSCAN1   EQU   *                                                  00138000
 167            TM    CTBFEAT,CTBFMOD2     2980-2?                       00139000
 168            BO    SETSTAT              YES                           00140000
 169            BCTR  R6,0                 REDUCE LENGTH FOR SCAN        00141000
 170            EX    R6,SCAN1             FIND BEGIN CHARACTER          00142000
 171            LA    R6,1(,R6)            RE-ADJUST LENGTH              00143000
 172            BC    10,SETSTAT           BRANCH IF NOT FOUND           00144000
 173            LA    R4,1(,R1)            1ST CHAR OF DESTINATION       00145000
 174            LA    R3,0(R6,R14)         POINT TO END OF SEGMENT       00146000
 175            LR    R1,R3                SET END OF SECOND SCAN        00147000
 176            SR    R3,R4                SCAN LENGTH                   00148000
 177            BCTR  R3,0                                               00149000
 178            EX    R3,SCAN2             SCAN FOR SECOND DELIMITER     00150000
 179            BC    6,FOUNDIT            BRANCH IF FOUND               00151000
 180            BCTR  R1,0                 LAST CHARACTER WAS DELIMITER  00152000
 181 FOUNDIT    EQU   *                                                  00153000
 182            SR    R1,R4                DESTINATION LENGTH            00154000
 183            CH    R1,=H'8'             VALID LENGTH?                 00155000
 184            BH    SETSTAT              NO                            00156000
 185            STC   R1,DESTLNG           STORE LENGTH                  00157000
 186            MVC   DEST,0(R4)           AND DESTINATION               00158000
 187 SETSTAT    EQU   *                                                  00159000
 188            MVI   PASSBOOK,C'N'        INDICATE NO PASSBOOK          00160000
 189            CLC   9(1,R5),CTBTERM+1    NORMAL ADDRESS?               00161000
 190            BE    CKEOM                YES                           00162000
```

```
 191             MVI   PASSBOOK,C'P'        INDICATE PASSBOOK PRESENT          00163000
 192 CKEOM       EQU   *                                                       00164000
 193             LA    R4,0(R6,R14)         R4 = END OF SEGMENT                00165000
 194             LR    R8,R4                                                   00166000
 195             CLI   0(R8),X'15'          ENDS WITH CARRIAGE RETURN?         00167000
 196             BNE   *+6                  NO                                 00168000
 197             BCTR  R8,0                                                    00169000
 198             BCTR  R8,0                                                    00170000
 199             CLC   0(2,R8),=C'**'       SEGMENT TO BE CANCELLED?           00171000
 200             BE    TESTMOD4             YES,  DON'T ADD STATUS INFO        00172000
 201             CLI   1(R8),COMMA          MORE SEGMENTS COMING?              00173000
 202             BE    NOTEOM               YES                                00174000
 203             CLI   1(R8),PERIOD         END-OF-MESSAGE?                    00175000
 204             BNE   ADDSTAT              NO                                 00176000
 205             OI    DECCSWST,X'01'       INDICATE END-OF-MESSAGE            00177000
 206 NOTEOM      EQU   *                                                       00178000
                 LA    R8,1(,R8)                                               00178100
 207             LR    R4,R8                R4 = END-OF-SEGMENT POINTER        00179000
 208             SR    R8,R14               RE-CALCULATE SEGMENT LENGTH        00180000
 209             LTR   R6,R8                AND TEST FOR NO-DATA SEGMENT       00181000
 210             BP    ADDSTAT              BRANCH IF DATA SEGMENT             00182000
 211 ZEROLNG     EQU   *                                                       00183000
 212             SR    R6,R6                SET ZERO LENGTH                    00184000
 213 RETURN      EQU   *                                                       00185000
 214             L     R13,4(,R13)          GET CALLER'S SAVE AREA             00186000
 215             ST    R6,28(,R13)          STORE LENGTH IN R2 OF CALLER       00187000
 216             L     R14,12(,R13)         GET RETURN ADDRESS                 00188000
 217             RETURN (0,12)              AND RETURN,  RC IN R15             00189000
 218+            LM    0,12,20(13) RESTORE THE REGISTERS
 219+            BR    14 RETURN

 221 ADDSTAT     EQU   *                                                       00191000
 222             LA    R6,L'STATUS(,R6)     ADD STATUS LENGTH TO SEG LENGTH    00192000
 223             MVC   0(L'STATUS,R4),STATUS  ADD STATUS INFO TO SEGMENT       00193000
 224             CLI   DESTLNG,0            DESTINATION LENGTH ZERO?           00194000
 225             BE    MOVESEG              YES                                00195000
 226             MVC   0(8,R5),DEST         PUT DESTINATION IN SEGMENT         00196000
 227             AH    R5,DESTL             UPDATE TEXT POINTER                00197000
 228             LA    R5,1(,R5)            INSURE 1 BLANK AFTER DESTINATION   00198000
 229 MOVESEG     EQU   *                                                       00199000
 230             BCTR  R6,0                 REDUCE LENGTH FOR MOVE             00200000
 231             EX    R6,MOVE              MOVE SEGMENT TO FRONT OF BUFFER    00201000
 232             LA    R6,1(,R6)            RE-ADJUST LENGTH                   00202000
 233             CLI   DESTLNG,0            DESTINATION LENGTH ZERO?           00203000
 234             BE    TESTMOD4             YES                                00204000
 235             LA    R6,1(,R6)            ADD 1 FOR BLANK AFTER TRAN CODE    00205000
 236             AH    R6,DESTL             ADD DEST LENGTH TO DATA LENGTH     00206000
 237 TESTMOD4    EQU   *                                                       00207000
 238             TM    CTBFEAT,CTBFMOD4     2980-4?                            00208000
 239             BZ    RETURN               NO                                 00209000
 240             BAL   R4,FINDCNT           FIND INPUTING CNT                  00210000
 241             B     RETURN               AND RETURN                         00211000

 243 RESEND      EQU   *                                                       00213000
 244             CH    R6,=H'8'             VALID MESSAGE LENGTH?              00214000
```

```
STMT   SOURCE STATEMENT                                                F15OCT70   1/14/72

 245            BNE   SETSCAN1              NO                                    00215000
 246            CLC   RESENDSQ,0(R14)       RESEND REQUEST?                       00216000
 247            BNE   SETSCAN1              NO                                    00217000
 248            BAL   R4,FINDCNT            GET CNT ADDRESS                       00218000
 249            L     R4,4(,R13)            GET CALLER'S SAVE AREA ADDRESS        00219000
 250            ST    R10,60(,R4)           STORE CNT ADDRESS IN CALLERS R10      00220000
 251            LA    R15,4                 SET RETURN CODE                       00221000
 252            B     ZEROLNG               ZERO DATA LENGTH AND RETURN           00222000

 254 FINDCNT    EQU   *                                                           00224000
 255            L     R2,SCDCNT             GET CNT BASE                          00225000
 256            LH    R10,CTBCNTP           OFFSET TO 1ST CNT ON CTB              00226000
 257            LR    R3,R10                                                      00227000
 258            AR    R3,R2                 SAVE ADDRESS OF 1ST CNT               00228000
 259 NEXTCNT    EQU   *                                                           00229000
 260            AR    R10,R2                ADDRESS OF NEXT CNT                   00230000
 261            CLC   TELLERID(1),CNTNAME   NAME MATCH TELLER ID?                 00231000
 262            BE    CNTFOUND              YES                                   00232000
 263            LH    R10,CNTCNTP           OFFSET OF NEXT CNT IN CHAIN           00233000
 264            LTR   R10,R10               LAST CNT?                             00234000
 265            BNM   NEXTCNT               NO                                    00235000
 266 CNTRET     EQU   *                                                           00236000
 267            LR    R10,R3                USE 1ST CNT IN CHAIN                  00237000
 268            BR    R4                    RETURN                                00238000

 270 CNTFOUND   EQU   *                                                           00240000
 271            CR    R10,R3                1ST CNT?                              00241000
 272            BE    CNTRET                YES                                   00242000
 273            XC    CNTCNTP-CNT(2,R3),CNTCNTP-CNT(R10) SWAP                     00243000
 274            XC    CNTCNTP-CNT(2,R10),CNTCNTP-CNT(R3)     CNT CHAIN            00244000
 275            XC    CNTCNTP-CNT(2,R3),CNTCNTP-CNT(R10)          POINTERS        00245000
 276            BR    R4                                 AND RETURN               00246000
```

```
278 ***                                                                *** 00248000
279 *                CONSTANTS,  DSECTS,  AND EQUATES                    * 00249000
280 ***                                                                *** 00250000

282 DESTL      DC    H'0'                                                  00252000
283 DEST       DS    CL8                                                   00253000
284 STATUS     DC    X'F9F90000'                                           00254000
285 TABLE1     DC    256XL1'00'                                            00255000
286            ORG   TABLE1+65                                             00256000
287            DC    X'41'                                                 00257000
288            ORG   TABLE1+89                                             00258000
289            DC    X'59'                                                 00259000
290            ORG                                                         00260000
291 TABLE2     DC    192XL1'FF',64XL1'00'                                  00261000
292 RESENDSQ   DC    C'&&RESEND '                                          00262000

294 SCAN1      TRT   0(0,R14),TABLE1                                       00264000
295 SCAN2      TRT   0(0,R4),TABLE2                                        00265000
296 MOVE       MVC   0(0,R5),0(R14)                                        00266000

298 DESTLNG    EQU   DESTL+1                                               00268000
299 PASSBOOK   EQU   STATUS+2                                              00269000
300 TELLERID   EQU   STATUS+3                                              00270000
301 PERIOD     EQU   X'4B'                                                 00271000
302 COMMA      EQU   X'6B'                                                 00272000
303 CTBFMOD4   EQU   X'02'          CTBFEAT SETTING IDENTIFING A 298C-4    00273000
304 CTBFMOD2   EQU   X'01'          CTBFEAT SETTING IDENTIFING A 2980-2    00274000
305 CTBA1SEG   EQU   X'02'          TO BE REMOVED                          00275000

307            LTORG                                                       00277000
308                  =H'10'
309                  =H'8'
310                  =C'**'
311                  =H'7'
```

