File Number S360-31
Form C24-5033-7

DOS

IBM

Systems Reference Library

IBM System/360 Disk Operating System System Generation and Maintenance

This reference publication describes the specifications and operating procedures for generating an installation-tailored Disk Operating System. With the IBM-supplied Disk Operating System and an IBM 2311 Direct Access Storage Device, or an IBM 2314 Direct Access Storage Facility, a Disk Operating System can be generated that supports those IBM-supplied programs desired by the user, in addition to his own programs.

With the storage requirements provided, an installation-tailored Disk Operating System can be planned. With the sample problems provided, the generated system can be tested. A thorough understanding of IBM System/360 Disk Operating System concepts and machine facilities are prerequisites to the effective use of this manual. Major topics discussed in detail are:

- Planning an Operational Pack
- System Generation and Maintenance
  - Macro Instructions for Generating a Supervisor
  - Maintenance Procedures
- Sample Problems
- Storage Requirements

Prerequisites to a thorough understanding of this manual are:

IBM System/360 Disk Operating System, System Control and System Service Programs, Form C24-5036

IBM System/360 Disk Operating System, Supervisor and Input/Output Macros, Form C24-5037

IBM System/360 Disk Operating System, Operating Guide, Form C25-5022

IBM System/360 Disk and Tape Operating Systems, Utility Programs Specifications, Form C24-3465

IBM System/360 Basic Programming Support, DASD Utility Programs Specifications, Form C24-3363

For titles and abstracts of other associated publications, see the IBM System/360 Bibliography, Form A22-6822.

Eighth Edition (April 1969)

This edition applies to Release 20 of IBM System/360 Disk Operating System and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the specifications herein; before using this publication in connection with the operation of IBM systems, consult the latest System/360 SRL Newsletter, Form N20-0360, for the editions that are applicable and current.

This edition, C24-5033-7, is a major revision of, and obsoletes, C24-5033-6.

Summary of Amendments

This edition reflects support for:

- Overlapping of I/O with processing for both the load and sequential retrieve functions for indexed sequential files.
- The capability of returning additional unrecoverable I/O errors to the problem program using the direct access, sequential disk, indexed sequential and magnetic tape macro functions.
- The 1401/1440/1460 Emulator Program for the IBM System/360 Model 40 and Model 25.
- The librarian copy and organize program extension that allows merging of libraries.
- PL/I Version 4.
- BTAM and QTAM in a multitasking environment.
- COBOL improvements for Indexed Sequential File Management

Maintenance changes and technical corrections are also included.

The storage estimates, formerly published in IBM System/360 Disk Operating System Performance Estimates, Form C24-5032, are now published in Appendix G of this manual.

Changes are indicated by a vertical line to the left of affected text and to the left of affected parts of figures. A dot (•) next to a figure title or page number indicates that the entire figure or page should be reviewed.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Programming Publications, Endicott, New York 13760.

Preface

This publication guides the user through the generation of an installation-tailored Disk Operating System for IBM System/360. The information herein is of particular interest to anyone who wants to build a Disk Operating System, including installation managers, system analysts, programmers and machine operators.

In particular, the user will find that familiarity with the following system programs and facilities is invaluable when using this publication: the control program, the system service programs, and the input/output control system (IOCS) logic modules. The control program, and the system service programs are described in the System Control and System Service Programs publication. The IOCS logic modules are described in the Supervisor and Input/Output Macros publication, both of which are listed on the front cover.

The publication is divided into the following major sections, and seven appendixes:

1. Introduction
2. Planning an Operational Pack
3. System Generation and Maintenance Procedure
4. IBM BPS Utility Programs
5. Macro Instructions for Generating a Supervisor
6. Two 2311 Disk Drives
7. One 2311 Disk Drive
8. IBM 2314 Direct Access Storage Facility
9. Maintenance Procedures
10. Disk Operating System Sample Problems

The first five sections contain required information for most users. Sections six, seven, and eight refer to specific machine configurations, and present techniques for generating a system. They are examples intended to guide the user through the generation of his system. Only the section that relates to the user's configuration need be read.

Section nine describes maintenance procedures and how to apply IBM-supplied Disk Operating Systems to maintain an existing Disk Operating System.

Section ten describes the IBM-supplied sample problems that can be used to test a system after it is generated.

The appendixes contain information needed for planning a Disk Operating System. Appendix B, IOCS Modules for COBOL, RPG, and PL/I; Appendix C, IBM System Components Identification; and Appendix G, Storage Requirements are of major importance for planning purposes. Appendixes B and C will assist the user when planning the contents of his libraries, and Appendix G when estimating the size of his supervisor, and libraries.

Closely related publications are:

IBM System/360 Disk Operating System, Timing Estimates, Form C24-5032

IBM System/360 Disk Operating System, Vocabulary File Utility Programs for IBM 7772 Audio Response Unit, Form C27-6924

IBM System/360 Basic Programming Support, Distribution Program Specifications and Operating Guide, Form C21-5001

IBM System/360 Basic Programming Support, DASD Utility Programs Operating Guide, Form C24-3392

References are made in this publication to the following:

IBM System/360 Disk and Tape Operating Systems, COBOL Programmer's Guide, Form C24-5025

IBM System/360 Disk Operating System, FORTRAN IV Programmer's Guide, Form C28-6397

IBM System/360 Disk and Tape Operating Systems, Basic FORTRAN IV Programmer's Guide, Form C24-5038

IBM System/360 Disk and Tape Operating Systems, PL/I Programmer's Guide, Form C24-9005

IBM System/360 Disk Operating System 1401/1440/1460 Emulator Programs for IBM System/360 Models 30 and 40; Compatibility Support/30; Compatibility Support/40, Form C27-6940

IBM System/360 Disk Operating System Autotest, Form C24-5062.

Contents

Introduction

The IBM System/360 Disk Operating System for 2311 resident systems is supplied to the user in two volumes. The volumes can be identified as a relocatable volume and a source statement volume. Users with a tape unit available receive the volumes on tape reels; users without a tape unit available receive the volumes on disk packs. These volumes are manipulated during system generation to create an operational system.

IBM supplies the 2314 system on two tape reels. These tapes will be restored to a 2314 disk pack to create a system consisting of a core image library, a relocatable library, and a source statement library. The capacity of one 2314 disk pack is enough to accommodate both tape reels shipped by IBM.

The disk operating system is composed of three libraries:

Source Statement Library

Relocatable Library

Core Image Library

The source statement library contains IBM-supplied macro definitions. When the desired parameters are chosen, the macros can be assembled. For the user's convenience, the source statement library also contains sample problems and system generation job streams that can be retrieved as needed.

The relocatable library contains IBM programs that have not been assigned addresses for execution and assembled macros from the source statement library. These assembled macros perform input and output procedures for IBM-supplied programs. These assembled macros (Logical IOCS modules) can also be used by user programs when applicable.

The core image library contains programs that are ready for execution. System control programs and system service programs are always shipped to the customer in the core image library. Where it is necessary for system generation purposes, an assembler program is provided. The system control programs must always be part of the system. The librarian programs are a key set to the system and should be carefully considered before ever removing them from the system.

During system generation the user works with the IBM-supplied system to tailor it to his individual needs. This consists of adding to and deleting from the libraries, IBM and user code (source, relocatable and core image).

Planning System Generation

Proper and detailed planning saves on total system generation time. Thus, it is very important to perform a very thorough job.

Planning system generation consists of:

1. Planning the contents, organization, and ultimate size of the system and/or private libraries. This entails distributing the storage space available (on the disk packs) between the libraries ultimately desired for day-to-day use. Major points of consideration are:

- the size of the core image library and, if desired, system and/or private relocatable and source statement libraries

- workfile space needed to assemble a supervisor, and to accommodate the linkage editor, which is needed to catalog the components selected to the system core image library

- standard assignments (allocation of space) for workfiles (assemblies), and linkage editing needed for every day operation.

2. Planning the contents and estimating the size of a supervisor. This entails selecting from the programming services provided by IBM, those to be included in the supervisor, and estimating the cost of these services in terms of bytes of storage.

PLANNING THE LIBRARIES

Two types of IBM libraries are:

1. System libraries

2. Private libraries

The system libraries are the core image, the relocatable, and the source statement. The private libraries are the private relocatable, and the private source statement library. (There is no private core image library.)

CORE IMAGE LIBRARY

Because the core image library contains the executable format of programs, it is the library in which the user is most likely to keep his programs. Otherwise, the programs must continually be placed in the core image library before each execution (linkage edited). Therefore, during system generation the user expands the size of the core image library to accommodate all the programs desired resident and on-line. (Both his own programs and IBM's.)

In addition, the user should try to envision future space he may require, and provide this space if possible. Such planning can eliminate the need for another system generation. Thus, to expand the core image library means making the remaining libraries smaller on the pack.

Before the size of a library is reduced, the user must delete those items that are not to be used, or those items that were transferred to another library in the required format. It is recommended that backup of the system (a copy of the IBM-supplied system) be obtained to protect a user who accidently removed something he desired from the system. It is also recommended that after successful completion of segments of system generation, a copy (backup) of the partially generated system be obtained, i.e., upon the creation of a library, or the assembly of a supervisor. This permits a user to return to a point other than the beginning of his procedure in case of an error. The components supplied in the IBM-shipped core image library facilitate system generation.

RELOCATABLE LIBRARY

All IBM-supplied components are shipped in the relocatable library. This library is the basis for the creation of a private relocatable library. Thus, it is from this library that most IBM components are directly or indirectly extracted (indirectly in the case of the existence of a private relocatable library and a system relocatable library).

SOURCE STATEMENT LIBRARY

All IBM-supplied macro definitions are in this library. This library is the basis for the creation of a private source statement library. Thus, it is from this library that many users extract, directly or indirectly, IBM-supplied component macro definitions (indirectly in the case of the existence of both a private source statement library, and a system source statement library).

PRIVATE LIBRARIES

If the user has more than one disk drive available, it is not necessary to decrease the size of the relocatable and source statement libraries. They can be assigned to other disks and are then referred to as private libraries. The user can build systems with private and system libraries containing those items that best fit his needs.

LIBRARY SIZES

The user must choose which of the libraries he desires, and then plan their precise content and size for daily use. Thus, the user should know the initial, intermediate, and final sizes of the libraries throughout system generation, and he must plan the exact contents of each library that is created during system generation. These contents should be listed, along with their sizes, and then the total number of cylinders to be allocated can be calculated.

The contents of the libraries are identified in Attachment 1 of the Memorandum To Users that accompanies the IBM system that is shipped. The storage requirements (sizes) for these components, and macro definitions are identified in Appendix G: Storage Requirements. Components shipped in the relocatable library are also identified in Appendixes B, C, D, and E. Those macro definitions supplied by IBM are also identified in Appendix C.

An alternate method for determining the number of cylinders to allocate for a library is given in the discussion Allocating Library Sizes for 2311 and 2314 Disk Systems. Note that once private library sizes have been allocated, they cannot be reallocated. Although excess room in the libraries may not offer maximum efficiency, in case of a calculation error, the benefits can be easily recognized when sufficient space is available.

ALLOCATING LIBRARY SIZES

When the user desires to reduce the size of an existing library, it may be important to know the minimum library size that can be allocated. This size should be increased to accommodate any additional user programs to be included in this library. The size of any IBM component to be included in this library can be determined by referring to the storage requirements given in Appendix G. For the details on how to allocate an existing library, see Allocating Library Sizes for 2311 and 2314 Disk Systems.

PLANNING A SUPERVISOR

The supervisor is a control program that provides specialized services to programs executed from the problem program area of main storage. This program is composed of a group of assembled macros (see Macro Instructions for Supervisor Generation. The options selected in each of the supervisor generation macros determine the size of the assembled supervisor. The size of each option is identified in Appendix G.

The SEND macro determines the end of the supervisor. If the user assembles a supervisor with a SEND address larger than the previous supervisor, some portion of the supervisor is overlaid by programs linkage edited to the previous SEND address.

The user must re-linkage edit programs after the new supervisor is loaded so that the new load address follows the newly assembled supervisor.

Note that at each IPL, the user must include ADD and ASSGN statements for each device until a system supervisor is built that describes his machine configuration and standard I/O assignments (see DVCGEN and ASSGN macros).

To calculate the size of the supervisor the user must list all of the options that are chosen and then determine the sum of the sizes obtained from the storage requirements (see Appendix G).

When choosing supervisor parameters, the user must check to see if the chosen options are compatible. Also, the user should check to determine which parameters automatically provide the support for another.

System Configuration

This section presents the minimum system configuration required to operate the Disk Operating System. The system control programs and basic IOCS must always be present in order to execute any other programs.

MACHINE REQUIREMENTS

Minimum features required:

16K bytes of main storage

Standard instruction set. See Note 1.

One I/O channel (either multiplexor or selector). See Note 2.

One Card Reader (1442, 2501, 2520, or 2540). See Note 3.

One Card Punch (1442, 2520, or 2540). See Note 3.

One Printer (1403, 1404, or 1443). See Note 3.

One 1052 Printer-Keyboard.

One 2311 Disk Storage Drive or

One 2314 Direct Access Storage facility

Note 1: Language translators may require extended instruction sets.

Note 2: Telecommunications requires a minimum of two channels, one multiplexor channel, and at least one selector channel. One channel is required for telecommunications, and the other for the system resident device. (Telecommunication devices should not be on the same selector channel as SYSRES.) Note that a 2701 Line Adapter Unit attached to an IBM System/360 Model 2025 must be placed on the multiplexor channel.

Note 3: One 2400-series magnetic tape unit may be substituted for this device (7- or 9-track). If 7-track tape units are used, the data-convert feature is required, except when substituted for a printer.

Planning an Operational Pack

An operational system is one used in day-to-day operations that contains a tailored supervisor and libraries appropriate to each customer's particular combination of system programs and application programs. A system maintenance volume is one used primarily to facilitate changes to programs supplied by IBM. Application programs can also be added to a maintenance volume for ease of program maintenance.

Change Distributions are those changes supplied by IBM to the IBM-shipped volume.

Proper planning is the key to successful system generations. The time spent in planning can save the user frustration and valuable time. The user's planning should reflect the initial, intermediate and ultimate capacity of the core image, relocatable and source statement libraries. Enough workfile capacity must be available throughout system generation for assemblies and linkage-edit steps.

Appendix G should be used to determine storage requirements for the components in the system.

IBM supplies a system containing either a 6K for 2311, or 8K for 2314 supervisor. The IBM supervisors are those described in Figure 6.

BACKGROUND PARTITION STORAGE REQUIREMENTS FOR DISK OPERATING SYSTEM IBM-SUPPLIED PROGRAMS

All IBM-supplied programs used with Disk Operating System that are language translators or utilities execute in the background partition. Figure 1 lists the minimum size background partition required.

IBM-Supplied Program	IBM-Program Number	Minimum Size Background Partition (in bytes)
Assembler (IJQD16TW) or (IJQD16DW)	360N-AS-465	10,240
Assembler (IJQD32)	360N-AS-465	14,336
Assembler (IJYASM)	360N-AS-466	45,056
COBOL	360N-CB-452	14,336
Basic FORTRAN	360N-FO-451	10,240
FORTRAN	360N-FO-479	40,960
RPG	360N-RG-460	10,240
PL/I	360N-PL-464	10,240
Utilities		
Group 1	360N-UT-461	10,240
Group 2	360N-UT-462	10,240
Group 3	360N-UT-463	10,240
Tape Sort/Merge	360N-SM-400	10,240
Disk Sort/Merge	360N-SM-450	10,240
Tape/Disk Sort/Merge	360N-SM-483	
Tape/2311		10,240
2314		22,528
1401/1440/1460 Emulator for IBM Model 30	360N-EU-484	16,384
1401/1440/1460 Emulator for IBM Model 40	360N-EU-485	*
Autotest	360N-PT-459	10,240

* The background partition storage requirement is as follows: 16,384 bytes for the supervisor plus 2,048 to 16,384 bytes for the 1400 being emulated plus the size of the emulator program defined.

• Figure 1. Disk Operating System Background Partition Storage Requirements for IBM-Supplied Programs

ONE 2311 DISK DRIVE

When the user plans his operational system he decides upon the ultimate appearance of his libraries. The one disk drive user will find it most convenient to build his operational system on the IBM volume that contains the core image and relocatable libraries. Many single disk drive users will want to build operational systems appearing as one of the following:

- Core image library, small system relocatable library, and/or small system source statement library
- Large core image library

The small system relocatable library is for users who need the compiler subroutines and/or system IOCS modules on line at all times. If the subroutines are to be on line, the relocatable library should also have enough room to contain the user's largest compiler.

The samll system source statement library can contain system control and logical IOCS macros. The system with the source statement library, supplied by IBM, can be used as an assembly pack.

AT LEAST TWO 2311 DISK DRIVES OR A 2314 DIRECT ACCESS STORAGE FACILITY

The user with at least two disk drives will find it most convenient to build his operational pack upon the IBM volume that contains the core image and source statement libraries. Many multiple disk drive users will want to build operational system appearing as one of the following:

- Core image library, private relocatable, and private source statement libraries.
- Core image library, private relocatable library, system source statement library, and private source statement library.
- Core image library, system relocatable library, private relocatable library, and private source statement library.
- Core image library, system relocatable and source statement libraries, and private relocatable and source statement libraries.
- Core image library, system relocatable and system source statement libraries.
- Core image library, system source statement library, small private relocatable library (on the operational pack) and private relocatable library.

IBM system control and system service programs are supplied in the core image libraries of both volumes for the 2311 systems and in the core image library for the 2314 system. The 2311 systems have the 10K background disk workfile assembler supplied in the core image library of the source statement library volume. The 2314 systems have the 14K background disk and tape workfile assembler in the core image library. For both 2311 and 2314 systems, all IBM components are supplied in the relocatable library.

If the user's assembled supervisor does not exceed the size of the IBM supervisor, re-linkage editing and re-cataloging of the IBM-supplied programs shipped in the core image library are unnecessary. Job control is self-relocating and need never be re-linkage edited. Only the steps required to retrieve the sample problems, delete unwanted components, assign standard labels, assemble a user's supervisor, allocate for and linkage edit IBM components, and condense libraries are necessary to perform system generation.

When the number of tracks required for each library has been calculated, the user should allocate a sufficient number of cylinders to each library of each operational system. Additional cylinders may be allocated to the core image library for application programs. Sometimes the user must reallocate the libraries on his disk pack(s) during system generation to ensure sufficient work file storage for assemblies and linkage-edit steps.

Users with more than one available disk drive will find it to their advantage during system generation to define private relocatable and source statement libraries for an operational system.

ALLOCATING LIBRARY SIZES FOR 2311 AND 2314 DISK SYSTEMS

When the size of an existing library is reduced, it may be important to know the minimum size library that can be allocated. Once the minimum size library is calculated, it should be increased to accommodate any user programs to be included in the library. The size of IBM components can be determined by referring to the storage requirements given in Appendix G.

The following explanation illustrates how to calculate the number of tracks required for a core image, relocatable, or source statement library for a 2311 or 2314 disk system. The formula for computing the size of a library is:

Library (size in tracks) = LBA/nn

where: library = either core image, relocatable or source statement

LBA = the number of LIBRARY BLOCKS ACTIVE for the library of interest and is obtained from a DIRECTORY (SYSTEM or PRIVATE)

nn = LIBRARY LAST AVAILABLE ENTRY in the R (record) column

Using the sample PRIVATE DIRECTORY that follows, an example of how to compute a library size is:

Relocatable Library = LBA/nn

where: LBA = 9849
nn = 9 , thus
Relocatable Library = 9849/9 = 1,094.3 tracks

The relocatable library size computed does not include the tracks allocated for the directory (see <u>Directory Allocated Tracks</u>). The directory size must be added to the relocatable library size computed. Thus,

Relocatable Library Allocation = Relocatable Library + Directory Allocated Tracks

Therefore,

Relocatable Library Allocation = 1,094.3+9
= 1,103.3 Tracks

For a 2311, cylinders = $\frac{\text{Library Size (Tracks)}}{10}$

For a 2314, cylinders = $\frac{\text{Library Size (Tracks)}}{20}$

For this example,

the Relocatable Library Allocation for a 2311 = $\frac{\text{1103.3 Tracks}}{10}$
= 110.3 or 111 cylinders,

rounded high

```
PRIVATE DIRECTORY                 PRIVATE-RELOCATABLE

02/01/69                          ------DECIMAL------

                                      C  H  R  E
DIRECTORY STARTING ADDRESS           01 00 01
DIRECTORY NEXT ENTRY                 01 05 01 08
DIRECTORY LAST ENTRY                 01 08 09 19

LIBRARY   STARTING ADDRESS           01 09  01
LIBRARY   NEXT AVAILABLE ENTRY      111 03  04
LIBRARY   LAST AVAILABLE ENTRY      127 09  09

                                  STATUS INFORMATION


DIRECTORY ENTRIES ACTIVE            903


LIBRARY   BLOCKS ALLOCATED        11349
LIBRARY   BLOCKS ACTIVE            9849
LIBRARY   BLOCKS DELETED             00
LIBRARY   BLOCKS AVAILABLE         1500


AUTOMATIC CONDENSE LIMIT             00


LIBRARY   ALLOCATED CYLINDERS       127
DIRECTORY ALLOCATED TRACKS           09
```

USER DECISIONS AND CONSIDERATIONS

The following considerations and decisions should be made before system generation:

1. Select supervisor options by coding a set of supervisor macro instructions (see Macro Instructions for Generating a Supervisor).