```
STMT    SOURCE STATEMENT                                              F15OCT70   1/14/72

 313           ICLI   CLBBASE=0,CTBBASE=0.CNTBASE=0                              00279000
```



```
STMT    SOURCE STATEMENT                                              F15OCT70   1/14/72

 594           PRINT NOGEN                                                       00280000
 595           ISCD  SCDBASE=0                                                   00281000
 942                 *,******************************************************
 943                 *,*             IMS/360 SYSTEM
 944                 *,*      VERSION 2   RELEASE 1   MOD LEVEL 1       *
 945                 *,******************************************************
 947           END                                                               00282000
```

CHAPTER 6. IMS/360 SAMPLE PROBLEM

The IMS/360 basic distribution tape contains two data sets, IMS2.GENLIB and IMS2.LOAD. These data sets are unloaded versions of direct access partitioned data set libraries as produced by the operating system utility program IEHMOVE. Contained in these libraries are the program modules and macro-definitions which comprise the sample application.

A series of steps are involved in the creation of the sample application environment. Detailed background information regarding these steps is available from the references shown below:

• Copying the distribution libraries to direct access storage devices	SPRM * Chapter 2
• Performing a system definition and related functions	SPRM * Chapter 3
• Performing a data base description (DBDGEN)	UTRM ** Chapter 2
• Performing a program specification block generation (PSBGEN)	UTRM ** Chapter 3
• Performing an application control blocks utility execution (ACBGEN)	UTRM ** Chapter 3A
• Moving sample problem programs and control blocks	SPRM * Chapter 6
• Executing a print of the data base in a batch environment	SPRM *** Chapter 6
• Executing a data base load in the batch environment	ORM *** Chapter 6
• Initializing the system in an online environment. Executing the online application program from user terminals.	ORM *** Chapter 6

* This Manual
** IMS/360 Utilities Reference Manual
*** IMS/360 Operator's Reference Manual

Before proceeding with the instructions for setting up the sample application, a description of the application and its data bases is appropriate.

DESCRIPTION OF SAMPLE PROBLEM

The application included within the sample problem is taken from the manufacturing industry. This application in its full sense includes the creation, usage, and maintenance of the logical data bases associated with the product data systems. This product data can be contained in three subject data bases. The product data is either related to engineering drawings, part numbers, or systems equipment structure. There are three logical data bases, each organized under one of the above subjects.

To facilitate the implementation of these three logical data bases, they have been split into three data bases of five data set groups (see Figure 11).

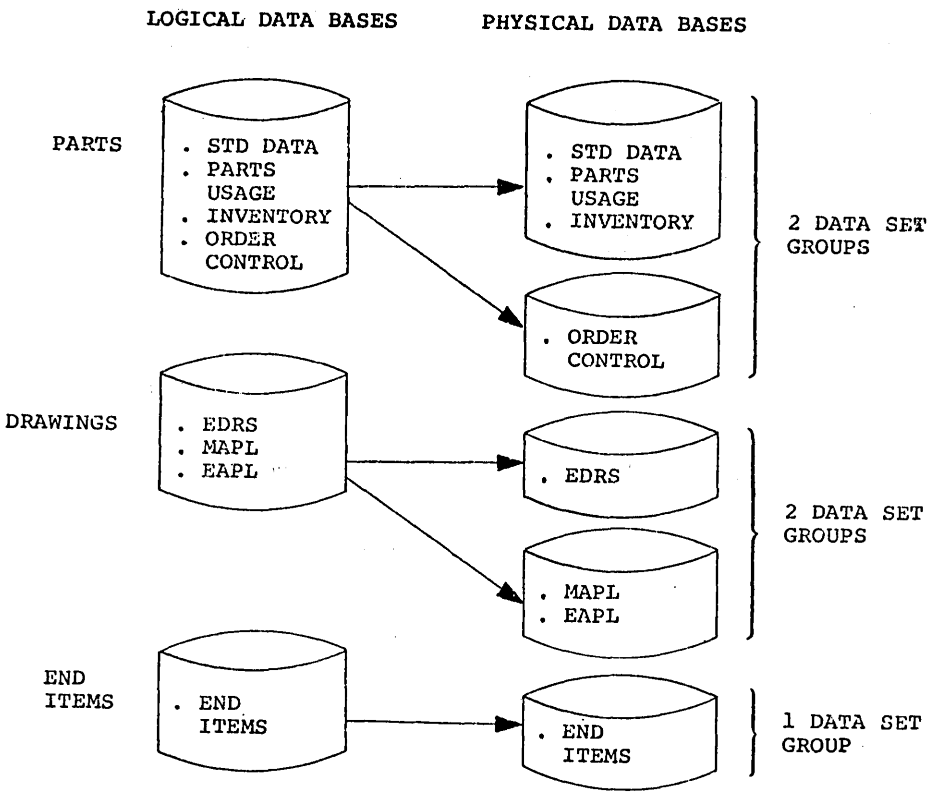


Figure 11. Logical and physical data bases

Each of the five physical data bases and the segments contained within these data bases is described in Figures 12, 13, and 14.

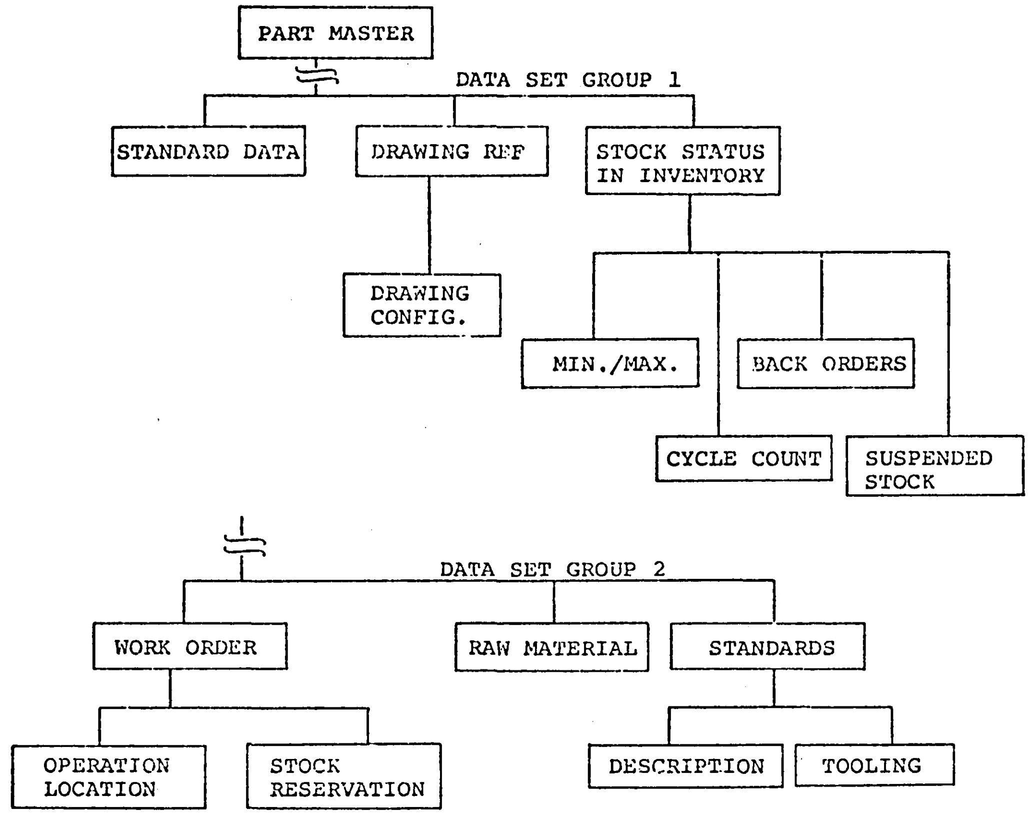


Figure 12. Part data base and segment

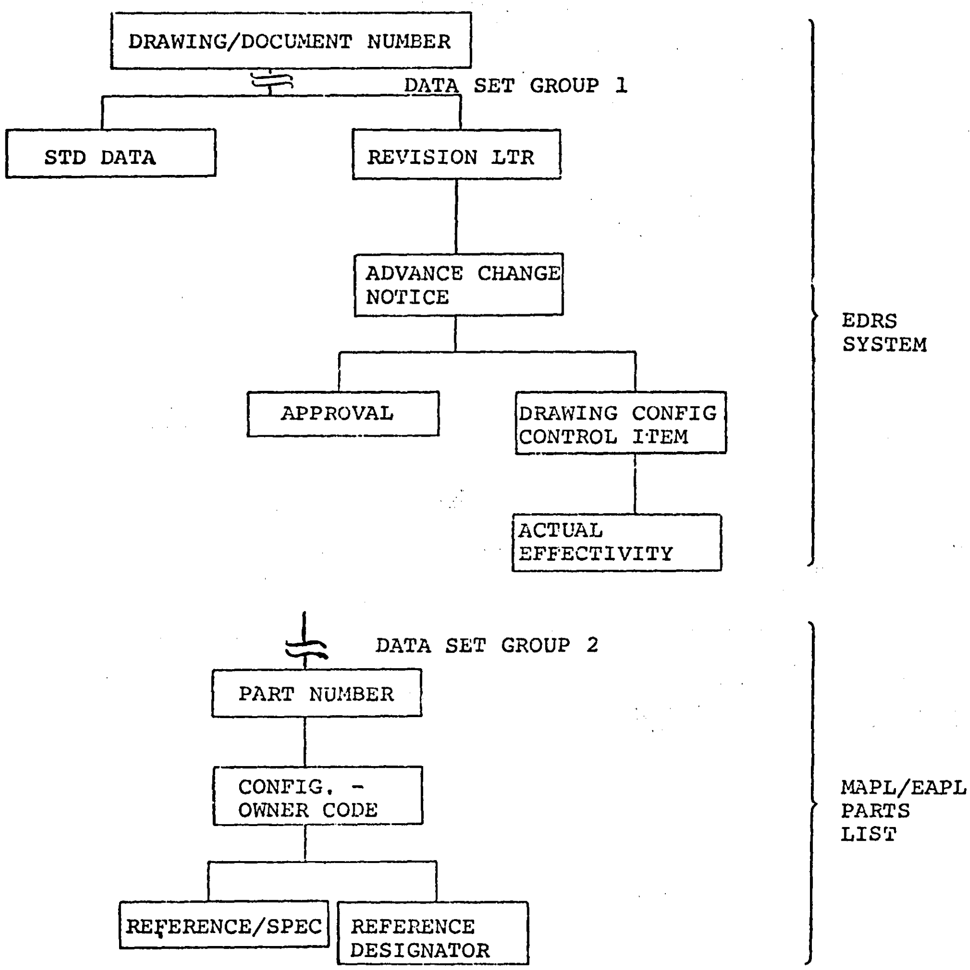


Figure 13. Drawing data base

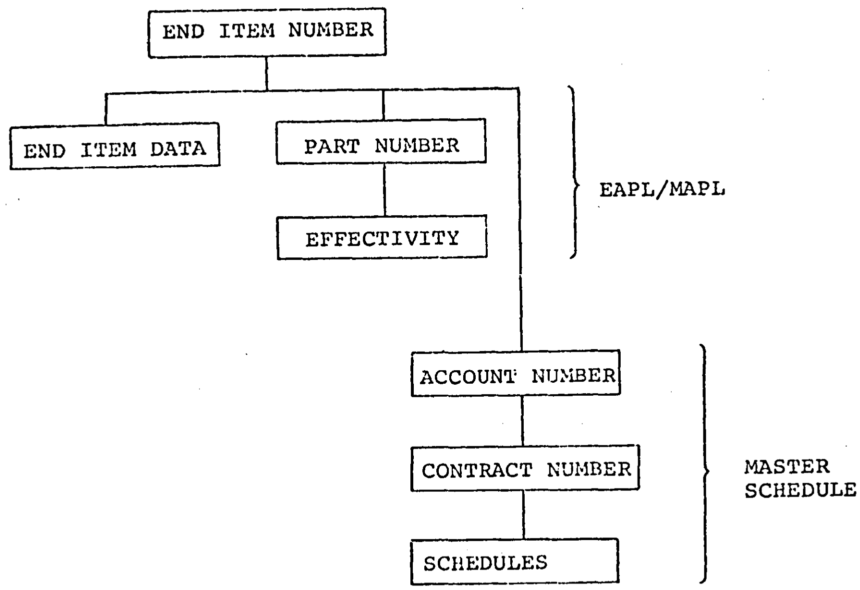


Figure 14. End item data base

The application portion of the IMS/360 sample problem includes the implementation of a small subset of this entire application. The data base structure of the application in the sample problem includes the segments and their structure described in Figure 15.

This data base subset structure includes:

- One part number description segment for each part within the data base
- A standard data segment for each part which provides additional information of a standard nature about the part
- Inventory stock status segments for each part. The application is designed with multiple inventory locations permissible and normally required for any particular part.
- Zero to n cycle count and back-order segments for each inventory location of a particular part.

In addition to the application data base substructure, the sample problem includes application programs to:

1. Create the data base substructure in an IMS/360 batch processing region. The input data for part, inventory, cycle count, back order, and standard part data to load into the data base substructure is provided.
2. Message processing programs and associated transactions to execute in an IMS/360 online control region to:

   a. Inquire about a part and its description

b. Inquire about a part's total inventory in all locations or by specific inventory location

c. Add a new part and its description

d. Add part inventory information by location to an existing part description

e. Delete part inventory information by location

f. Delete a part after deletion of all its subordinate part inventory information

g. Close a part order to increase the part inventory at a specific location

h. Disburse a specific quantity of a particular part on a planned or unplanned basis at a particular part inventory location, thereby reducing inventory.

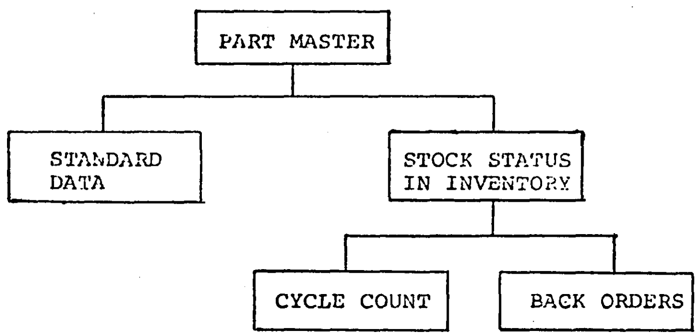


Figure 15. Sample problem application data base

The foregoing describes the application aspects of the sample problem.

Figure 16 interrelates the sample problem transactions, programs, and data bases.

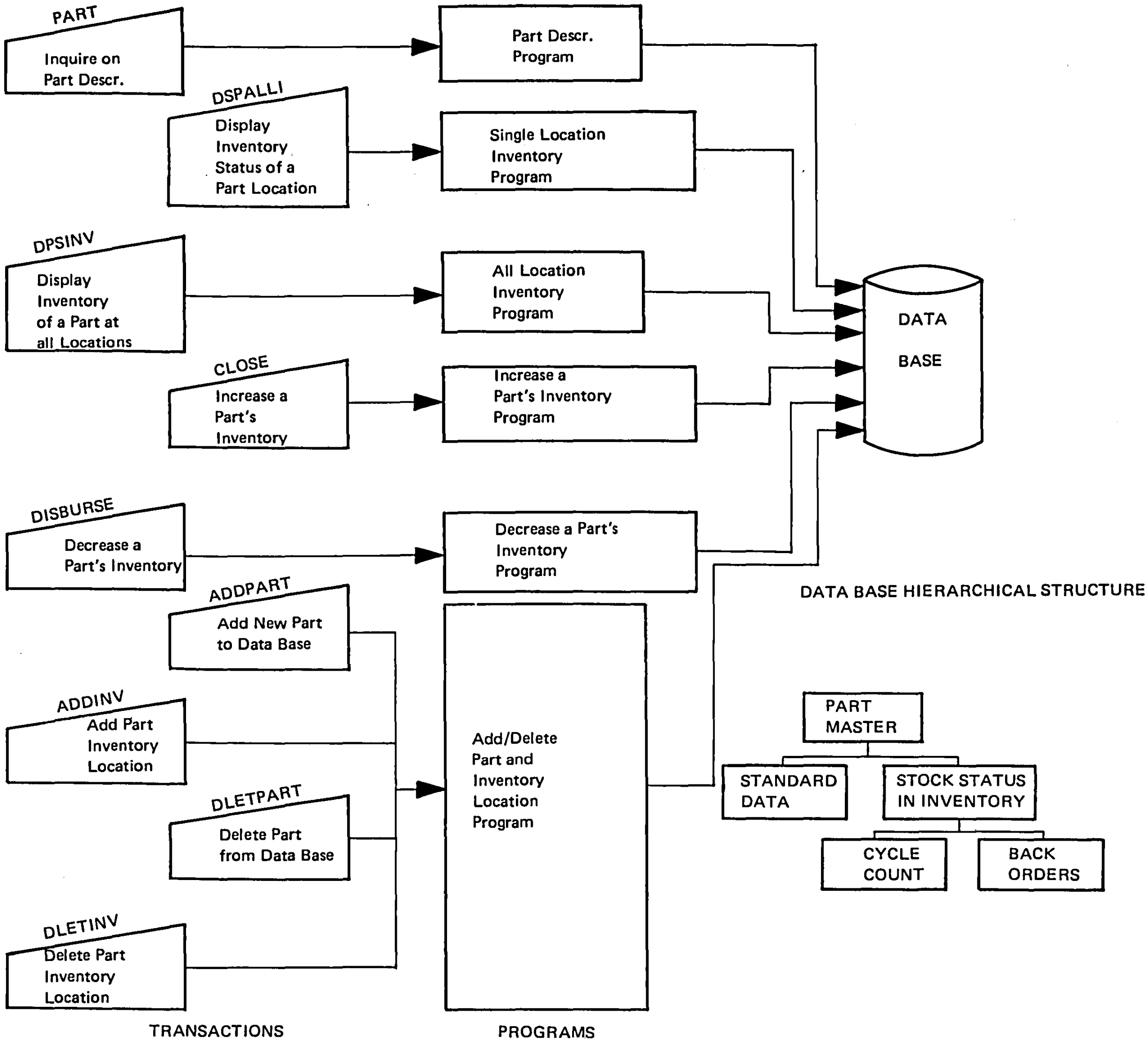


Figure 16. Sample problem transactions, programs, and data bases

CREATING SAMPLE PROBLEM ENVIRONMENT

As outlined in the introduction to this chapter, a series of steps must be performed to create the sample problem environment. The remainder of this chapter describes these in detail or provides references for the required steps.

COPYING IMS/360 DISTRIBUTION LIBRARIES

Figure 17 is an example of the JCL necessary for the allocation and cataloging of the data sets required for IMS/360 system definition and execution.

The tape move described in Chapter 2 of this manual should move the data sets into libraries created by the JCL example.

PERFORMING AN IMS/360 SYSTEM DEFINITION

Prior to performing Stages 1 and 2 of IMS/360 system definition, certain data sets must be allocated and cataloged. Figure 17 is an example of the JCL required to allocate and catalog the data sets required by the sample problem. Space requirements should be adjusted if devices other than 2311 are to be used.

If a batch-only execution is planned, the message queue data sets need not be allocated.