2. Determine which programs will be in the core image library of each operational pack (e.g. COBOL, FORTRAN, etc).

3. Determine which assembler will be used to generate a new supervisor. (See Variants of IBM-Supplied Assembler.) The 2311 systems have the 10K background disk workfile assembler supplied in the core image library of the source statement volume. The 2314 systems have the 14K background disk and tape workfile assembler supplied in the core image library.

4. Determine which modules are to be deleted from the relocatable library of each operational pack. Deleting from the relocatable library allows the user to expand the core image library to hold a greater number of components. Refer to Appendix G for IBM component sizes.

5. Users must also determine if the macro definitions used to build the supervisor and IOCS modules are to be deleted from the source statement library. Retaining the macros in the source statement library facilitates building a new supervisor and new user IOCS modules.

6. When possible, maintenance is supplied on magnetic tape or cards, rather than on disk packs. (Unusually large maintenance distributions are supplied on a disk pack to 2311 resident systems without magnetic tapes.) If a seven-track tape unit is to be used for systems maintenance, it must have the Data Conversion feature. The 2314 resident systems will always receive maintenance on tape.

7. One disk drive users may prefer to maintain only enough room in the relocatable library of the operational pack to contain the modules used to build the largest component in the system. This small relocatable library permits temporary insertion of any component in relocatable form. It can then be immediately linkage edited into the core image library and then deleted from the relocatable library. When the relocatable library is subsequently condensed, only the updated core image form of the component remains, thus conserving disk-storage capacity. Reducing the size of the relocatable library allows expansion of the core image library. The expanded core image library allows a greater number of components to be contained in a single systems volume.

8. Copy and restore programs are necessary to transfer the resident system from tape to 2311/2314, 2311/2314 to tape, 2311/2314 to cards, and cards to 2311/2314 for maintenance and backup purposes.

9. The procedures for the configurations shown in this publication assume the system packs to be initialized with the VTOC on cylinder 199 and the work packs to be initialized with the VTOC on cylinder zero or 199.

MAPS AND LISTINGS PRODUCED DURING SYSTEM GENERATION

All linkage editor output on SYSLST from the system generation procedure and any future updates (including maps produced by the linkage editor) must be retained. These maps provide necessary information on the level of the system and the load address (relocation) of each component. Similarly, supervisor assembly listings should be retained. These maps and listings will be used by the systems programmer and the Field Engineer maintaining the system.

VARIANTS OF IBM-SUPPLIED ASSEMBLER

The relocatable library of the IBM-supplied system residence contains assembler modules suitable for building an Assembler using Tape Work Files (TWF), or Disk Work Files (DWF), or both tape and disk workfiles. The following are the names of the IBM-supplied assembler variants:

IJQD16TW (10K assembler, TWF)
IJQD16DW (10K assembler, DWF)
IJQD32 (14K assembler)
IJYASM (44K assembler)

The first two variants require a minimum of 10K bytes of contiguous problem program storage for use by the assembler and can be used with a minimum machine size of 16K. The third variant of the assembler listed can be built to use either tape and/or disk work files and use additional problem program storage for buffering of work file input/output functions. The tape and/or disk work file variant requires a minimum of 14K bytes of contiguous problem storage for use by the assembler and requires a minimum machine size of 24K. A second assembler, Assembler F, is also available. This assembler requires a minimum of 44K bytes of contiguous storage and requires a minimum machine size of 64K. Either the 14K variant or the 44K Assembler must be used when using private source statement libraries.

The names of the 10K and 14K modules begin with IJQ and the 44K modules with IJY; thus modules can be copied, punched, or deleted by a COPYR, PUNCH, or DELETR statement specifying IJQ.ALL or IJY.ALL as an operand. To linkage edit the TWF only variant of the assembler, use the following job control statements:

```
   INCLUDE IJQD16TW
// EXEC    LNKEDT
```

To linkage edit the DWF only variant, use the following job control statements:

```
   INCLUDE IJQD16DW
// EXEC    LNKEDT
```

To linkage edit the 14K variant, use the following job control statements:

```
   INCLUDE IJQD32
// EXEC    LNKEDT
```

To linkage edit the 44K assembler, use the following job control statements:

```
   INCLUDE IJYASM
// EXEC    LNKEDT
```

If the 14K variant of the assembler is to be linkage edited to the core image library of a system containing a 10K variant, the assembler supplied in the core image library of the IBM-supplied pack should first be deleted (DELETC ASSE.ALL).

Only one of the IJQ variants may reside in the core image library at any one time. One of the IJQ variants can co-reside with the IJY assembler if the name of the first phase of one of them is changed from ASSEMBLY to some other name by means of the RENAMC function of the MAINT program. The renaming must be performed before the second assembler is linkage edited into the core image library. The first one is then invoked under its new name; the second under the name ASSEMBLY.

It is expedient, during system generation, to use the largest assembler the machine can support because the performance improvement is most significant.

VARIANTS OF PL/I

Two variants of the PL/I compiler can be built. One of the PL/I variants requires 10K bytes (of problem program storage), while the other requires 12K. The 12K variant allows the system input and output files to be assigned to a 2311 disk drive or a 2314 direct access storage facility, if the supervisor supports SYSFIL. PL/I is capable of using either disk or tape work files (as are COBOL, FORTRAN, Basic FORTRAN, and RPG). Thus, PL/I also furnishes compile-time device independence for work files.

PL/I AND SYSTEM INPUT/OUTPUT ON DISK

The two variants of the PL/I compiler differ in their treatment of SYSIPT, SYSLST, and SYSPCH. If one or more of these logical units is assigned to a 2311 during compilation, the variant of the PL/I compiler requiring 24K bytes of main storage (12K for the compiler itself), must be built.

The following statements will generate the 12K variant:

```
   INCLUDE IJXPLID
// EXEC    LNKEDT
```

If the 12K variant is not required, the following statements must be specified:

```
   INCLUDE IJXPLI
// EXEC    LNKEDT
```

If either SYSIPT or SYSLST is assigned to a 2311 or 2314 for execution of PL/I object programs, the I/O modules invoked to support these assignments must be retained. During system generation, a relocatable library maintenance run should be performed to rename these modules before deleting the PL/I compiler from the relocatable library. The following control statements delete the I/O modules that do not support the disk facility from the relocatable library (because they are no longer needed), rename the PL/I object time I/O modules that do support the disk facility, and delete the PL/I compiler from the relocatable library.

```
// EXEC MAINT
...
   DELETR IJKSYSA,IJKSYSI
   RENAMR IJXSYSA,IJKSYSA,IJXSYSI,IJKSYSI
   DELETR IJX.ALL
...
```

The user may want to retain the card modules that do not support the disk facility (if there is a possibility that card or tape input only and printer or tape output only are desired at some future time). The following control statements permit punching these modules. The user must punch these modules before they are deleted from the relocatable library as shown in the preceding example.

```
// EXEC RSERV
   PUNCH IJKSYSA,IJKSYSI
/*
```

If neither SYSIPT nor SYSLST is assigned to a 2311 or 2314 during execution of PL/I object programs, only the DELETR IJX.ALL statement is required for the maintenance run.

During system generation, utilities and work-file variants that are not needed can be deleted. If the system volume received by the user is a disk pack, it is capable of operating as a system. If the system volume is a tape reel, it is self-loading tape that must be restored onto a disk pack before it is operable.

FORTRAN Compatibility

The FORTRAN library contains a module, ILFACOM, to provide compatibility with Basic FORTRAN subprograms. Because of differences in design objectives, object modules produced by the FORTRAN and Basic FORTRAN compilers are not compatible without this interface. With it, subprograms compiled under Basic FORTRAN can be incorporated into a FORTRAN program.

The compatability module can be invoked on a job-by-job basis, or the system can be altered at system generation time so that it is brought in with all Basic FORTRAN linkage edits. The procedure for using it for a single job is discussed in the FORTRAN IV Programmer's Guide, listed in the Preface.

The system can be altered permanently by executing the jobstream book named Z.ILFMERGE which is in the source statement library of the DOS system residence volume supplied by IBM. This action deletes all Basic FORTRAN object time library routines from the relocatable library. It replaces them with references to the corresponding FORTRAN object time library routines and to ILFACOM. Subsequently, any Basic FORTRAN object module that refers to a library routine, such as the square root subprogram, IJTSSQT, actually uses the corresponding FORTRAN routine, in this case, ILFSSQRT.

This procedure is recommended only for installations that intend to convert entirely from Basic FORTRAN to FORTRAN, but do not want to recompile existing programs and subprograms. The action provides additional space in the relocatable library by eliminating the Basic FORTRAN routines, but it also increases the execution time of all Basic FORTRAN modules.

WARNING DIAGNOSTICS

The following warning diagnostics appear in the linkage editor maps during system generation, but they do not indicate errors.

Assembler

DWF variant (incorporated by including IJQD16DW for assembler linkage edit).

```
*UNREFERENCED SYMBOLS

EXTRN     IJQD0$21
EXTRN     IJQD0$24
EXTRN     IJQD0$39
EXTRN     IJQD0$60
EXTRN     IJQD2$30
EXTRN     IJQRTA35
EXTRN     IJQRTA45
EXTRN     IJQRTB30
EXTRN     IJQRTB42
EXTRN     IJQRTB57

POSSIBLE INVALID ENTRY POINT DUPLICATION IN INPUT
```

TWF variant (incorporated by including IJQD16TW for the assembler linkage edit).

```
*UNREFERENCED SYMBOLS

EXTRN     IJQD0$15
EXTRN     IJQD0$18
EXTRN     IJQD0$58
EXTRN     IJQD0A10
EXTRN     IJQD0$60
EXTRN     IJQRTA35
EXTRN     IJQRTA45
EXTRN     IJQRTB30
EXTRN     IJQRTB42
EXTRN     IJQRTB57
EXTRN     IJQD2$30
EXTRN     IJQD2$60
```

POSSIBLE INVALID ENTRY POINT DUPLICATION IN INPUT

Variant utilizing additional main storage (incorporated by including IJQD32 for the assembler linkage edit).

```
*UNREFERENCED SYMBOLS

EXTRN     IJQD0$45
EXTRN     IJQD0$57
EXTRN     IJQD0$58
EXTRN     IJQD0A10
EXTRN     IJQRTA30
EXTRN     IJQRTB39
EXTRN     IJQRTB54
EXTRN     IJQD2$30
EXTRN     IJQD2$60

POSSIBLE INVALID ENTRY POINT DUPLICATION IN INPUT
```

System Generation and Maintenance Procedure

Many techniques exist for generating and maintaining operational volumes. Each installation uses techniques dependent on its machine configuration and its selection of system and application programs. The techniques described here correspond to the following principal machine configurations relevant to system generation and maintenance:

1. At least two 2311 disk drives.

2. One 2311 disk drive.

3. One 2314 direct access storage facility.

The three examples given (Figures 8, 10, and 12) do not show the coding necessary to linkage edit and delete all IBM-supplied components. They are meant only to be samples and must be tailored to meet each user's needs. Additional control statements required to meet the user's needs for linkage editing and deleting any IBM-supplied components are given in Appendix C.