```
//ALLOCATE JOB 1,IMS,MSGCLASS=A,MSGLEVEL=1,PRTY=12
// EXEC PGM=IEHPROGM
//TWO  DD VOL=SER=222222,UNIT=2311,DISP=OLD
//THR  DD VOL=SER=333333,UNIT=2311,DISP=OLD
//ILIB01   DD VOL=SER=ILIB01,UNIT=2311,DISP=OLD
//ILIB02   DD VOL=SER=ILIB02,UNIT=2311,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *,DCB=BLKSIZE=80
 SCRATCH VTOC,VOL=2311=222222,PURGE
 SCRATCH VTOC,VOL=2311=333333,PURGE
 SCRATCH VTOC,VOL=2311=ILIB01,PURGE
 SCRATCH VTOC,VOL=2311,ILIB02,PURGE
//LIBRARYS EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//SIDC01 DD UNIT=2311,VOL=SER=SIDC01,DISP=OLD
//OBJDSET DD DSN=IMS2.OBJDSET,UNIT=2311,VOL=SER=222222,DISP=(,KEEP),
//           SPACE=(CYL,(1,1,1))
//CATALOG DD DSN=SYSCTLG,UNIT=2311,VOL=SER=ILIB02,DISP=(,KEEP),
//           SPACE=(TRK,(2,1))
//RESLIB DD DSN=IMS2.RESLIB,UNIT=2311,VOL=SER=ILIB02,DISP=(,KEEP),
//           SPACE=(CYL,(40,5,20)),DCB=SYS1.LINKLIB
//MACLIB DD DSN=IMS2.MACLIB,UNIT=2311,VOL=SER=ILIB01,DISP=(,KEEP),
//           SPACE=(CYL,(30,5,15)),DCB=SYS1.MACLIB
//PGMLIB DD DSN=IMS2.PGMLIB,UNIT=2311,VOL=SER=ILIB01,DISP=(,KEEP),
//           SPACE=(CYL,(10,2,10)),DCB=SYS1.LINKLIB
//PSBLIB DD DSN=IMS2.PSBLIB,UNIT=2311,VOL=SER=ILIB02,DISP=(,KEEP),
//           SPACE=(CYL,(10,2,5)),DCB=SYS1.LINKLIB
//DBDLIB DD DSN=IMS2.DBDLIB,UNIT=2311,VOL=SER=ILIB02,DISP=(,KEEP),
//           SPACE=(CYL,(10,2,5)),DCB=SYS1.LINKLIB
//IMSACB DD DSN=IMS2.ACBLIB,UNIT=2311,VOL=SER=ILIB02,DISP=(,KEEP),
//           SPACE=(CYL,(10,2,5)),DCB=SYS1.LINKLIB
//PROCLIB DD DSN=IMS2.PROCLIB,UNIT=2311,VOL=SER=ILIB02,DISP=(,KEEP),
//           SPACE=(CYL,(1,1,5)),DCB=SYS1.PROCLIB
//QBLKS DD DSN=IMS2.QBLKS,VOL=SER=ILIB01,DISP=(,KEEP),UNIT=2311,
//           SPACE=(CYL,1),DCB=DSORG=PS
//SHMSG DD DSN=IMS2.SHMSG,VOL=SER=ILIB01,DISP=(,KEEP),UNIT=2311,
//           SPACE=(CYL,5),DCB=DSORG=PS
//LGMSG DD DSN=IMS2.LGMSG,VOL=SER=ILIB02,DISP=(,KEEP),UNIT=2311,
//           SPACE=(CYL,5),DCB=DSORG=PS
//DBLLOG DD DSN=IMS2.DBLLOG,VOL=SER=SERNUM,DISP=(,KEEP),
//           UNIT=2311,DCB=DSORG=PS,SPACE=(512,50)
//SYSIN DD *,DCB=BLKSIZE=80
 RELEASE INDEX=IMS2
 DLTX INDEX=IMS2,CVOL=2311=SIDC01
 CONNECT INDEX=IMS2,CVOL=2311=SIDC01,VOL=2311=ILIB02
 CATLG DSNAME=IMS2.GENLIB,VOL=2311=222222,CVOL=2311=ILIB02
 CATLG DSNAME=IMS2.LOAD,VOL=2311=333333,CVOL=2311=ILIB02
 CATLG DSNAME=IMS2.RESLIB,VOL=2311=ILIB02,CVOL=2311=ILIB02
 CATLG DSNAME=IMS2.DBLLOG,VOL=2311=SERNUM,CVOL=2311=ILIB02
 CATLG DSNAME=IMS2.MACLIB,VOL=2311=ILIB01,CVOL=2311=ILIB02
 CATLG DSNAME=IMS2.PGMLIB,VOL=2311=ILIB01,CVOL=2311=ILIB02
 CATLG DSNAME=IMS2.PSBLIB,VOL=2311=ILIB02,CVOL=2311=ILIB02
 CATLG DSNAME=IMS2.DBDLIB,VOL=2311=ILIB02,CVOL=2311=ILIB02
 CATLG DSNAME=IMS2.ACBLIB,VOL=2311=ILIB02,CVOL=2311=ILIB02
 CATLG DSNAME=IMS2.PROCLIB,VOL=2311=ILIB01,CVOL=2311=ILIB02
 CATLG DSNAME=IMS2.QBLKS,VOL=2311=ILIB01,CVOL=2311=ILIB02
 CATLG DSNAME=IMS2.SHMSG,VOL=2311=ILIB02,CVOL=2311=ILIB02
 CATLG DSNAME=IMS2.LGMSG,VOL=2311=ILIB01,CVOL=2311=ILIB02
 CATLG DSNAME=IMS2.OBJDSET,VOL=2311=222222,CVOL=2311=ILIB02
```

Figure 17. Example of allocation and cataloging

After having completed the allocation of required data sets, Stage 1 of IMS/360 system definition is performed. Figure 18 describes the control cards needed as input to system definition for an online system. The TRANSACT, PROGRAM, and DATABASE cards describe the resources of the application sample. For the sake of simplicity, only one telecommunications line group, one line, and one physical terminal (2740) are described. Two logical terminals, the MASTER and one named HOWARD, are described. Logical terminal HOWARD is used by the application as a destination for exception messages.

If the user of the sample problem desires to perform the sample problem by means of a 2260 Display Station, it must be included in his Stage 1 system definition. The user must follow the rules of Chapter 3 of this manual for system definition and modify the Figure 18 control cards accordingly.

WARNING: IMS/360 does not support the 2260 Display Station as a master terminal. Prior to Stage 1 of IMS/360 system definition, IMS2.GENLIB and IMS2.LOAD must be cataloged. Those parameters in the system definition control cards which are underlined can be redefined by the IMS/360 user to meet the requirements of his data processing environment with no effect on the application.

If a batch-only execution is planned, a batch system definition as described in Chapter 3 of this manual may be executed.

If assistance is required in making control card changes, refer to Chapter 3 of this manual. In particular, the message queue data set DD names and the IMS/360 library names should be reviewed.

The communication line and terminal operands may be modified as required to conform to the user's system and the operating system specifications. Chapter 3 of this manual provides information on the various operands permitted.

Once Stage 2 of system definition is successfully completed, the IMS/360 user must perform the following.

1. Include the one Type 1 and the one Type 2 IMS/360 SVC interface modules in the operating system nucleus. This can be done with a relink-edit of the operating system nucleus if available user SVC numbers were generated at time of operating system generation. If available SVC numbers do not exist, the user must perform at least an operating system nucleus-only system generation to provide the required SVC numbers. In addition, a Type 4 SVC number must exist or be made available. The Type 1 and Type 2 SVC interface modules are placed into IMS2.RESLIB by system definition. The user must link edit these modules from this library. The Type 1 SVC module has the name IGCXXX, and the Type 2 SVC module, IGCYYY. (The Type 1 and 2 SVCs are required only for an online execution.)

2. Link edit the OSAM channel end appendage IGG019Z8 or the equivalent to SYS1.SVCLIB and link edit the OSAM open modules into SYS1.SVCLIB. The names of the Type 4 SVC modules are listed in Chapter 3 of this manual as specified in system definition. Both the channel-end appendage and the OSAM Type 4 SVC modules exist in IMS2.RESLIB after system definition is performed. These modules must be link-edited as RENT and RFER.

3. Allocate and catalog the four sequential data sets used for message queuing in this example. Their data definition statement names are IMSDBL, QBLKS, SHMSG, and LGMSG. The associated data set names must be IMS2.DBLLOG, IMS2.QBLKS, IMS2.SHMSG, and IMS2.LGMSG. Chapter 3 of this manual provides information on the allocation of these data sets.