If the system has a tape unit (7-track tapes must have the Data Conversion feature to be used during system generation and maintenance), the system is shipped from IBM on tape. The IBM-supplied tape is a self-loading tape that must be transferred from tape to a disk pack before starting system generation. The resulting disk pack is then ready for system generation. The IBM-supplied tape should be retained as an additional backup tape. If a system has no tapes, the 2311 resident system is shipped on a disk pack that is ready for system generation. The 2314 resident system is shipped on tape(s) only.

Each System Generation Job begins with a new IPL procedure, including the necessary ADD, SET, and ASSGN statements. Typically, each job consists of many job steps, including such librarian programs as:

- CSERV (core image library service) to punch out (or write on magnetic tape or disk) programs from the core image library during maintenance.

- SSERV (source statement library service) to punch out (or write on magnetic tape or disk) macro definitions.

- RSERV (relocatable library service) to punch out (or write on magnetic tape or disk) the relocatable modules used to build IBM-supplied processor programs.

- DSERV (directory service) to display on SYSLST the current contents of one or more library directories and their remaining library capacities.

- MAINT (library maintenance) to delete and/or catalog library elements, and also to condense and reallocate (2311) library extents.

- CORGZ (copy or merge and organize) to selectively copy or merge one disk pack onto another disk pack, with the option of allowing larger/smaller allocations for each library of the new pack.

The sequence of job steps depends on the configuration available and the operational packs each user is building. Certain activities are common to many users:

1. The general system generation procedure is:

a. Initialize the disk pack to contain the system.

b. For disk and tape users, restore the IBM-supplied tape onto disk. For two disk drives and no tape drives, copy the IBM-supplied disk to another disk to obtain backup.

c. Retrieve sample problems.

d. Delete unwanted programs from the system (all libraries).

e. Allocate library sizes required.

f. Set standard labels for SYSLNK, SYS001, SYS002, and SYS003 if desired.

g. Assemble supervisor into cards.

h. Linkage edit and catalog supervisor and IBM components.

i. Assemble user IOCS modules.

j. Catalog IOCS modules to the relocatable library.

k. Delete supervisor and IOCS macros if not desired.

2. The supervisor generation macro instructions must be keypunched to form a single source deck.

3. DELETR cards are selected or prepared for each relocatable library component that the user does not need in his system.

   On a system with at least two disk drives, the user can copy selectively rather then delete (DELETR). To copy selectively, prepare the librarian cards in the form: COPYR IJx.ALL.

4. DELETS A.xxxxxxxx cards are selected or prepared for each macro definition that the user does not need in his system. The list of IBM-supplied macro definitions appears in Appendix C. The user may wish to retain certain low-usage macro definitions, such as supervisor generation macros, only on his system maintenance pack, rather than on his operational pack(s). The books, Z.DELETECL, Z.DELETERL, Z.DELETESL and Z.LINKEDIT, can be retrieved through SSERV. These books contain the necessary statements for the user to delete or linkage edit selectively any components from the system. A pause card is read immediately before a deletion or linkage-edit job is performed. This allows the user to enter EOB to perform the job or to type in CANCEL at the 1052 printer keyboard to bypass that deletion or linkage edit. If a user does not wish to go through the complete book performing selective jobs, he can choose only those cards he needs from the book.

5. After a component is linkage edited into the core image library of a single drive system, the user can free a large extent on any operational pack by deleting the modules from the relocatable library used to build the component. The delete book Z.DELETERL can be used to perform this step. However, the user must then rebuild the relocatable library of the pack prior to updating a component, by entering the appropriate modules either from cards, magnetic tape, or disk. On a system without magnetic tape this tradeoff must be carefully considered: disk tracks made available vs. speed and simplicity of component maintenance.

6. To use the COBOL, PL/I, and RPG languages, the user must ensure that a certain collection of IOCS modules are available in the relocatable library of each operational pack. These modules are preassembled and

supplied in the relocatable library. Certain of these IOCS modules are linkage edited into each COBOL, PL/I or RPG object program. These modules are generated using the following macro definitions supplied by IBM:

CDMOD	Card Reader/Punch
PRMOD	Printer
MTMOD	Magnetic Tape
SDMODxx	Sequential DASD. SDMOD consists of ten similar macro definitions: SDMODFI for sequential disk with fixed input, SDMODFO for sequential disk with fixed output, etc. A complete description of SDMODxx is contained in the <u>Supervisor and Input/Output Macros</u> publication.
ISMOD	Indexed Sequential DASD
DAMOD	Direct Access Method, DASD
DIMOD	Device Independent Module

Assembler-language users can assemble these IOCS functions directly into their application programs, or the IOCS modules can be assembled separately and cataloged into the system relocatable library. The assembler also supports PTMOD, ORMOD, and MRMOD. Separate assembly of IOCS modules requires no additional main storage or additional execution overhead in speed. Those modules, shipped pre-assembled for IBM components, can also be used by any other program, if applicable. A separate assembly is preferable for most users because:

- Program assembly and reassembly time is minimized.
- Many IOCS modules will already have been assembled for COBOL, PL/I and RPG programs.
- Use of preassembled IOCS modules facilitates program maintenance and standardization.
- The xxMOD macro definitions just cited require a substantial number of cylinders in the source statement library.

The corresponding generated modules ordinarily require fewer cylinders in the relocatable library. Thus, users may prefer to retain xxMOD macro definitions only on the system backup volume, cataloging a selection of generated modules onto each operational volume.

During each system generation and maintenance procedure, system libraries are periodically copied onto magnetic-tape reels, disks, or cards to provide backup in case of subsequent specification errors or machine errors. Users may omit these backup procedures, but the indicated maintenance procedures and any additional precautionary procedures desired by the user should be followed.

The following IBM System/360 Basic Programming Support Utility Programs <u>may be</u> required for system generation and should be ordered with the initial distribution volume, depending on the configuration of the system. Each BPS utility must be loaded from cards by a separate IPL procedure.

- Distribution Program 360P-UT-208
- Initialize Disk 360P-UT-206
- Copy Disk-to-Tape Restore Tape-to-Disk 360P-UT-061
- Copy Disk-to-Card Restore Card-to-Disk 360P-UT-062
  - Initialize Tape 360P-UT-057
  - Universal Character Set 360P-UT-048
  - Initialize Data Cell 360P-UT-204

8. Users are encouraged to perform a DSERV or check a system directory printout to determine the contents and sizes of the system libraries during system generation. This procedure enables the user to determine that enough blocks remain for linkage edit and catalog procedures. A DSERV requires the following control statements:

```
// JOB DSERV
// EXEC DSERV
   DSPLY ALL
/*
/&
```

   A printout of the system directory is provided automatically following a linkage edit with an OPTION CATAL specified or any // EXEC MAINT.

9. The initial system volume from IBM contains the volume serial number 111111.

   If the IBM-supplied volume is a tape, the user must initialize the disk pack with his volume serial number before restoring the tape to the disk. The standard labels (DLBL and EXTENT statements) shown for the examples are adequate for system generation. However, the EXTENT statement must be adjusted to reflect the user's volume serial number.

   To use the standard labels for configurations with two disks, SYSLNK, SYS001, SYS002, and SYS003 can be assigned to the second disk. The standard labels shown in the following discussions assume the VTOC to be on Cylinder 199 of the residence volume. Following system generation, standard label assignments should be set that will be adequate to support the user's installation requirements.

   The following rules must be observed when extents are assigned to Autotest work files:

   - The extents for IJSYS01, IJSYSAT, and IJSYSLN must always be unique.
   - The extent for IJSYSAT and IJSYSLN must be on the same device but need not be contiguous.
   - The extents for IJSYSAT may be the same as those for IJSYS02 or IJSYS03.
   - When the job is to compile-and-execute, the extents for IJSYSLN must be different from those for IJSYS02 or IJSYS03.

10. Standard labels (OPTION STDLABEL) have been defined on the system distributed by IBM for SYSLNK and SYS001. Appendix G discusses the format of the supplied system volume.

A reply of delete to the following message will destroy the system residence file unless it is encountered during an allocate (ALLOC) run:

```
DOS SYSTEM RESIDENCE FILE
4444A OVERLAP ON UNEXPIRED FILE
```

The core image library allocations on the IBM-supplied volume(s) are not sufficient to contain all of the system components on a 2311. The adequacy of allocations can be determined through the use of Appendix G. Refer to Appendix C for core image phase names, relocatable module names, and source statement macro names. Phases, modules, and macros are identified by component.

Also included in Appendix C are those statements required to linkage edit and delete each component.

11. If the installation-tailored supervisor does not exceed the SEND address of the IBM-supplied supervisor, linkage editor, librarian, and assembler do not need to be linkage edited and cataloged again to the core image library.

IBM BPS Utility Programs

At the end of system generation, the user is advised to copy the system volume for operational volume backup. To restore the volumes to a disk pack, the IBM BPS copy and restore or the distribution program utility programs should be used. The user should be familiar with the BPS utility publications as listed on the front cover of this publication. The following control information must be inserted in the program deck to operate these programs:

<u>Note</u>: The channels used by the BPS supervisors are:

Multiplexor (channel 0)
Selector 1
Selector 2

Copy and Restore

- The following job control cards must be placed in the program deck before the first card containing a D in Column 73.

<u>Copy Disk-to-Tape or Card Utility Programs</u>

```
            {CD}
// JOB CDSK {TP}
// VOL SYSIPT,UIN
// DLAB (must describe the format one label for SYSRES in the
   SYSRES VTOC)
// XTENT
// CONFG 001
// ASSGN SYSIPT,X'cuu',dd
// ASSGN SYSOPT,X'cuu',dd,X'ss'
// ASSGN SYSRDR,X'cuu',dd,X'ss'
// ASSGN SYSLST,X'cuu',dd
// ASSGN SYSLOG,X'cuu',dd
// DATE  68110
// EXEC
```

<u>Restore Card or Tape-to-Disk Utility Programs</u>

```
         {CD}
// JOB R {TP} DSK
// CONFG 001
// ASSGN SYSIPT,X'cuu',dd,X'ss'
// ASSGN SYSOPT,X'cuu',dd
// ASSGN SYSLST,X'cuu',dd
// ASSGN SYSRDR,X'cuu',dd
// ASSGN SYSLOG,X'cuu',dd
// DATE  68110
// EXEC
```

<u>Note</u>: For the BPS description (dd) parameter see <u>Appendix F</u>.

- The following utility modifier card must be placed in the program deck before the first card containing an E in Column 73. This card is used only for the copy programs and must be omitted from the program deck for the restore program.

```
//bUCRbTF,A=(1600),N=(1),OL,IPL,LOG
```

The Copy Disk to Card and Restore Card to Disk programs have the ability to be restarted if required. Refer to the BPS DASD utility publication referenced on the front cover of this publication.

Note: In this control card b indicates one blank space.

The following operating procedures are required for running the IBM-supplied utility object programs:

1. Ready the necessary I/O units.
2. Set the console switches as follows:
   a. ROS CONTROL, RATE, ADDRESS COMPARE, and CHECK CONTROL TO PROCESS.
   b. The three Load-Unit switches to the address of the card reader used for loading the program.
3. Loading of the utility program.
   a. Check the preceding units and settings; then press the System-Reset key.
   b. Place the object deck with control cards in the reader hopper.
   c. When the reader unit is ready, press the console Load key.

Distribution Program 360P-UT-208

The distribution program can be used to copy a DOS system to tape for backup. The resulting tape is a self loading tape that is similar to the IBM-supplied system tape. The only difference in the tapes is in the contents of the systems. The tape produced by this program has the following format:

- IPL
- Initialize disk program
- Tapemark
- IPL
- Restore program
- File identification record
- File label information
- Disk file (DOS system)
- Tapemark

When restored to a disk pack, this tape will not only restore the system as it was copied, but will also initialize the pack to which it is being copied. The instructions for restoring the system to a disk are the same instructions for restoring the IBM-supplied system to a disk pack before performing system generation.

COPYING A SYSTEM TAPE TO DISK

To copy a DOS system to tape:

- Mount the DOS system and a tape.

- The following job control cards must be placed in the program deck between the cards containing ID numbers C208 and D208 in columns 73-76:

- Place the deck in the card reader.

```
// JOB DISCPY
// DATE  68110
// ASSGN SYSLOG,X'cuu',dd
// ASSGN SYSIST,X'cuu',dd
// ASSGN SYS000,X'cuu',dd (disk)
// ASSGN SYS001,X'cuu',dd[,X'ss'](tape)
// CONFG 001 (optional, 16K assumed)
// EXEC
```

Note: For the BPS description parameter (dd) see Appendix F.

- The following utility modifier card must be placed immediately following the program deck:

  //bUDSb'field one of format 1 DASD file label 44-characters' Col.53

- Dial on the console the address of the card reader.

- Check the units and settings; then press the System-Reset key.

- Press the console load key.

- When the Wait light comes on, press Start and EOF on the card reader.

RESTORING A SYSTEM TAPE TO DISK

The IBM-supplied system residence tape must be copied onto a disk pack before system generation can be performed. The pack that is to contain the system must be initialized with a volume label and a volume table of contents (VTOC). If the disk pack is not initialized or if the VTOC is not at cylinder 199, the pack must be initialized.

INITIALIZE DISK

The following instructions and control cards will allow the user to properly initialize his disk:

- Mount the distribution tape.

- Place the following control cards in the card reader, in the sequence shown:

```
Card Col. 1

Card 1        // JOB INTDSK

Card 2        // DATE yyddd  yy = 00 to 99 decimal
                            ddd = 001 to 366 decimal

Card 3        // ASSGN SYSOPT,X'cuu',Dd
                            cuu = channel and unit of disk drive
                             Dd = D1 for 2311 or D3 for 2314

Card 4        // ASSGN SYSLOG,X'cuu',C1
                            cuu = channel unit 1052

Card 5        // EXEC

Card 6        // UIDbIR,C1

Card 7        // VTOC STRTADR=(0199000),EXTENT=(y)
                            y = number of tracks allotted
                                to VTOC in decimal (1-10).

Card 8        VOL1nnnnnn    nnnnnn = Volume serial number
                            Columns 42-51 are reserved for user's
                            identification.

Card 9        // END
```

- Dial on the console the address of the tape unit containing the distribution tape into the CPU load address switches.
- Press Load.
- When the Wait light comes on, press Start and EOF on the card reader.
- The message EOJ will be printed on the 1052 printer-keyboard when initialization is complete. The system can then be copied on the disk.

BYPASS INITIALIZE DISK

If the disk has previously been properly initialized, the Initialize Disk procedure can be omitted and the following procedure performed to bypass the initialize disk routine:

- Mount and ready the distribution tape.
- Place the following control cards in the card reader in the sequence shown:

```
Card Col. 1

Card 1        // JOB INTDSK

Card 2        // DATE yyddd  yy = 00 to 99 decimal
                            ddd = 001 to 366 decimal

Card 3        // ASSGN SYSLOG,X'cuu' ,C1

Card 4        // LOG
```

```
Card 5        // ASSGN SYSIPT,X'cuu',Tz[,X'90']
                           cuu = channel and unit
                             z = 2 for 9-track tape
                             z = 1 for 7-track tape
                                 The X'90' entry is
                                 required if z = 1.

Card 6        // FILES SYSIPT,1
```

- Dial the address of the tape drive containing the distribution tape in the CPU load address switches.
- Press Load.
- When the Wait light comes on, ready the card reader containing the control cards by pressing Start and EOF.
- The tape will forward-space past the initialization program and the following message will appear on SYSLOG:

  000C
  4000A

RESTORE A SYSTEM TAPE TO DISK

Following initialization or bypassing initialization of the disk pack, the IBM-supplied system tape can be restored to disk by this procedure:

- Do not rewind the distribution tape.
- Clear the card reader.
- Place the following control cards in the card reader:

```
Card Col. 1 ──┐
              ↓
Card 1        // JOB DISRST

Card 2        // DATE yyddd  yy = 00 to 99 decimal
                            ddd = 001 to 366 decimal

Card 3        // ASSGN SYS000,X'cuu',Dd
                            cuu = channel and unit of
                                  the initialized disk
                             Dd = D1 for 2311 or D3 for 2314

Card 4        // ASSGN SYSLST,X'cuu',L1

Card 5        // ASSGN SYSLOG,X'cuu',C1

Card 6        // EXEC
```

- IPL from the distribution tape and press Load on the console.
- When the Wait light comes on, ready the card reader by pressing Start and EOF.
- The following message appears in SYSLOG:

  4444A

  Type in 4 B, and press INTERRUPT to continue.

- If the message END OF VOLUME ON SYSIPT 3777A occurs while restoring a multivolume file, ready the next reel on SYSIPT reply 22 and press interrupt to continue. This message will not be encountered if the second volume of a two-volume system distribution is mounted on an alternate drive assigned to SYS002.

```
// ASSGN SYS002,X'cuu',TZ
```

- When the job is finished, the following message is printed on the IBM 1052 Printer-Keyboard and the disk pack is ready for system generation:

```
DOS SYSTEM RESIDENCE FILE
nnnnnn RECORDS RESTORED FOR ABOVE FILE
3007
END OF JOB
3008
```

Macro Instructions for Generating a Supervisor

Using the IBM-supplied volume, each installation normally generates a more efficient supervisor using the supervisor generation macro instructions. The macro instructions describe the machine configuration, standard I/O assignments, and standard processing options.

This section defines the ten macro instructions and their parameters required to generate an installation-tailored supervisor for the Disk Operating System. Figure 4 is a consolidated list of the supervisor generation macros discussed.

RULES FOR USING SUPERVISOR GENERATION MACROS

1. The assumed value for an omitted parameter is underlined in the following discussion and in Figure 4. Figure 5 shows device type codes to be used for system generation.
2. Material enclosed in braces { } indicates a programmer option. One of the enclosed values must be selected by the programmer.
3. Bracketed operands are optional, e.g., [n].
4. Replace the letter <u>n</u> in a parameter with a decimal number.
5. The name field must be blank. The operation field always contains the mnemonic operation code. The operand field contains the parameters.
6. Several parameters may be included on one line for all macros. Separate each parameter with a comma. No embedded blanks are permitted. Continuation cards are permitted (nonblank character in column 72; the continue column is column 16).
7. In the expression X'cuu', replace <u>cuu</u> with the hexadecimal number for channel and unit.
8. The macros must be issued in the following sequence: SUPVR, CONFG, STDJC, FOPT, PIOCS, ALLOC, IOTAB, DVCGEN, ASSGN, SEND.
9. The DVCGEN, ASSGN, and ALLOC macros are not required. They are specified if input/output tables (DVCGEN) are being specified, if standard assignments (ASSGN) are being made or if storage is allocated (ALLOC) for MPS at system generation time.
10. An END card and a /* card must follow the SEND macro instruction.

<u>SUPERVISOR</u>

SUPVR This macro instruction and its parameters define the system as disk resident and its ability to perform multiprogramming, MICR or Teleprocessing.

Parameters for SUPVR

SYSTEM=DISK — Always used with the SUPVR macro instruction. SYSTEM=DISK is assumed if this parameter is omitted.

MPS={NO | YES | BJF} — Specify if there is to be multiprogramming support. When YES or BJF is specified the system generated is capable of supporting two foreground programs. YES or BJF must be specified if TP=QTAM. When BJF is specified, batched job environment will be supported for both foreground partitions. Multiple communication regions are generated only if BJF is specified. MPS=YES is implied if MPS=BJF is supplied.

TP={NO | BTAM | QTAMn} — Specify if Teleprocessing support is desired and if so, whether Basic or Queued Access Method (BTAM or QTAM) is desired. When QTAM is specified, SVC support for BTAM is also included. n is the maximum number of QTAM message processing programs in the system at one time. n may have any value from 2 to 12.

MICR={NO | 1412 | 1419 | 1419D} — Specify if the supervisor is to support magnetic ink character readers. If both 1412's and 1419's are present, indicate 1419. If 1259's are to be supported, also indicate 1419. 1419D indicates Dual Address Adapter 1419's. If 1412/1419's are attached to the multiplexor channel, the PIOCS parameter BMPX=YES is not supported. 1419 support gives 1259 capability.

AP={NO | YES} — Specify if there is to be multitasking support. Multitasking allows the execution of more than one program within a partition. MPS=YES and WAITM=YES are assumed if AP=YES.

EU={NO | YES} — Specify if the IBM 1401/1440/1460 Emulator program is to be executed. The MODEL operand in the CONFG macro can be MODEL=25, or MODEL=30 for IBM Model 2025, and it must be MODEL=30 and MODEL=40 for IBM Models 2030 and 2040, respectively.

CONFIGURATION

CONFG — This macro instruction and its parameters define the system configuration and can be used to specify generation of optional supervisor services. If the assumed options are all satisfactory, the only entry required is the CONFG macro itself without any parameters.