```
//IMSDEF    JOB 1,IMS,MSGLEVEL=1
//STEP      EXEC    PGM=IEUASM,PARM='DECK,NOLOAD',REGION=130K
//SYSLIB    DD  DSN=IMS2.GENLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=A
//SYSPUNCH DD   SYSOUT=B
//SYSUT1    DD  UNIT=SYSDA,SPACE=(1700,(500,50))
//SYSUT2    DD  UNIT=SYSDA,SPACE=(1700,(500,50))
//SYSUT3    DD  UNIT=(SYSDA,SEP=(SYSUT1,SYSUT2)),                X
//                  SPACE=(1700,(500,50))
//SYSIN         DD  *

   IMSCTRL      SYSTEM=(MVT,ALL),MAXIO=7,MAXREGN=1
   IMSCTF       SVCNO=(243,244,245),APNDG=Z8,                   X
                CPLOG=500,CORE=(,2000)

   MSGQUEUE     DSETS=(2311),RECNG=(176,1056),                  X
                BUFFERS=(9,1056),SHUTDWN=50
   DATABASE     DBD=DI21PART
   APPLCTN      PSB=DFSSAM02
         TRANSACT  CODE=PART,PRTY=(7,10,2),INQUIRY=YES
   APPLCTN      PSB=DFSSAM03
         TRANSACT  CODE=DSPINV,PRTY=(7,10,2),INQUIRY=YES
   APPLCTN      PSB=DFSSAM04
         TRANSACT  CODE=ADDPART,PRTY=(7,10,2),INQUIRY=NO
         TRANSACT  CODE=ADDINV,PRTY=(7,10,2),INQUIRY=NO
         TRANSACT  CODE=DLETPART,PRTY=(7,10,2),INQUIRY=NO
         TRANSACT  CODE=DLETINV,PRTY=(7,10,2),INQUIRY=NO
   APPLCTN      PSB=DFSSAM05
         TRANSACT  CODE=CLOSE,PRTY=(7,10,2),INQUIRY=NO
   APPLCTN      PSB=DFSSAM06
         TRANSACT  CODE=DISBURSE,PRTY=(7,10,2),INQUIRY=NO
   APPLCTN      PSB=DFSSAM07
         TRANSACT  CODE=DSPALLI,PRTY=(7,10,2),INQUIRY=YES
   LINEGRP      DDNAME=DD2740,UNIT=2740
      LINE      ADDR=032
        TERMINAL    ADDR=E2
       NAME         (CTRL,MASTER)
        NAME        HOWARD
   IMSGEN       ASMPRT=ON,LKPRT=(XREF,LIST),OBJDSET=IMS2.OBJDSET,X
                PRTY=8
   END
```

Figure 18. Input to system definition

4. Move the IMS/360 cataloged procedures named IMS (online only) and IMSRDR from IMS2.PROCLIB to SYS1.PROCLIB. The instructions for Executing Jobs Using Procedures from IMS2.PROCLIB are in Chapter 3 of this manual.

5. Link edit the IMS/360 region control module from IMS2.RESLIB to SYS1.LINKLIB. The name of this module is DFSRRC00 and it should be marked (RENT, RFER) in the link edit.

6. Change the operating system privileged name list module in SYS1.LINKLIB to include DFSRRC00. How to perform this operation is described in Chapter 3 of this manual. (This is required only for an online execution.)

A narrative is provided at the end of the output listing from IMS/360 system definition Stage 1. This narrative describes the functions a system user must perform prior to execution of his IMS/360 system. <u>Please read this narrative</u>.

PERFORMING A DATA BASE DESCRIPTION (DBDGEN) GENERATION

Part of the sample problem is the generation of a data base description which is used by the sample application. The generation process consists of an assembly and linkage edit. A member in IMS2.GENLIB titled, DI21PART, contains the source input to generation of a DBD. A procedure is placed in IMS2.PROCLIB by system definition to perform DBD generation. The following JCL should be used to invoke this procedure and use the DBD source input to create the DBD. The output of the DBD generation becomes a member in the partitioned data set IMS2.DBDLIB.

```
//DBD       JOB     SAMPLE,MSGLEVEL=1
//              EXEC    DBDGEN,MBR=DI21PART
//C.SYSIN   DD      DSNAME=IMS2.GENLIB(DI21PART),DISP=SHR
```

The operating system reader used to read this JCL must have access to IMS2.PROCLIB. The suggested START command is START IMSRDR, XXX, where XXX is the card reader unit address.

PERFORMING A PROGRAM SPECIFICATION BLOCK GENERATION (PSBGEN)

An optional part of the sample problem involves generation of a program specification block (PSB). The generation process is called PSBGEN. Like DBDGEN, the process consists of an assembly and linkage edit. A member of IMS2.GENLIB, named DFSSAP04, contains the source input which will generate the PSB for the online application program named DFSSAM04. IMS/360 system definition placed a procedure named PSBGEN in IMS2.PROCLIB. The following JCL uses this procedure to place the output PSB in the partitioned data set IMS2.PSBLIB as defined in the PSBLIB statement of system definition.

```
//PSB       JOB     1,IMS,MSGLEVEL=1
//STEP      EXEC    PSBGEN,MBR=DFSSAM04
//C.SYSIN   DD      DSN=IMS2.GENLIB(DFSSAP04),DISP=SHR
```

Note: The input member DFSSAP04 creates an output PSB named DFSSAM04.

The OS reader used to read this JCL must have access to IMS2.PROCLIB. The suggested START command is START IMSRDR, XXX, where XXX is the card reader unit address.

MOVING SAMPLE PROBLEM PROGRAMS AND CONTROL BLOCKS

The next step in the sample problem is to have the IMS/360 user relink edit the remaining PSBs and programs for data base creation and message processing from IMS2.LOAD into their respective IMS/360 libraries (that is, IMS2.PSBLIB and IMS2.PGMLIB).

A load module exists within IMS2.LOAD for each PSB and application program. The following JCL and link edit control statement are used to copy the PSBs from IMS2.LOAD to IMS2.PSBLIB.

```
//PSBMOVE  JOB 1,IMS,MSGLEVEL=1
//         EXEC    PGM=IEWL,REGION=110K,                          X
//         PARM='XREF,LIST,LET,NCAL,SIZE=(100K,7248)'
//SYSLIB   DD  DSNAME=SYS1.COBLIB,DISP=SHR
//SYSLIN   DD  DDNAME=SYSIN
//SYSLMOD  DD  DSNAME=IMS2.PSBLIB,DISP=OLD
//SYSPRINT DD  SYSOUT=I
//SYSOBJ   DD  DSNAME=IMS2.LOAD,DISP=SHR
//SYSUT1   DD  UNIT=2311,DISP=(NEW,DELETE),                       X
```

```
//                   SPACE=(CYL,(10,1),RLSE)
//SYSIN         DD  *
   INCLUDE      SYSOBJ(DFSSAM11)
   NAME    DFSSAM01(R)
   INCLUDE      SYSOBJ(DFSSAM12)
   NAME    DFSSAM02(R)
   INCLUDE      SYSOBJ(DFSSAM13)
   NAME    DFSSAM03(R)
   INCLUDE      SYSOBJ(DFSSAM15)
   NAME    DFSSAM05(R)
   INCLUDE      SYSOBJ(DFSSAM16)
   NAME    DFSSAM06(R)
   INCLUDE      SYSOBJ(DFSSAM17)
   NAME    DFSSAM07(R)
   INCLUDE      SYSOBJ(DFSSAM18)
   NAME    DFSSAM08(R)
```

If a batch-only execution is desired, only DFSSAM11 and DFSSAM18 are required.

The parameters underlined in the JCL statements should be modified to conform to the user's system configuration.

The following JCL and control card statements are used to relink edit the application program load modules from IMS2.LOAD to IMS2.PGMLIB.

```
//PGMMOVE   JOB 1,IMS,MSGLEVEL=1
//      EXEC     PGM=IEWL,REGION=110K,                              X
//          PARM='XREF,LIST,LET,NCAL,SIZE=(100K,7248)'
//SYSLIB    DD  DSNAME=SYS1.COBLIB,DISP=SHR
//SYSLIN    DD  DDNAME=SYSIN
//SYSLMOD   DD  DSNAME=IMS2.PGMLIB,DISP=OLD
//SYSPRINT DD   SYSOUT=I
//SYSOBJ    DD  DSNAME=IMS2.LOAD,DISP=SHR
//SYSUT1    DD  UNIT=2311,DISP=(NEW,DELETE),                       X
//              SPACE=(CYL,(10,1),RLSE)
//SYSIN         DD  *
   INCLUDE      SYSOBJ(DFSSAM01)
                ENTRY DLITCBL
   NAME    DFSSAM01(R)
   INCLUDE      SYSOBJ(DFSSAM02)
                ENTRY DLITCBL
   NAME    DFSSAM02(R)
   INCLUDE      SYSOBJ(DFSSAM03)
                ENTRY DLITCBL
   NAME    DFSSAM03(R)
   INCLUDE      SYSOBJ(DFSSAM04)
                ENTRY DLITCBL
   NAME    DFSSAM04(R)
   INCLUDE      SYSOBJ(DFSSAM05)
                ENTRY DLITCBL
   NAME    DFSSAM05(R)
   INCLUDE      SYSOBJ(DFSSAM06)
                ENTRY DLITCBL
   NAME    DFSSAM06(R)
   INCLUDE      SYSOBJ(DFSSAM07)
                ENTRY DLITCBL
   NAME    DFSSAM07(R)
   INCLUDE      SYSOBJ(DFSSAM08)
                ENTRY DLITCBL
   NAME    DFSSAM08(R)
```

If a batch-only execution is desired, only DFSSAM01 and DFSSAM08 are required.

EXECUTION OF THE APPLICATION CONTROL BLOCKS UTILITY PROGRAM

After the sample PSBs and DBDs are stored in their respective libraries, the IMS2.ACBLIB data set must be created. This is done using the Application Control Blocks utility program. A procedure is placed in IMS2.PROCLIB at system definition to accomplish this. The following JCL will invoke this procedure. The output from this execution is placed in the IMS2.ACBLIB data set.