Parameters for CONFG

MODEL={30, nn} Specify the model number of the system (nn=30, 40, etc). If MICR support is required on a Model 65, MODEL=65 should be indicated for maximum performance. If 1401/1440/1460 emulator program support is desired for the:

- IBM Model 2025, specify MODEL=25 or MODEL=30.
- IBM Model 2030, specify MODEL=30.
- IBM Model 2040, specify MODEL=40.

SP={NO, YES} Specify if the storage protection feature is available to the system. YES is assumed if the MPS=YES or MPS=BJF parameter is specified in the SUPVR macro.

DEC={NO, YES} Specify if the decimal feature is present.

FP={NO, YES} Specify only if the floating point feature is present.

TIMER={NO, YES} Specify if the timer feature is present. If YES is specified, GETIME support is provided.

JOB CONTROL OPTIONS

STDJC This macro instruction and its parameters specify the standard settings for job control. If the assumed options are all satisfactory, the only entry required is the STDJC macro instruction itself, without any parameters. These standard options can be locally overridden by an OPTION statement.

Parameters for STDJC

DECK={YES, NO} Specify if language translators are to output object modules on SYSPCH.

LIST={YES, NO} Specify if language translators are to write source module listings and diagnostics on SYSLST.

LISTX={NO, YES} Specify if compilers are to write hexadecimal object module listings on SYSLST.

SYM={NO, YES} Specify if assembler is to output symbol tables on SYSPCH. SYM=YES must be specified if the PL/I compiler is to produce a symbol and offset table listing.

XREF={YES, NO} Specify if assembler is to write symbolic cross-reference lists on SYSLST.

ERRS= {YES | NO} Specify if compilers are to summarize all errors in source programs on SYSLST. Assembler and PL/I always assume ERRS=YES.

CHARSET= {48C | 60C} Specify either the 48- or 60-character set for PL/I translator input on SYSIPT.

LOG= {YES | NO} Specify for a listing of all control statements on SYSLST. LOG=NO suppresses the listing of all job control statements on SYSLST.

DUMP= {YES | NO} Specify if a dump of the registers and main storage is to be written on SYSLST in the event of an abnormal program end, cancel, or program check.

LINES= {56 | nn} Specify the number of lines per page on SYSLST. The minimum is 30; the maximum is 99.

DATE= {MDY | DMY} Specify the format of the date MDY=month/day/year. DMY=day/month/year.

OPTIONAL FEATURES IN THE SUPERVISOR

FOPT This macro and its parameters specify additional optional features that can be included in the supervisor.

Parameters for FOPT

OC= {NO | YES} Specify if external interruptions (except timer) can be handled by problem programs. If YES is specified, the facility is available to all programs in MPS. The Tape Compare Utility program requires OC=YES. YES is required if emulator program operator services are to be requested through the INTERRUPT key.

IT= {NO | BG | F1 | F2} Specify if the interval timer can be handled by problem programs. When interval timer facilities are available (i.e., IT=BG, F1 or F2) TIMER=YES is assumed for the CONFG macro. Timer support is available to only one program in MPS. QTAM requires F1.

PC= {NO | YES} Specify if a user program check routine will be used. Included in supplied supervisor because QTAM, FORTRAN, Basic FORTRAN, COBOL, PL/I, RPG, and Autotest require PC=YES.

TEB= {NO | n} Specify if tape error statistics are to be accumulated and logged where n is the number of tape drives and/or tape cartridge readers attached to the system. Allow extra TEBs for possible future expansion of the system. The maximum number is 254.

SKSEP= {NO, YES, n} Specify if SEEKs are to be separated from the remainder of channel programs. Seek separation allows other devices on the channel to be accessed (including other seeks) during the seek. YES indicates support for all DASD type devices specified by the DVCGEN macro at system generation time. n is the number of DASD devices to be supported and cannot be less than the number of DASD devices specified at system generation. The maximum is 254.

CE= {NO, YES, n} Specify the number of bytes to be allocated to the customer engineer serviceability routines. The minimum number of bytes that can be specified for n is 600. For the actual number of bytes allocated see Appendix G.

PTO= {NO, YES} Specify if the Physical Transient Overlap (PTO) feature is to allow the system to select tasks while Fetch is reading a fetched (or loaded) phase from the system residence file, or while I/O operations are being performed during Error Recovery Procedures (ERP). This overlap allows increased throughput. MPS=YES or BJF must be specified.

CBF= {NO, n} Specify I/O requests are to be buffered under the following conditions:

1. The actual record to be written does not exceed 80 characters.

2. Data or command chaining is not performed.

3. The CCB associated with this operation does not indicate the acceptance of unrecoverable I/O errors, posting at device end, or user error routines.

4. The CCB does not request sense information.

Console buffering allows overlap of CPU processing with write operations to SYSLOG by satisfying the requestor's WAIT immediately, rather than at I/O completion time. When this option is selected, the number of CHANQ entries should exceed the number of CBF entries so that the buffering process is not bound by the number of CHANQ entries. If the assumed option is taken for CHANQ and CBF is selected, the number of CHANQ entries will be assumed to be six more than the CBF entry. Numbers 1 through 9 are valid, and one is assumed if the operand is invalid.

CCHAIN= {NO, YES} Specify if command chaining support for retry on I/O operations is to be handled. When command chaining and an error occurs, CCHAIN allows the user to retry at the last CCW executed. Normal retry would return to the first CCW in the channel program. To make use of this option the user must have the appropriate bit set in the CCB. CCHAIN=YES must be specified if data or command chaining of IBM 2495 Tape Cartridge Reader input is performed.

TRKHLD=n Specify if the track hold function is to be supported. When processing sequential disk workfiles or updates to direct access files, specify if a hold is to be placed on the track of the record being read. The hold prevents anyone else that is using track hold from accessing that track. The maximum number of tracks that can be held at one time is 255. The default is 10 if n is an invalid parameter (nonnumeric or outside the range 1-255). MPS=YES or BJF must be specified if TRKHLD=n.

AB=$\left\{\begin{array}{l}\underline{\text{NO}}\\ \text{YES}\end{array}\right\}$ Specify if the abnormal termination exit function is to be supported. The abnormal termination exit allows the user to exit to a user's routine before an abnormal end of job causes a program to be cancelled.

WAITM=$\left\{\begin{array}{l}\underline{\text{NO}}\\ \text{YES}\end{array}\right\}$ Specify if the multiple wait function is to be supported. This function allows the user to use the WAITM macro to wait for one of a number of events to occur. WAITM=YES is assumed if AP=YES.

*DASDFP=$\left\{\begin{array}{l}\underline{\text{NO}}\\ (\text{n,n,}\left\{\begin{array}{l}2311\\2314\\2321\end{array}\right\})\end{array}\right\}$

Specify if supervisory DASD file protection is handled where n,n indicates the range of channels to which DASDs may be attached. Either 2311 or 2314 indicates file protection for both and 2321 indicates file protection for 2311, 2314, and 2321. If 2311, 2314, or 2321 are omitted, both 2311 and 2314 are assumed. DASDFP prevents the user from writing outside the extents of his file in case of program error. Extents are protected to the nearest cylinder except for 2321, where they are protected to the nearest head bank. This feature does not protect the file from being overwritten.

*SYSFIL=$\left\{\begin{array}{l}\underline{\text{NO}}\\ (\left\{\begin{array}{l}2311\\2314\end{array}\right\}[,n_1,n_2])\end{array}\right\}$

Specify if system input and system output (SYSRDR, SYSIPT, SYSLST, SYSPCH) files may be assigned to a 2311 or 2314. Specification of either gives support for both. If MPS=BFJ in the SUPVR macro, this parameter supports foreground logical units when running in batched mode. If the emulator program parameter SYSIO=222 or SYSIO=333 is indicated, a value must be specified for SYSFIL.

n_1 = residual capacity for beginning of operator notification when SYSLST assigned to disk. $100 \leq n_1 \leq 65535$. If n_1 is omitted, 1000 is assumed.

n_2 = residual capacity for beginning of operator notification when SYSPCH assigned to disk.

$100 \leq n_2 \leq 65535$. If n_2 is omitted, 1000 is assumed.

*Valid when 24K bytes of main storage are available. (See PL/I variant for the conditions governing logical units assigned to a disk.)

Note: Information on the macro instructions STXIT, EXIT, and SETIME is contained in the Supervisor and Input/Output Macros publication.

PHYSICAL IOCS

PIOCS This macro instruction and its parameters define the configuration requirements to be supported by physical IOCS. If the assumed options are all satisfactory, the only entry required is the PIOCS macro itself, without any parameters.

Parameters for PIOCS

SELCH={YES | NO} Specify if selector channels are attached to the system.

BMPX={NO | YES} Specify if burst mode devices will be supported on multiplexor channel. If YES is specified, unbuffered devices will not multiplex. If 1412/1419s are attached to multiplexor channel, BMPX=YES is not supported.

CHANSW={NO | RWTAU | TSWTCH} Specify if channel switching tape control unit, RWTAU = 2404 or 2804, TSWTCH = 2816. If either 2403 or 2803 and 2816 is specified, RWTAU must be specified. If a 2804 is specified, RWTAU must be specified. If a 2816 is specified, TSWTCH must be specified.

TAPE={NO | 9 | 7} Indicates required tape PIOCS support.

9 = nine track only.

7 = seven or nine track.

NO = No tape drives attached. This is the assumed value.

ALLOCATE

ALLOC F1=nK,F2=nK Specifies storage partitioning MPS, where n must be a multiple of 2. This macro is optional. Most IBM components require 10K bytes of background area. Foreground area must be a minimum of 10K to allow BJF processing.

INPUT/OUTPUT TABLES

IOTAB This macro instruction and its parameters define the area for the necessary device tables for the system. If the assumed options are all satisfactory the only entry required is the IOTAB macro itself without any parameters.

Parameters for IOTAB

BGPGR=$\left\{\begin{matrix}\underline{10}\\ n\end{matrix}\right\}$ Specify the number of logical unit blocks (LUBs) for programmer units, i.e., the number of symbolic programmer logical units (SYS000-SYSnnn). The minimum value generated is 10. The maximum value generated is 222.

F1PGR=$\left\{\begin{matrix}\underline{5}\\ n\end{matrix}\right\}$ Specify the number of symbolic units of the class SYSnnn for F1. Valid only for MPS. Otherwise zero is assumed. The maximum value is 222.

F2PGR=$\left\{\begin{matrix}\underline{5}\\ n\end{matrix}\right\}$ Specify the number of symbolic units of the class SYSnnn for F2. Valid only for MPS. Otherwise zero is assumed. The maximum value is 222.