```
//BLKS      JOB      1,IMS,MSGLEVEL
//STEP      EXEC     ACBGEN
//G.SYSIN   DD *
            BUILD    PSB=ALL
```

The OS reader used to read this JCL must have access to IMS2.PROCLIB. The suggested START command is "START IMSRDR,XXX", where XXX is the card reader unit address.

EXECUTING AN IMS/360 DATA BASE LOAD IN A BATCH ENVIRONMENT

IMS/360 system definition has placed into IMS2.PROCLIB a procedure to execute the data base load. The input data for the data base load execution, which contains the SYSIN for load, is a member of IMS2.GENLIB. The name of the member is MFDFSYSN. This procedure contains DD statements for the prime and OSAM data sets, the names of which are IMS2.DI21PART and IMS2.DI21PARO respectively. The following JCL statements will invoke the procedure to create the data base.

```
//DBLOAD   JOB 1,IMS,MSGLEVEL=1
//STEP EXEC MFDBLOAD,PSER=333333,PUNIT=2311,OSER=222222,OUNIT=2311
```

The symbolic parameters designate the volume serial and unit for the prime and OSAM data sets.

The data base must be scratched if a second execution of the MFDBLOAD procedure is desired, since this procedure allocates the data base.

A message is printed on the operating system console when the data base load is started and another when the load is completed.

PRINTING THE SAMPLE DATA BASE IN A BATCH ENVIRONMENT

IMS/360 system definition has placed into IMS2.PROCLIB a procedure to print the sample problem data base. This procedure contains the DD cards necessary to read and print the data base. The following JCL statements will invoke the procedure to print the data base.

```
   //DBPRINT   JOB 1,MSGLEVEL=1
   //STEP  EXEC    MFDBDUMP
```

INITIALIZING IMS/360 IN AN ONLINE ENVIRONMENT

At this point, the IMS/360 system has been defined for the user's environment, the application sample DBD has been created, the PSBs and programs have been created, the PSBs and programs have been relink-edited to their respective libraries, and the data base has been built.

We are now ready to execute the IMS/360 telecommunications region control program and perform message processing in an IMS/360 message region.

The system user should review the IMS/360 Operator's Reference Manual (SH20-0913) for information concerning IMS/360 cold start. The procedure named IMS, which is described in this manual, should be used to start the IMS/360 control program.

After the IMS/360 control program region has been initiated, a message is printed on both the operating system system console and the IMS/360 master terminal indicating IMS READY.

At this point, the master terminal operator should enter the restart command message

/NRESTART CHKPT 0 FORMAT ALL

The FORMAT ALL parameter will cause the IMS/360 message queues to be formatted. Formatting is only required at the initial cold start or after an I/O error occurs in the queue data sets. Formatting requires about .75 seconds per 2311 cylinder and 1.5 seconds per 2314 cylinder. These times are approximately doubled if write-checking is included. Immediately upon entry of the cold start command, the IMS/360 system responds with a message

*NRESTART IN PROGRESS

After completion of the restart, which includes opening the message log and message queue data sets and formatting the message queue data sets, the following message is generated:

*IMS COLD START COMPLETE, ENTER START COMMANDS

The system, through the operating system console, will request the mounting of a standard label, nine-track tape for the system log during cold start.

Although the IMS/360 control program is now available for message entry, no message region exists for message processing. This may be accomplished by means of the /START REGION command entered from the master terminal.

The /START REGION command causes an operating system reader, which will read the JCL packet for a message region into the Operating System/360 job queue, to be started. The JCL packet for the message region is obtained from the IMS2.PROCLIB library. Once the message region has been started and communicated with the IMS/360 online control region, a message, IMS MESSAGE REGION STARTED, is transmitted to the master terminal. Message processing may now begin.

EXECUTING ONLINE APPLICATIONS FROM USER TERMINALS

At this point, each transaction code is discussed. Both input and output information and format are included in the discussion. Figure 19, at the end of this discussion, provides a list of some part number records placed into the data base at time of data base load. Those part numbers may be used by the system user to enter transactions. The general format of all transactions is transaction code, blank, and each transaction operand separated only by a comma from the next operand. No blanks should appear between one operand and another. Most of the transaction codes have been defined as multiple line transactions and will require an EOT or equivalent to complete input.

The first transaction, PART, allows the terminal operator to inquire into the part number data base for information from the part master and standard information segments of a particular part number. The input format is:

transaction_code	part_number
part	an960C10

The output or response format is:

part_number	description	procurement_code

```
PART=AN960C10            DESC=WASHER          PROC CODE=74

INV CODE=2     MAKE DEPT=12-00     PLAN REV NUM=    MAKE TIME= 63   COMM CODE=14
```

The second transaction, DSPALLI, allows the terminal operator to display all inventory, cycle count, and back order information for a particular part. The input format is:

transaction_code	part_number
dspalli	an960c10

The output format is:

part_number	description	procurement_code
PART=AN960C10	; DESC=WASHER	; PROC CODE=74

followed by inventory description and detail information

	AREA	INV DEPT	PROJ CD	DIV	UNIT PRICE	CURRENT REQMTS	ON ORDER	IN STOCK	TOTAL DISBURSE	COUNT TAKEN	BACK ORDR
1.		AA	165	11		146	20	126	104	N	0
2.		AK	287	7F		88	0	88	37	N	0
3.	2	80	091	26		630	0	680	1157	N	0

The third transaction, DSPINV, allows the terminal operator to display inventory information at a particular inventory location. Assume it is wished to display only the third inventory entry listed in the above output. Inventory location key is obtained by concatenating AREA, INVDEPT, PROJCD, and DIV.

The input format for this transaction is:

transaction code	part number	inventory item
dspinv	an960c10,	28009126

The resultant output is:

```
PART=AN960C10      ; DESC=WASHER       ; PROC CODE=74
AREA=2; INV DEPT=80; PRJ=091; DIV=26; PRICE=   .000; STK CT DATE=513; UNIT=EACH
CURR REQMTS=   630 ; ON ORDER=     0 ; TOTAL STOCK=    680
DISB PLANNED=  1053 ; DISB UNPLANNED=      104 ; STK CT VARIANCE=      0
```

The fourth transaction, ADDPART, allows the terminal operator to add a new part into the data base with its associated description.

The input format is:

transaction code	part number	description	proc.code
addpart	ab960c10,	rivet,	74

The resultant terminal output is:

```
PART NUMBER AB960C10    ADDED TO DATA BASE
```

The fifth transaction, ADDINV, allows the terminal operator to add inventory information to an existing part in the data base.

The input format is:

transaction code	part number	inventory key
addinv	ab960c10,	8009126

The resultant output is:

```
INVENTORY 8009126   ADDED TO PART NUMBER AB960C10
```

If we wished to display the part's inventory information, we could enter

DSPINV ab960c10,8009126

The resultant output would be:

```
PART=AB960C10      ; DESC=RIVET             ; PROC CODE=74
AREA=8; INV DEPT=00; PRJ=912; DIV=6 ; PRICE=   .000; STK CT DATE=  ; UNIT=
CURR REQMTS=   0 ; ON ORDER=   0 ; TOTAL STOCK=        0
DISB PLANNED=      0 ; DISB UNPLANNED=     0 ; STK CT VARIANCE=    0
```

The sixth transaction code, DLETINV, allows the terminal operator to delete a specific inventory item for a specific part. The input format is:

transaction code	part number	inventory key
dletinv	ab960c10,	8009126

The resultant output is:

```
INVENTORY 8009126   DELETED FROM PART NUMBER AB960C10
```

If all the inventory items are deleted, then a particular part number may be deleted from the data base with the transaction code DLETPART.

The input format is:

transaction code	part number
dletpart	ab960c10

The resultant output is:

```
PART NUMBER AB960C10    DELETED FROM DATA BASE
```

The terminal operator may now wish to close an open order for a specific part in a specific inventory item. The transaction to close an open order is CLOSE. The input format is:

transaction code	part number	inventory key	quantity received
close	an960c10,	28009126,	15, 15

The resultant output is:

```
UPDATE  COMPLETE
```

The terminal operator may now wish to display inventory item 28009126 for part AN960C10. The input format is:

transaction code	part number	inventory key
dspinv	an960c10,	28009126

The resultant output is:

```
PART=AN960C10          ; DESC=WASHER          ; PROC CODE=74

AREA=2; INV DEPT=80; PRJ=091; DIV=26; PRICE=   .000; STK CT DATE=513; UNIT=EACH

CURR REQMTS=   630 ; ON ORDER=     15-; TOTAL STOCK=   695

DISB PLANNED=  1053 ; DISB UNPLANNED=      104 ; STK CT VARIANCE=      0
```

Notice that the on-order quantity has been reduced by 15 and the total stock quantity has been increased by 15 to 695 from the earlier display of this inventory information.

The final transaction code, DISBURSE, allows the terminal user to allocate a quantity on a planned or unplanned basis of a given part from a given inventory item. The input format is:

transaction code	part number	inventory key	disbursement planned or unplanned	quantity disbursed
DISBURSE	an960c10,	28009126,	u,	10

The resultant output is:

```
UPDATE COMPLETED
```

If the terminal operator now wishes to display the inventory information for key 28009126 and part number AN960C10, the input would be:

```
Transaction Part        Inventory
Code        Number      Key

dspinv      an960c10,   28009126
```

The resultant output is:

```
PART=AN960C10          ; DESC=WASHER          ; PROC CODE=74

AREA=2; INV DEPT=80; PRJ=091; DIV=26; PRICE=   .000; STK CT DATE=513;UNIT=EACH

CURR REQMTS=   630 ; ON ORDER=     15-; TOTAL STOCK=   685

DISB PLANNED=  1053 ; DISB UNPLANNED=     114 ; STK CT VARIANCE=      0
```

The user may now terminate the IMS/360 system with a checkpoint command such as described below.

Terminal input:

```
/checkpoint purge
```

Resultant output:

```
CHECKPOINT COMMAND IN PROGRESS
```

```
*CHKPT 99365/132102**IMSDBS**PURGE**
```

The following is a list of available part records in the data base which the user may employ for message processing. Those parts marked with an asterisk have dependent back-order segments. All parts have at least one dependent inventory status segment.

Part Numbers	Back Order Segments
AN960C10	
3003806	*
3007228	
3013412	
652799	
7438995P002	
7618032P101	*
922399-001	
82125-869	

Figure 19. Part number records

A complete listing of the part numbers available on the data base may be obtained by executing the procedure MFDBDUMP as follows:

```
//DBDUMP    JOB 1,  IMS,MSGLEVEL=1
//STEP      EXEC    MFDBDUMP
```

APPENDIX A. DATA LANGUAGE/I TEST PROGRAM: DFSDDLTO

The Data Language/I (DL/I) test program is basically an IMS/360 application program that issues calls to DL/I based upon control card information. It also has the facility to compare the results of those calls with expected results which are also provided in control cards. Its primary purpose is to provide a facility for testing of DL/I by the issuance of calls based on control card information and, optionally, to compare the results of those calls to anticipated results.

To a limited extent, this test program may also be used as a general purpose data base utility program. However, the control card language does not lend itself well to executing large volumes of calls. It is useful as a debugging aid, since it has the facility to display DL/I control blocks, and provides an easy method of executing any call against any data base.

GENERAL DESCRIPTION

This test program is intended to be a control card processor. There are four different types of control cards:

- Status cards - establish print options and select processing PCB
- Comments cards - conditionally or unconditionally print comments
- Call cards - format the desired DL/I call
- Compare cards - compare anticipated results with actual results

The status card is used to establish print options and to select which PCB within a PSB will be used. The call to be issued is provided in the call card. A compare card is optional and is used to tell the program what the results of this call should be in the data base PCB and in the user input/output area. Various print and display options are available; these are based on whether the results of the call agree with the data in the compare card. Comments cards are also optional. As the name implies, they are only comments and can be used by the programmer at his discretion. As will be seen later, there are two types of comments: conditional and unconditional.

The general sequence of operation is to read call cards until a noncontinued call card is detected. The DL/I call is issued based on data in the call card(s). The program then reads the next control card. If a compare card is read, it compares the contents of the compare card with the corresponding field in the PCB, or, if a data compare card, with the data in the user input/output area. The comments, call, compare, PCB, input/output area, and compare data are printed if requested. If any control card other than a compare card is read after a call was issued, the results of the prior call are printed first and the new control card is then processed.

JOB	This statement initiates the job.
EXEC	This statement specifies the program name or may invoke a cataloged procedure. The required format is: PGM=DFSRRC00,PARM='AAA,DFSDDLT0,BBBBBBBB, CCCCCCCC,DDDDDDDD' where AAA is the region type and BBBBBBBB is the name of the PSB to be used. Parameters CCCCCCC and DDDDDDD are optional and can be used to specify symbolic input terminal and output terminal names respectively.
IMS DD	This statement is comprised of two concatenated data sets. The first DD statement normally defines the library containing the PSB to be used by the test program. The second DD statement defines the library containing the DBD of the data base to be processed.
database DD	This DD statement(s) references the data base to be processed. The DDNAME must agree with the DDNAME specified in the DBD.
IEFRDER DD	This statement defines the log data set, if one is desired. A DD DUMMY statement may be used if a log is not required.
PRINTDD DD	This DD statement defines the output data set for the test program, including displays of control blocks using the SNAP macro.
SYSUDUMP DD	This statement is optional and is used by the test program only when normal termination is unavailable.
SYSIN DD	This statement defines the control card input data set.

CONTROL CARD FORMATS

In the card formats below, the $ indicates those fields which are normally filled in; the absence of the $ indicates that normally the field can be left blank and the default value will be accepted. If COL 1 is left blank on any control card, the card type defaults to the prior card type.

STATUS CARD FORMAT - establish print options and PCB selection

```
$COL 01        = S identifies this as a status card
 COL 02        = output device option
                 blank -  use DD card PRINTDD when in a DLI region;
                          use I/O PCB in the MSG region.
                 1     -  use DD card PRINTDD in MSG region if the DD
                          card is provided; otherwise, use I/O PCB.
                 A     -  same as if 1, and disregard all other fields
                          in this status card.
 COL 03        = print comment option
                 blank -  do not print
                 1     -  print always
                 2     -  print only if compare done and unequal
 COL 04        = not used
$COL 05        = print call option
                 blank - do not print
                 1     - print always
                 2     - print only if compare done and unequal
 COL 06        = not used
 COL 07        = print compare option
                 blank -  do not print
                 1     -  print always
                 2     -  print only if compare done and unequal
 COL 08        = blank
$COL 09        = print PCB option
                 blank -  do not print
                 1     -  print always
                 2     -  print only if compare done and unequal
 COL 10        = not used
$COL 11        = print segment option
                 blank -  do not print
                 1     -  print always
                 2     -  print only if compare done and unequal
COL 12 thru 15 = reserved
$COL 16 thru 23=DBD Name
                This must be one of the Data Base PCBs in
                the PSB.  This determines which PCB in the PSB to
                issue subsequent calls against.  If blank, the
                current PCB will be used.  The default PCB is
                the first Data Base PCB in the PSB.
                If DBD name is 7 blanks and non blank, non blank
                is number of which relative data base PCB in the
                PSB to use.
$COL 24        =print status option
                1      -  do not use print options in this card
                2      -  do not print this status card
                3      -  do not print this status card or use print option
                blank  -  use print options and print this card.
```

```
COL 25 thru 28 =PCB processing option - This is optional and
                is only used when two PCBs have the same DBD name
                but different processing options.  If non blank, it
                is used in addition to the DBD name in COL 16
                thru 23 to select which PCB in the PSB to use.
COL 29 thru 80 =not used
```

If no status card is read, default PCB is first data base PCB in the PSB and print options are 2: New status cards may occur at any point in the SYSIN stream, changing either the data base to be referenced or options.

COMMENTS CARD FORMAT - unconditional or conditional

```
Unconditional Comments:
  $COL 01         = U identifies this as an unconditional Comments Card
   COL 02 thru 80 is comments - any number of unconditional commments
                  are allowed; they are printed when read.  Time and date
                  are printed with each unconditional Comments card.
```

```
Conditional Comments
   COL 01         = T identifies this as a conditional Comments card
   COl 02 thru 80 is comments - up to 5 conditional Comments cards per
                  call are allowed; no continuation in COL 72 is required.
                  Printing is conditioned on the status card.  Printing
                  is deferred until after the following call and optional
                  compare are executed but prints prior to the printing of
                  the following call.
```

CALL CARD FORMAT - establish IMS/360 call

```
$COL 01          = L identifies this as a call card
 COL 03          = SSA level (optional) - see below
 COL 04          = Format options
                   U if from COL 16 is unformatted; no blanks
                   separating fields
                   blank for formatted calls with intervening
                   blanks in columns 24, 34, and 37
 COL 05 thru 08= Number of times to repeat this call (optional) in the
                   range of 0001 thru 9999
$COL 10 thru 13= DL/I application program call function
$COL 16 thru 23= SSA segment name
 COL 24          = not used
$COL 25          = ( if segment is qualified
$COL 26 thru 33= SSA field name
 COL 34          = not used
$COL 35 thru 36= DL/I call operator or operators
 COL 37          = not used
$COL 38 thru XX= Field value
$COL XX + 1      = ) end character
$COL 72          = Nonblank if more SSAs
```

This program does no checking for errors in the call; invalid functions, segments, fields, operators, or field lengths will not be detected by this program.

COL 03, the SSA level is normally blank, in which case the first call card fills SSA 1 and each following call card fills the next lower SSA. If the SSA level, COL 3, is nonblank, the card fills the SSA at that level and the following call card fills the next lower SSA.

COL 04 can contain a U to indicate an alternative format for the call card, in which case COL 16 on is the exact SSA with no intervening blanks in columns 24, 34 and 37.

COL 05 thru 08 are normally blank, but, if filled, must be right-justified with leading zeros. The identical call will be repeated equal to the value in 5-8.

COL 10 thru 13 - the DL/I call function is only required on the first SSA of the call.

COL 16 thru 23 - the segment name is not filled for unqualified calls.

If multiple SSAs in the call, put nonblank in COL 72 of prior call card and put next SSA in next card using same format; COL 1 and COL 10-13 are not required.

If field value extends past 71, put nonblank in 72 and 'CONT' in 10 through 13 of next card and continue field value starting in COL 16. Maximum field value is 256 bytes.

Maximum number of levels for this program is IMS/360 limit of 15.

On insert or replace calls, data must follow last (noncontinued) call card with an L in COL 1, 'DATA' in COL 10 thru 13, and the segment data in 16 thru 71. Data may be continued with a nonblank in 72 and data starting in COL 16 of the next card. Maximum length of segments is 1500 bytes. Note: On insert calls, last SSA should have segment name only with no qualification and may not be continued.

Since this program is unaware of segment lengths, the length of the segment displayed on REPL or ISRT calls is the number of data cards read times 56 (the data portion of the call card, continued to a maximum of 27 cards). IMS/360 knows the segment characteristics and will use the proper length.

COMPARE CARD FORMAT (PCB COMPARE)

COL 01	=	E identifies this as a compare card
COL 02	=	H indicates hold compare card (see below for details) blank indicates a reset of the hold condition or a single compare card
COL 03	=	Option requested if compare results are unequal 1 request print of I/O buffers 2 request SNAP of entire region 4 request SNAP of DLI blocks 8 abort this step; go to end of job. Multiple functions of the first three options may be obtained by summing their respective numeric values. For example, a value of 5 is a request for a print of the I/O buffers and the DL/I blocks. S SNAP subpools
COL 04	=	not used
COL 05 thru 06	=	Segment level
COL 07	=	not used
COL 08 thru 09	=	Status code XX - do not check status code OK - allow blank, GA, or GK
COL 10	=	not used
COL 11 thru 18	=	Segment name
COL 19	=	not used
COL 20 thru 22	=	Length of feedback key
COL 23	=	not used
COL 24 thru XX	=	Concatenated key feedback
COL 72	=	Nonblank to continue key feedback

The compare card is optional and is normally used to do regression testing of known data bases or to call for print of blocks or buffer pool.

Any fields left blank are not compared to the corresponding field in the PCB. Since a blank is a valid status code, to not compare status code put XX in COL 08 and 09. To accept any valid status code, (that is blank, GA, or GK), make status code in COL 8 and 9 = C'OK'.

To execute the same compare after each call, put an H in COL 2. This is useful when loading a data base to compare to a blank status code only. Since the compare was done, the current control card type is E in COL 1; the next control card read must therefore have its type in COL 1 or it will default to E. The hold-compare card will stay in effect until another compare card is read. If a new compare card is read, two compares will be done for the preceding call, since the hold compare and optional printing are done prior to reading the new compare card.

COMPARE CARD FORMAT (USER I/O AREA COMPARE)

COL 01	=	E identifies this as a compare card
COL 05 thru 08	=	Length of data in card(s) to be compared (optional) to a maximum of 27 cards
COL 10 thru 14	=	The character string 'DATA'
COL 16 thru XX	=	Data to be compared with segment in user input/output area
COL 72	=	Nonblank to continue data in next card

This compare card is optional. Its purpose is to compare the segment returned by IMS/360 to the data in this card to verify that the correct segment was retrieved.

The length in COL 5 thru 8 is optional; if present, this length will be used in the compare and in the display. If no length is specified, the shorter of either the length of data moved to the I/O area by IMS/360, or the number of data cards read times 56 is used for the length of the compare and display.

If both a compare data and a compare PCB card are present, the compare data card must precede the compare PCB card.

The conditions for printing the compare data card are the same as for printing compare PCB card; COL 7 of the status card is used. The same unequal switch is set for either the compare data or PCB. However, if control block displays are requested for unequal compares, a compare PCB card is required to request these options.

SPECIAL CONTROL CARD FORMATS

PUNCH Card

The PUNCH control card provides the facility for this program to produce an output data set consisting of the PCB compare cards, the user I/O area compare cards, all other control cards read, or any combination of the above. An example of the use of this facility would be to code only the call but not the compare cards for a new test. Then, after verifying that the calls were executed as anticipated, another run would be made where the punch control card is used to cause the test program to merge the proper compare cards, based on the results of the call, with the call cards read, producing a new output data set which would then be used as input for subsequent regression tests. If segments in an existing data base were changed, the use of this control card could cause a new test data set to be produced with the proper compare cards, rather than having to manually change the compare cards because of a change in the segments of the test data base.

The PCB compare cards are produced based on the information in the PCB after the call is completed. The data compare cards are produced based on the data in the I/O area after the call is completed. All input control cards, other than compare cards, may also be produced to provide a new composite test with the new compare cards properly merged in. The data set produced may also be sequenced.

Since the key feedback area of the PCB compare card could be rather long, two options are provided for producing these compare cards. Either the complete key feedback can be provided or the portion of the key feedback which will not fit in one card may be dropped. Forty eight bytes of key feedback fit in the first card.

Producing the full data in the I/O area into the data compare card might also be excessive and here the option again is to put it all in the data compare cards or put only the first 56 bytes in the first card and drop the rest. The test program will only compare the first 56 bytes if it only receives one data compare card.

PUNCH CARD FORMAT - produce compare cards

```
$COL 01 thru 03  =  CTL - identifies this card type.
$COL 10 thru 14  =  PUNCH - further identifies this card type as
                    controlling the punch output data set

$COL 16          =  starts keyword parameters controlling the
                    various options.  These keywords are -
                    PCBL -  Produce the full PCB compare card.
                    PCBS -  Produce the PCB compare, dropping the
                            key feedback if it exceeds one card.
                    DATAL - Produce the complete data compare cards.
                    DATAS - Produce only one card of compare data.
                    OTHER - Reproduce all control cards except
                            for compare control cards.
                    START - starting sequence number to be punched
                            in COL 73 through 80.  Eight numeric
                            characters must follow the START=
                            parameter; leading and/or trailing
                            zeros are required.
                    INCR -  Increment to be added to the sequence
                            number of each card.  Four numeric
                            characters must follow the INCR=
                            parameter; leading and/or trailing
                            zeros are required.
```

An example of the PUNCH control card is:

```
CTL     PUNCH PCBL,DATAL,OTHER,START=00000010,INCR=0010
```

The DD card for the output data set is called PUNCHDD; the data set characteristics are fixed, unblocked, LRECL=80.

DD Example:

```
//PUNCHDD      DD   SYSOUT=B
```

Other control cards:

```
COL 01           =  WTOR - puts message in remainder of card on
                           system console and waits for any reply,
                           then continues.