JIB=$\left\{\begin{matrix}\underline{5}\\ n\end{matrix}\right\}$ Specify the number of job information blocks for the system (JIBs). One is required for each logical unit temporarily reassigned by a // ASSGN statement that differs from standard system assignment (i.e., established by the operator at IPL time). One JIB is required for each alternate logical unit assignment. One JIB is required for each open 2311 extent with the DASD file protect feature except for system input/output extents. Two JIBs are required for each open 2321 extent with the DASD file protect feature. The minimum value generated is 5. The maximum value generated is 255.

CHANQ= $\left\{\begin{matrix}\underline{6\ \text{or}\ 6+\text{CBF}} \\ n\end{matrix}\right\}$ Specify the number of entries in the channel queue. The minimum value generated is 6. If the assumed option is to be taken and CBF is to be specified, the assumed option will be six more than the CBF entry. The number of the channel queue should exceed the total number of I/O requests the user wishes to accumulate simultaneously. The start I/O commands for all channels are stored in this queue. The maximum value is 255.

IODEV= $\left\{\begin{matrix}\underline{10} \\ n\end{matrix}\right\}$ Specify the number of I/O devices attached to the system. The maximum is 255. The minimum value is 5.

Note: The sum of BGPGR + F1PGR + F2PGR LUB's must not exceed 222.

INPUT/OUTPUT UNITS

DVCGEN This macro instruction and its parameters define the physical input and output units attached to the system. This macro instruction is optional.

Parameters for DVCGEN

CHUN=X'cuu' Specify the hexadecimal number of the channel and unit for the device.

DVCTYP=xxxxxx Specify the device type. Figure 2 contains the codes for each IBM-supported device.

CHANSW= $\left\{\begin{matrix}\underline{\text{NO}} \\ \text{YES}\end{matrix}\right\}$ Specify if the device is attached to more than one selector channel. Indicates if the device can be switched (IBM 2816, 2804 or 2404 available).

MODE=X'ss'

1. 2400T9. MODE is used to specify the tape mode. X'C0' is the default value.

2. 2400T7. MODE is used to specify the tape mode. X'90' is the default value.

3. 2702. MODE designates the SADxxx command. X'00' is the default value.
   X'00' SAD0
   X'01' SAD1
   X'02' SAD2
   X'03' SAD3
   See Appendix A for other values of ss.

4. 2260 (Local). MODE is used to specify the 1053 printer when CHUN=X'cuu' refers to a 1053 attached to a 2848. The operand must be entered as MODE=X'01'.

5. 1412/1419/1259. MODE designates the external interrupt bit associated with magnetic ink character readers. The mode X'01' through X'20' correspond to external interrupt PSW' bits 31 through 26 respectively. For the dual address adapter 1419, this parameter is needed for both 1419P and 1419S.

   X'01' Device attached to external line 7.
   X'02' Device attached to external line 6.
   X'04' Device attached to external line 5.
   X'08' Device attached to external line 4.
   X'10' Device attached to external line 3.
   X'20' Device attached to external line 2.

Rules for Using DVCGEN

1. A separate DVCGEN macro instruction is required for each device. For a 2314, each individual unit needs a DVCGEN card.

2. The total number must not exceed the total number of devices specified in the IODEV parameter of the IOTAB macro.

3. DVCGEN macros must be specified in ascending channel address sequence.

4. Switchable units (attached to more than one selector channel) must be defined once on the lowest channel by which they are addressable.

5. The sequence of the DVCGEN cards determines the priority of the devices on their channel. SYSRES should be the first DVGEN card if it is to have the highest priority. Switchable units must be the last devices specified for each channel, and must be on consecutive channels.

6. The specifications of these macros can be altered at IPL by ADD and DEL statements.

7. IBM 1052 Printer-Keyboards that are not on-line but were defined by DVCGEN statements must be deleted by DEL statements when performing IPL from the card reader.

ASSIGNING SYMBOLIC DEVICE NAMES TO ACTUAL I/O DEVICES

ASSGN This macro instruction and its parameters assign symbolic device names (LUBs) to physical I/O devices (PUBs). A separate macro instruction is required for each symbolic device name with a standard system generation assignment. This macro instruction is optional.

Parameters for ASSGN

SYSnnn,X'cuu' Symbolic-name will be any of the system background logical units (SYSIPT, SYSLOG, etc) or programmer logical units (SYS000, SYS001, etc). X'cuu' is the hexadecimal number of the channel and unit to which the symbolic device is attached. A separate macro is required for each standard assignment desired. Programmer assignments are made only for background jobs.

System input/output units (SYSRDR, SYSIPT, SYSLST, or SYSPCH) that are assigned to a tape or DASD when the system is generated will be unassigned by IPL. An unassigned device can cause a job to be canceled.

END OF SUPERVISOR MACRO INSTRUCTIONS

SEND (n) This macro instruction indicates the end of the supervisor. n is a multiple of 8 and cannot be greater than 32,760.

Figure 2 illustrates protected and unprotected supervisor storage generated by various combinations of supervisor options that affect storage protection.

The supervisor consists of two parts:

- the nucleus that extends from the address SYSS00 to NUCEND, and
- the combined transient area, CE Serviceability Programs (CE=YES), and save area that extends from the address SYSEND to PPBEG (Figure 2A).

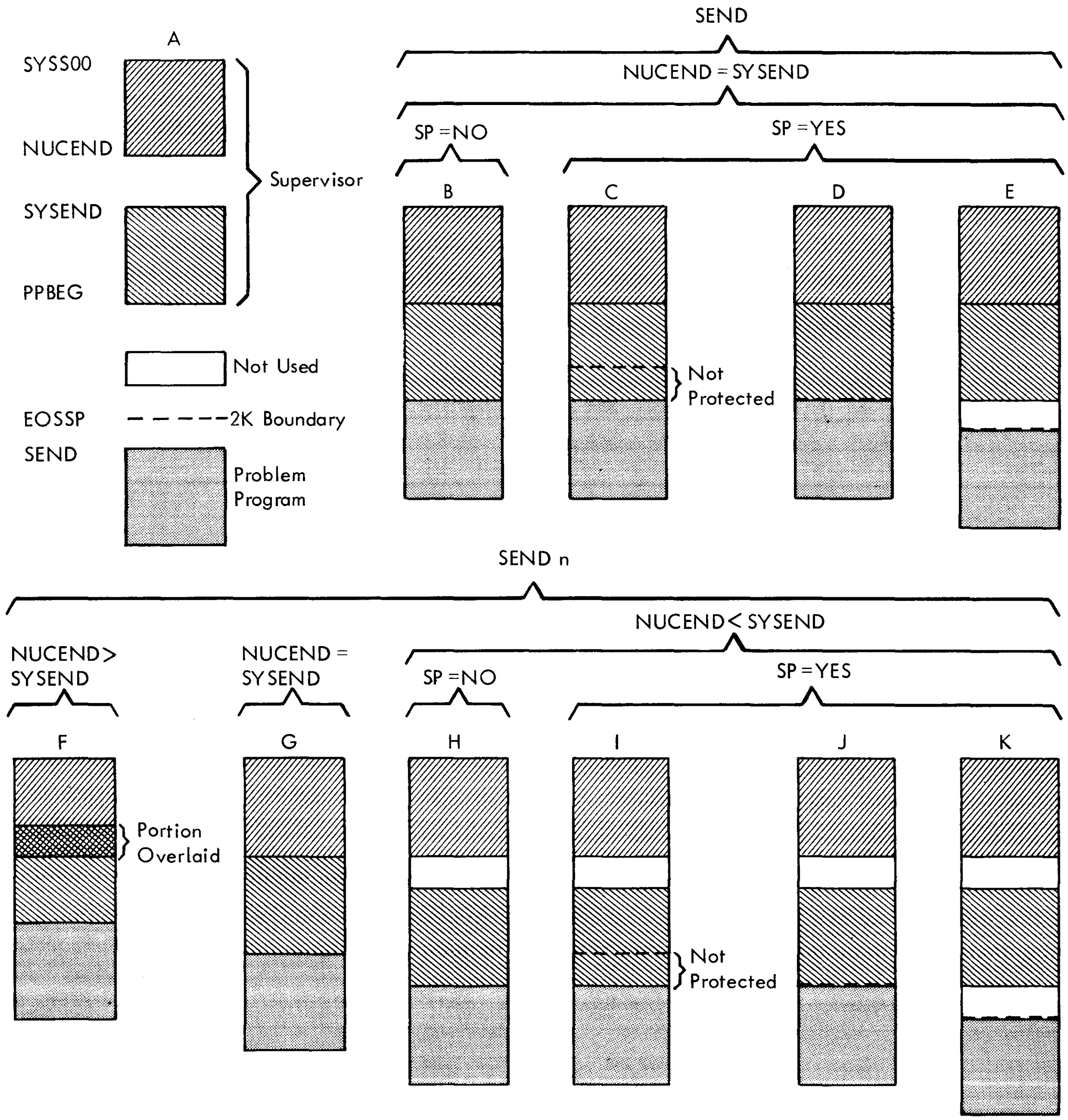


Figure 2. Unprotected Supervisor Storage Generated

SEND,SP=NO (Figure 2B)
If the operand n is not specified (Figure 2B), the address of NUCEND equals the address of SYSEND. Programs that are linkage edited to the end of the supervisor begin at PPBEG (Figure 2B). No space is provided for supervisor expansion, and growth in the supervisor will necessitate re-linkage editing user programs at the end of the supervisor.

SENDn,NUCEND>SYSEND (Figure 2F)/
SENDn,NUCEND=SYSEND (Figure 2G)/
SENDn,NUCEND<SYSEND,SP=NO (Figure 2H)

If the operand n is specified (Figure 2F-2H), the address of NUCEND can be greater than, equal to, or less than the address of SYSEND.

1. If NUCEND is greater than SYSEND (Figure 2F), a portion of the transient area overlays the supervisor nucleus. n must be increased, and the supervisor reassembled.

2. If NUCEND is equal to SYSEND (Figure 2G), the result is identical to the case where n was not specified (see Figure 2B) and the effect is also the same.

3. If NUCEND is less than SYSEND (Figure 2H), the area between SYSEND and NUCEND is not used and is available for future expansion of the supervisor. The difference between SYSEND and NUCEND is the number of bytes the supervisor may expand without having to re-linkage edit programs at the end of the supervisor.

SEND,SP=YES (Figures 2C, 2D, and 2E)
SENDn,NUCEND<SYSEND,SP=YES (Figures 2I, 2J, and 2K)

If Storage-Protect (SP=YES) is specified, the End-Of-Supervisor Storage-Protect (EOSSP) address is the first storage protect boundary after the SYSEND address when MPS=NO. When MPS=YES or BJF, the end of the supervisor will be the first storage protect boundary after the PTA address. The EOSSP address may be less than, equal to, or greater than the PPBEG address (end of supervisor).