COL 01           =  WTO same as WTOR but does not wait for reply

COL 01           =  . or N; disregard this card

COL 01           =  ABEND - issue ABEND 252 with the DUMP option
```

SPECIAL CALL CARD FORMAT

```
$COL 01          =  L identifies this as a call card

$COL 05 thru 08  =  Number of times to repeat series of calls
                    with a range from 0001 thru 9999
$COL 10 thru 13  =  STAK - Start stacking control cards for
                           later execution
                    END  - Stop stacking control cards and
                           begin execution
                    STAT - Print the current buffer pool statistics
```

The purpose of this format is to provide the facility of repeating a series of calls which have been read from SYSIN and held in storage. All control cards between the STAK card and the END card are read and saved. When the END card is encountered, the series of calls is executed as many times as the number punched in COL 5 through 8 of the STAK card. This can be used to test exclusive control and scheduling by having two different regions executing stacks of calls concurrently.

FORMAT OF DISPLAY OF DL/I BLOCKS

The OS SNAP macro is used to display the DL/I blocks. The first block in the snap is the first part of module DFSBNUC0. Within this first block, the first decimal 40 bytes are the PSB directory. Following the PSB directory is the PST. The format of the remaining blocks displayed by the snaps will vary depending upon whether the blocks came from the ACB library or were dynamically built. The fields in the PST, such as PSTSCLAD and PSTDBPCB, can be used to locate relevant information in the subsequent blocks.

EXECUTION IN DIFFERENT TYPE REGIONS

This program is primarily designed to operate in a DLI or BMP type region but can also be executed in a MSG region. The input and output devices are dynamically established based on the type of region in which the program finds itself executing. In a BMP or DLI region, the execute card allows the program name to be different from the PSB name. There is therefore no problem with executing calls against any data base in a BMP or a DLI region. In a MSG region, the program name must be the same as the PSB name so, in order to execute in a MSG region, the program must be given the name or an alias of the PSB named in the IMS/360 definition.

When in a DLI region, input is read from SYSIN and output is written to PRINTDD.

When in a BMP region, if a symbolic input terminal was specified as the fourth parameter of the execute card, input will be obtained from that SMB, and output will be sent to the I/O PCB. The name of the I/O PCB may be specified as the fifth parameter of the execute card. If no SMB is specified on the execute card, SYSIN is used for input and PRINTDD is used for output just as in the DLI region.

In the MSG region, the I/O PCB is used for both input and output unless COL 2 of the status card is either a 1 or an A, in which case PRINTDD is used for output if the DD card is present in the JCL for that message region. A limit of 50 lines per schedule is sent to the I/O PCB and, after that, PRINTDD will be used for output if present; if not present, the program terminates.

Since the input is fixed form, it is difficult to key it from a terminal. The technique used by the development group to test DL/I in a message region using this program is to first execute another message program which, based on the message from the terminal, reads control cards which are stored as a member of a partitioned data set and insert the control cards to an SMB. This program is then scheduled by IMS/360 to process those transactions. This allows the same control cards to be used to execute in any region type.

HINTS ON USAGE

1. To load a data base:

   This program is only applicable for loading very small data bases, since all the calls and data must be provided to it rather than it generating data. It could be used to load large volume data bases if the control cards were generated as a sequential data set.

2. To display a data base:

   To display a data base, the following sequence of control cards may be used.

```
S 1 2 2 2 1        DBDNAME          Display comments and segment
L        GN                         DO 1 Get Next
EH8    OK                           Hold compare, GA, GK, OK, terminate on GB
L   9999 GN                         DO 9,999 Get Next calls
```

3. To do regression testing:

   This program is well suited to regression testing. By using a known data base, calls can be issued and the results compared to expected results using compare cards. The program then can, in effect, determine if DL/I calls are being executed correctly. By making the print options of the status card all twos, only those calls not satisfied properly will be displayed.

4. To use as a debugging aid

   When doing debugging work, usually a print of the DL/I blocks is required. By use of compare cards, the blocks may be displayed at appropriate times. Sometimes the blocks are needed even though the call is executed correctly, such as the call before the failing call. In those cases, an extra incorrect compare card may be inserted. This causes the blocks to be displayed even though the call was executed correctly.

5. To verify how a call is executed

   Since it is easy to execute a particular call, this program can be used to verify how a particular call is handled. This can be of value when DL/I is suspected of not operating correctly in a specific situation. The calls which are suspected can be issued using this program and the results examined.

SAMPLE JCL

```
//JCLSAMP JOB ACCOUNTING,NAME,MSGLEVEL=(1,1),MSGCLASS=3,PRTY=8
//GET EXEC PGM=DFSRRC00,PARM='DLI,DFSDDLTO,PSBNAMEE'
//IMS DD DSN=IMS2.PSBLIB,DISP=(SHR,PASS)
//    DD DSN=IMS2.DBDLIB,DISP=(SHR,PASS)
//DDCARD DD DSN=DATASET,DISP=(OLD,KEEP)
//IEFRDER DD DUMMY
//PRINTDD DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSIN DD *
S 1 1 1 1      DBDNAMEE
```

SAMPLE CONTRCL CARD INPUT

1. Data base load:

```
//SYSIN DD *
U START TEST LOAD
T ISRT ROOT SEGMENT A060000111
L           ISRT    A1111111
L           DATA    A0600011     1069999888  ROOT SEG1
EH
T ISRT ROOT SEGMENT A06C00511
L           ISRT    A111111
L           DATA    A060000511   1069999488  ROOT SEG2
L           ISRT    A111111      (A1111111 = A060000511)  X
                    AA222222
            DATA    XAA040511Z
/*
```

2. Data base retrieve and update

```
//SYSIN DD *
S 1 1 1 1 1                 1
L           GHU     JHNXXX       (J11NXXX = A10H102000)   *
                    JM2PABCX     (JM2PABCX = DIOHIO2A10)
S 1 1 1 1 1                 2
L           ISRT    J11NXXXX     (J11NXXXX = A10H02000)   C
                    JK2PADXX
L           DATA    A10HDC2000D10HD02A1U
S 1 1 1 1 1                 1
L           REPL
L           DATA    A10HD0200DB10HD02A10
```

APPENDIX B. FILE SELECT AND FORMATTING PRINT PROGRAM: DFSERA10

PROGRAM DESCRIPTION

The File Select and Formatting Print Program is provided as a service aid to be used with IMS/360 and its related data bases. Its primary function is to act as a utility to assist in the examination and display of data from the IMS/360 log data set. The program has the ability to:

- Print an entire log data set
- Print from multiple log data sets based upon control card input
- Select and print log records on the basis of sequential position in the data set
- Select and print log records based upon data contained within the record itself, such as the contents of a time, date, or identification field

These features are selected and controlled by a series of statements that allow the user to define the input options and selection ranges as well as to specify various field and record selection criteria.

PROGRAM INPUTS AND OUTPUTS

All data input is processed using QSAM and may reside on either tape or direct access storage devices. Data set organization should be physical sequential, while the record format may be fixed or variable in length, blocked or unblocked, or of undefined length. Since multiple log data sets can be used as input, multiple ddnames may be defined or, in the case of only one input, the default ddname of SYSUT1 can be used. The data set containing control information must be in card image form.

Parameter cards and related error messages are produced on the output print data set in the same format and sequence as they are processed. Data output is displayed in both hexadecimal and EBCDIC form, 32 bytes per line, with the hexadecimal relative offset value preceding each line.

PROGRAM CONTROL

The flow of control for the program passes through two major stages:

Control card processing - where construction of record test and selection parameters takes place and control card errors are diagnosed

Record selection and print processing - where the input data is read, analyzed, and compared with the selection parameters to determine the applicability of the record for printing

During the first phase, parameter cards are read and examined, and the required test or test series is constructed to create a test group. This test group is then used in record selection when control passes to the next phase of the program. In the second phase, the input data records are read, and disposition is decided by the results of each test in the group. When the end of the input data is reached, either by an end-of-file condition being encountered or the indicated record count being satisfied, program control shifts back to phase one, where the next group of tests is constructed.

CONTROL STATEMENTS

Three types of control statements are used to guide the program through the described phases. An additional card type can be used to provide titles or comments on the output listings. Operands on these cards may be extended to additional cards, to a maximum of nine, by placing a nonblank character in column 72 and continuing the operand in column 16.

The CONTROL statement defines the beginning and ending limits of the data set to be scanned; it also provides the ddname of the data input if the default name of SYSUT1 is not used. Inclusion of this card is optional if the default operands are satisfactory.

The OPTION statement defines the test or series of tests to be performed upon the data of the candidate record to determine its qualification for selection. One or more tests can be executed on each logical record by the appropriate number of OPTION cards, creating the logical "OR" function. Records can be analyzed with the logical "AND" function by using the multifield test capability of the COND operand and the necessary number of OPTION cards, creating a test series. The operands COND=M and COND=E are used to denote the beginning and ending, respectively, of a series for multifield testing of a record. The maximum length of selection parameters that may be specifed by the VALUE= operand is 510 bytes.

The END statement is a delimiter used to separate one group of tests (comprised of one or more OPTION statements), from subsequent groups of tests on the next data set. When an END statement is encountered in the control input stream, the construction of record selection parameters ceases and the processing of input data records starts. Proper use of the END statement allows one execution of the utility program to perform a varied number of tests on one or more IMS/360 log data sets.

The * or Comments statement may be used to include any information deemed helpful by the user to identify tests or data and has no effect on the utility program.

JCL REQUIREMENTS

The File Select and Formatting Print program executes as a standard operating system job and, as such, requires a JOB card as defined by the users installation. Additionally, an EXEC and appropiate DD cards to define inputs and outputs are required.

EXEC	This statement must be of the format PGM=DFSERA10 or may be included in a cataloged procedure.
SYSPRINT	This statement describes the output data set to contain the formatted print records and control messages. It will usually be defined as SYSOUT=A.
SYSIN	This statement describes the input control data set. This file must be in card image format.
input data DD	These statements define the input data set(s) to be examined to produce the formatted print records. These data sets must be standard labeled files, either direct access or tape. They may be of any record format (F, FB, V, VB, VBS, or U), as long as they are of DSORG=PS. If a file with RECFM=U is used, the DCB BLKSIZE parameter must be specified. These files are processed using QSAM. Therefore, any file that QSAM supports can be described as input.
SYSUT1 DD	This statement defines the default ddname used for data input if explicit reference is not used.

Parameter Card Descriptions

CONTROL Card Format

1	10	16
CONTROL	CNTL	[SKIP = {<u>0</u> | aaa}] [,STOPAFT = {<u>16777215</u> | bbb | EOF | (bbb,E)}] [,DDNAME = {<u>SYSUT1</u> | ddname}]

This card is optional. If not specified, the default values cause the SYSUT1 input file to be examined.

SKIP=

This keyword is used to define the first record tested. All prior records are ignored.

If this keyword is not specified, a default value of zero is used and causes the first record on the input file to be tested.

aaa

The value specified must be in the range of zero to 999999 and cannot have embedded commas.

STOPAFT=

This keyword is used to define the last record to be tested. When this value has been reached by counting processed records, the current group of tests is terminated.

If this keyword is not specified, a default value of 16777215 is used.

bbb

The value specified must be in the range of 1 to 99999999 with no embedded commas. If the value zero is specified, one record will be processed. The word EOF, which denotes end-of-file condition, will allow record processing beyond the stated maximum of 99999999 records.

E

This parameter causes records to be counted for test sequence termination only if they satisfy selection criteria. Otherwise, all records read (after the SKIP value) will be counted.

DDNAME=

This keyword is used to identify the input data set for the current group of tests. A corresponding DD statement must be supplied.

If this keyword is not specified, a default of SYSUT1 is used and the appropriate DD statement must be supplied.

ddname

This name must be the DDNAME of the input file if the default of SYSUT1 is not used.

OPTION Card Format

```
|1          |10        |16
|           |          |  [OFFSET = { 1 }]  [,FLDTYP = { X }]
|           |          |  [         {aaa}]  [           { C }]
|  OPTION   |  PRINT   |
|           |          |  [,VALUE = bbb] [,  FLDLEN = { 1 }]
|           |          |                 [            {ddd}]
|           |          |  [,COND = { E          }]
|           |          |  [        { M          }]
|           |          |  [        { T   {Y}    }]
|           |          |  [        {     {N}    }]
|           |          |  [        { MT  {Y}    }]
|           |          |  [        {     {N}    }]
|           |          |  [        { ET  {Y}    }]
|           |          |  [        {     {N}    }]
```

This card causes one set of tests to be constructed. One or more OPTION statements can be defined for one or more sets of tests to be performed against each input record. When all operands of this statement are omitted, all records processed by phase two of the program are printed.

PRINT

This parameter is used to cause records satisfying this test, and prior tests in this series if this is a multifield test, to be displayed on the SYSPRINT data set.

OFFSET=

This keyword is used to define the location of the first byte of the field to be tested in the record. Default value is position one of the record.

aaa

This value can be in the range from one up to and including the length of the record under test. Maximum value is 32767 bytes, and no checking is performed to determine if the logical record length is exceeded.

Note: If DSECTs are used to locate values in control records or blocks, the user must adjust the starting value for the OFFSET parameters. Most DSECTs start with a relative value of ZERO, while the value specified in the OFFSET keyword is always expressed as relative to byte one.

FLDTYP=

This keyword is used to define the type of data in the VALUE=field.

X

This parameter defines the data to be treated as hexadecimal pairs. The test data will be packed - two bytes into one to form hexadecimal equivalents. This is the default value.

<u>Example</u>: If VALUE=D9D6D6E3E2C5C7 (14 bytes) is specified with the FLDTYP=X parameter, then the resultant VALUE= will look like this: ROOTSEG in EBCDIC or D9D6D6E3E2C5C7 in hexadecimal; in either case, the length is only 7 bytes.

C

This parameter defines the data to be treated as EBCDIC. The test data will be used as punched in the card, with no alterations.

VALUE=

This keyword defines those characters that comprise the test field. If FLDTYPE=X was specified, this data must be entered as hexadecimal character pairs. For a 'Test Under Mask' condition, a single pair must represent the hexadecimal value for the test. If FLDTYP=C was specified, this data must be entered as EBCDIC characters. If the characters of blank or comma are to be included in this operand, FLDTYP=X must be used with the appropriate hexadecimal equivalent.

bbb

This value can not exceed 255 EBCDIC or 510 hexadecimal characters. The length of this field is determined by the FLDLEN= keyword value and not by the number of 'non-null' characters in this field.

COND=

This keyword defines the type of test and its relationship to other tests in the group.

M

This parameter indicates that this is a multifield test. That is, more than one test is to be made on each input record. All tests in this series must be satisfied before the record will be printed.

E

This parameter marks the last (or only) element in a test series. Any OPTION control statements appearing after this form a new series of tests. This allows various tests to be performed on each record and each test series can be used upon different fields within the record.

T

This parameter causes the VALUE= byte to be used as a 'Test Under Mask' value, instead of a compare field. Only the first byte (two hexadecimal characters if FLDTYP=X) of the VALUE= field will be used. If FLDTYP=C is used, the hexadecimal equivalent of the EBCDIC character will be the test value. If this parameter is used, the FLDLEN= keyword must not be specified and a default length of one will be assumed.

Y

This parameter indicates that for the 'Test Under Mask' to be considered satisfied, there must be a bit in the record test field for each corresponding bit of the test byte. This is equivalent to a 'Branch if Ones' Test.

N

This parameter indicates that for the 'Test Under Mask' to be considered satisfied, there must not be a bit in the record test field for any of the corresponding bits of the test byte. This is equivalent to a 'Branch if Zeros' test.

MT

This parameter defines a 'Test Under Mask' OPTION as described above in the 'T' discussion but with the properties of a multifield test as described in the 'M' discussion. Since the T parameter assumes a default value of one, the MT parameter must be used for a multifield test that starts with a 'Test Under Mask' value.

ET

This parameter signifies that a multifield test seroes ends with a 'Test Under Mask' condition.

FLDLEN=

This keyword defines the number of characters to be used from the test field.

ddd

This value represents the actual number of bytes to be used, not the number of characters specified in the VALUE= keyword. The acceptable range of values for this field is one to and including 255. Default value is 1.

END Card Format

```
|1            |10         |16                          |
|   END       |           |                            |
|             |           |                            |
```

When all tests have been defined for the current input file, this card must be used to cause execution of those tests to begin.

Any columns of the END card after 9 can be used for comments.

COMMENTS Card Format

```
|1            |10         |16                          |
|*            |           |                            |
|             |           |                            |
```

This card is optional and, if used, causes its contents to be displayed on the SYSPRINT data set.

Examples

Example 1: Print all records from a data base Image Copy data set

```
//EPRT  JOB
//IMAGEX1  EXEC PGM=DFSERA10
//SYSPRINT DD   SYSOUT=A
//SYSUT1   DD   DSN=IMAGE,VOL=SER=123456,UNIT=2400,LABEL=(,SL),
//              DISP=(OLD,KEEP)
//SYSIN    DD   *
* THE ABSENCE OF A CONTROL CARD WILL ASSUME SYSUT1 INPUT
* THIS OPTION CARD WILL CAUSE THE ENTIRE FILE TO BE PRINTED
OPTION     PRINT
END
/*
```

Example 2: Print the first 30 records from an IMS/360 log tape

```
//RINT  JOB
//LOGEX1   EXEC PGM=DFSERA10
//SYSPRINT DD   SYSOUT=A
//LOGTAP1  DD   DSN=IMSLOG,VOL=SER=111111,UNIT=2400,LABEL=(,SL),
//              DISP=(OLD,KEEP)
//SYSIN    DD   *
* THIS CONTROL CARD DEFINES THE INPUT TO BE LOGTAP1
* AND ONLY THE FIRST 30 RECORDS FROM THE FILE WILL BE USED
CONTROL CNTL DDNAME=LOGTAP1,STOPAFT=30
OPTION PRINT
END
/*
```

Example 3: Extract log records of a type X'50'. Only those records after record No. 1000 are to be examined.

```
//RINT JOB
//LOGEX2   EXEC PGM=DFSERA10
//SYSPRINT DD   SYSOUT=A
//LOGTAP1  DD   DSN=IMSLOG,UNIT=2400,VOL=SER=111111,LABEL=(,SL),
//              DISP=(OLD,KEEP)
//SYSIN    DD  *
CONTROL CNTL SKIP=1000,DDNAME=LOGTAP1
* THIS OPTION CARD CAUSES ONLY THOSE RECORDS WITH A HEX '50' IN
* RECORD BYTE 5 TO BE PRINTED ON THE SYSPRINT DATA SET.
OPTION PRINT OFFSET=5,FLDTYP=X,FLDLEN=1,COND=E,VALUE=50
END
/*
```

Example 4: Print record No. 158 of an OSAM Image Copy data set and all type X'50' records on a log tape that references this block number (assuming unblocked OSAM).

```
//PRNT  JOB
//COMBEX1  EXEC PGM=DFSERA10
//SYSPRINT DD   SYSOUT=A
//IMAGFILE DD   DSN=OSAMIMAG,UNIT=SYSDA,DISP=SHR,VOL=SER=DA0002
//SYSUT1   DD   DSN=IMSLOG,UNIT=SYSDA,VOL=SER=DA0003,DISP=SHR
//SYSIN    DD   *
* THIS CONTROL CARD CAUSES THE INPUT FILE IMAGFILE TO CLOSE
* AFTER THE FIRST RECORD OF THE SELECTED GROUP IS PRINTED
CONTROL  CNTL   STOPAFT=(1,E),DDNAME=IMAGFILE
OPTION   PRINT FLDEN=4,OFFSET=1,FLDTYP=X,COND=E,VALUE=0000009E
* THIS END CARD CAUSES THE SELECTION OF THE FILE TO BEGIN.
END
* THIS CONTROL DEFAULTS TO THE STANDARD INPUT FILE, SYSUT1
CONTROL CNTL
* THIS CARD LIMITS THE SELECTION TO ONLY THOSE RECORDS
* THAT CONTAIN A HEX '50' IN RECORD BYTE 5
OPTION   PRINT  FLDLEN=1,OFFSET=5,FLDTYP=X,COND=M,VALUE=50
* THIS CARD FURTHER LIMITS SELECTION TO ONLY THOSE RECORDS THAT
* CONTAIN THE DATA BASE NAME OF DATABAS1 IN RECORD BYTES 25 THRU 32
OPTION PRINT FLDTYP=FLDTYP=C,FLDLEN=8,OFFSET=25,COND=M,VALUE=DATABAS1
* THIS CARD FURTHER LIMITS SELECTION TO ONLY THOSE RECORDS THAT
* CONTAIN THE FLAG MARKING THIS RECORD AS AN OSAM RECORD
OPTION   PRINT  FLDTYP=X,OFFSET=7,COND=MTN,VALUE=04
* THIS CARD FURTHER LIMITS SELECTION TO ONLY THOSE RECORDS THAT
* CONTAIN THE RBN OF 0000009E
OPTION   PRINT  FLDLEN=4,OFFSET=43,COND=E,VALUE=0000009E,FLDTYP=X
* THIS CARD CAUSES THE SELECTION OF RECORDS TO BEGIN
END
/*
```

INDEX

This index was prepared using an automated indexing program which is under continuing development. Your comments and suggestions will be appreciated.

SH20-0911-4

IMS/360 Version 2, System Programming Reference Manual Printed in U.S.A. SH20-0911-4

IBM®

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

IMS/360 Version 2
System Program Reference Manual

SH20-0911-4

Reader's Comment Form

Your comments about this publication will help us to improve it for you. Comment in the space below, giving specific page and paragraph references whenever possible. All comments become the property of IBM.

Please do not use this form to ask technical questions about IBM systems and programs or to request copies of publications. Rather, direct such questions or requests to your local IBM representative.

If you would like a reply, please provide your name, job title, and business address (including ZIP code).

Name: ______________________________

Job Title: ______________________________

Address: ______________________________

______________________ Zip ______________

Fold on two lines, staple, and mail. No postage necessary if mailed in the U.S.A. (Elsewhere, any IBM representative will be happy to forward your comments.) Thank you for your cooperation.

SH20-0911-4

Your comments, please . . .

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

fold fold

FIRST CLASS
PERMIT NO. 439
PALO ALTO
CA. 94301

Business Reply Mail

No postage stamp necessary if mailed in the U.S.A.

Postage will be paid by:

IBM Corporation
Department J04
1501 California Avenue
Palo Alto, California 94304

fold fold

IMS/360 Version 2, System Programming Reference Manual Printed in U.S.A. SH20-0911-4

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)