1. If the EOSSP address is less than the PPBEG address, the area between these two addresses is not storage-protected. This will happen if the combination of MPS=NO and CE=YES was chosen (Figure 2I), and may happen if the:

   a. SEND address is not specified (Figure 2C).

   b. SEND address specified is not a multiple of 2048 bytes (Figure 2I).

2. If the EOSSP address is equal to the PPBEG address, the entire supervisor is storage-protected. This happens if the:

   a. SEND address is not specified and the PPBEG address is a multiple of 2048 bytes (Figure 2D), or

   b. SEND address specified is a multiple of 2048 bytes (Figure 2J).

3. If the EOSSP address is greater than the PPBEG address, the entire supervisor is storage-protected. The area between the PPBEG address and the EOSSP address is not used and is available for future expansion of the supervisor. The difference between the EOSSP address and the PPBEG address is the number of bytes the supervisor may expand without having to re-linkage edit programs at the end of the supervisor. This expansion area is in addition to any area between SYSEND and NUCEND. This may happen if the:

   a. SEND address is not specified (Figure 2E), or

   b. SEND address specified is not a multiple of 2048 bytes (Figure 2K).

MAXIMUM UNPROTECTED STORAGE
If SP=YES and a portion of the supervisor is not storage-protected (Figure 2C and 2I), the user should verify that the unprotected area falls within the following limits.

Figure 3 shows the maximum number of bytes in the area between the EOSSP address and the PPBEG address that may be unprotected.

	CE=NO	CE=YES	CE=n
MPS=NO PTO=NO FP=NO/YES	1832	2480	1880+n
MPS=YES/BJF PTO=NO FP=YES	664	1312	712+n
MPS=YES/BJF PTO=NO FP=NO	632	1280	680+n
MPS=YES/BJF PTO=YES FP=YES	120	760	160+n
MPS=YES/BJF PTO=YES FP=NO	88	728	128+n
MPS=NO PTO=YES FP=NO/YES	88	728	128+n

Figure 3. Maximum Unprotected Area between EOSSP and PPBEG (End of Supervisor) Address

Operation	Operand	Explanation
SUPVR		Supervisor macro instruction.
	SYSTEM=DISK	Specify a disk-resident supervisor. SYSTEM=DISK is assumed if this parameter is omitted.
	MPS= {NO / YES / BJF}	Specify if there is to be multiprogramming support. When YES or BJF is specified the system generated is capable of supporting two foreground programs. YES or BJF must be specified if TP=QTAM. BJF must be specified if batched job environment is desired for foreground partitions.
	TP= {NO / BTAM / QTAMn}	Specify if teleprocessing support is desired and if so, whether Basic or Queued Access Method (BTAM or QTAM) is desired. When QTAM is specified, SVC support for BTAM is also included. n is the maximum number of QTAM message processing programs in the system at one time. n may be any value 2 to 12.
	MICR= {NO / 1412 / 1419 / 1419D}	Indicates whether the supervisor is to support magnetic ink character readers. If 1412/1419's are attached to the multiplexor channel, the PIOCS parameter BMPX=YES is not supported.
	AP= {NO / YES}	Specify if there is to be multiprogramming within a partition (multitasking) support. Multiprogramming within a partition provides the ability to execute more than one program (multitasking) within a partition. MPS=YES and WAITM=YES are assumed if AP=YES.
	EU= {NO / YES}	Specify if the IBM 1401/1440/1460 Emulator program is to be executed. The MODEL operand in the CONFG macro can be MODEL=25, 30, or 40.
CONFG		Describes the hardware features.
	MODEL= {30 / nn}	Specify the model number.
	SP= {NO / YES}	Storage protection feature. YES must be specified for MPS or BJF.
	DEC= {NO / YES}	Decimal feature.
	FP= {NO / YES}	Floating point feature.
	TIMER= {NO / YES}	Timer feature.
STDJC		Specify standard settings for job control
	DECK= {YES / NO}	Output of object modules of language translators on SYSPCH.
	LIST= {YES / NO}	Source module listings and diagnostics from language translators on SYSLST.
	LISTX= {NO / YES}	Hexadecimal object module listings from PL/I and COBOL on SYSLST.
	SYM= {NO / YES}	Assembler and PL/I outputs symbol tables on SYSPCH: COBOL compiler outputs DATA DIVISION map on SYSLST.
	XREF= {YES / NO}	Assembler outputs symbolic cross reference lists on SYSLST.
	ERRS= {YES / NO}	COBOL, PL/I, FORTRAN, and Basic FORTRAN summarize all errors in source programs on SYSLST.
	CHARSET= {48C / 60C}	Specify the 48- or 60-character set for PL/I input on SYSIPT.

Figure 4. Macro Instructions for Supervisor Generation (Part 1 of 4)

Operation	Operand	Explanation
	LOG = {YES, NO} (YES underlined)	Listing of all control statements on SYSLST.
	DUMP = {YES, NO} (YES underlined)	Dump of registers and main storage on SYSLST.
	LINES = {56, nn} (56 underlined)	Number of lines per page on SYSLST.
	DATE = {MDY, DMY} (MDY underlined)	Format of the date.
FOPT		Specify optional support in the supervisor.
	OC = {NO, YES} (NO underlined)	STXIT option is available for external interrupt (except timer). YES is required for tape compare utility program.
	IT = {NO, BG, F1, F2} (NO underlined)	STXIT option is available for interval timer interruption in the area specified. TIMER=YES is assumed.
	PC = {NO, YES} (NO underlined)	STXIT option is available for program check interruption. Included in supplied supervisor because FORTRAN, COBOL, RPG, QTAM, PL/I and Autotest require PC=YES.
	TEB = {NO, n} (NO underlined)	Specify if tape error statistics are to be accumulated and logged where n is the number of tape drives attached to the system. Allow extra TEBs for possible future expansion of system.
	SKSEP = {NO, YES, n} (NO underlined)	Specify if SEEK's are to be separated from the remainder of channel programs. Seek separation allows other devices on the channel to be accessed (including other seeks) during the seek. YES indicates support for all DASD type devices specified by the DVCGEN macro at system generation time. n is the number of DASD devices to be supported and cannot be less than the number of DASD devices specified at system generation. The maximum number is 254.
	CE = {NO, YES, n} (NO underlined)	Specify the number of bytes to be allocated to the customer engineer serviceability programs. 600 is the minimum number of bytes that can be specified. For the actual number of bytes allocated, see Appendix G.
	PTO = {NO, YES} (NO underlined)	Specify if the physical transient overlap feature is to allow the system to select tasks while Fetch is reading a fetched (or loaded) phase from the system residence file, or while I/O operations are performed during error recovery procedures.
	CBF = {NO, n} (NO underlined)	Specify if I/O requests are to be appraised for console buffering and indicate the number of buffers (1 -9) to be generated.
	CCHAIN = {NO, YES} (NO underlined)	Specify if command chaining support for retry on I/O operations is to be handled.
	TRKHLD = {NO, n} (NO underlined)	When processing sequential disk workfiles or updates of direct access files, specify if a hold is to be placed on the track of the record being read. The hold prevents anyone else who is using track hold from accessing that track. The maximum number of tracks that can be held at one time is 255 and the assumed value is 10.
	AB = {NO, YES} (NO underlined)	Specify if the abnormal termination exit function is to be supported. The abnormal termination exit allows the user to exit to a user's routine before an abnormal end of job causes a program to be cancelled.
	WAITM = {NO, YES} (NO underlined)	Specify if the multiple wait function is to be supported. This function allows the user to use the WAITM macro to wait for one of a number of events to occur.
	*DASDFP = {NO, (n,n, {2311, 2314, 2321}) } (NO underlined)	Specify if supervisory DASD file protection is handled where n,n indicates the range of channels to which DASDs may be attached. Either 2311 or 2314 indicates file protection for 2311 and 2314. 2321 indicates file protection for 2311, 2314 and 2321.
	*SYSFIL = {NO, ({2311, 2314}[,n_1,n_2]) } (NO underlined)	Specify if system input and system output (SYRDR, SYSIPT, SYSLST, SYSPCH) files may be assigned to a 2311 or 2314. Specification for either gives support for both. n_1 = residual capacity for beginning of operator notification when SYSLST assigned to 2311/2314. $100 \leq n_1 \leq 65535$. If n_1 is omitted, 1000 is assumed. n_2 = residual capacity for beginning of operation notification when SYSPCH assigned to 2311. $100 \leq n_2 \leq 65535$. If n_2 is omitted, 1000 is assumed.
	*Valid when 24K bytes of main storage are available.	

Figure 4. Macro Instructions for Supervisor Generation (Part 2 of 4)

Operation	Operand	Explanation
PIOCS		Define options and configuration requirements to be included in physical IOCS.
	SELCH= {YES, NO}	Specify if selector channels are attached to the system
	BMPX= {NO, YES}	Specify if burst mode devices on multiplexor channel is supported. If 1412/1419's are attached to the multiplexor channel BMPX=YES is not supported.
	CHANSW= {NO, RWTAU, TSWTCH}	Specify if channel switching.
	TAPE= {9, 7, NO}	Specify 9- or 7-track tape. 7 indicates support for both.
ALLOC	F1=nK, F2=nK	Specify storage partitioning.
IOTAB		Define the necessary input/output tables for the system.
	BGPGR= {10, n}	Specify the number of logical unit blocks (LUBs) for programmer units, i.e., the number of symbolic programmer logical units (SYS000 - SYSnnn).
	F1PGR= {5, n}	Specify the number of symbolic units of the class SYSnnn for F1.
	F2PGR= {5, n}	Specify the number of symbolic units of the class SYSnnn for F2.
	JIB= {5, n}	Number of JIBs for the system. Minimum value generated is 5.
	CHANQ= {6, n}	Number of entries in the channel queue. Minimum value is 6.
	IODEV= {10, n}	Specify the number of I/O devices attached to the system. The minimum value is 5.
DVCGEN		Specify the physical I/O units attached to the system.
	CHUN=X'cuu'	Hexadecimal number of channel and unit.
	DVCTYP=xxxxxx	Specify the device type. See Figure 2.
	CHANSW= {NO, YES}	YES indicates that the device is attached to more than one selector channel (the device is switchable).
	MODE=X'ss'	1. 2400T9. MODE is used to specify the tape mode. X'C0' is the default value. 2. 2400T7. MODE is used to specify the tape mode. X'90' is the default value. 3. 2702. MODE designates the SADxxx command. X'00' is the default value. X'00' SAD0, X'01' SAD1, X'02' SAD2, X'03' SAD3. 4. 2260 (Local). MODE is used to specify the 1053 printer when CHUN=X'cuu' refers to a 1053 attached to a 2848. This operand must be entered as MODE=X'01'. 5. 1412, 1419, 1259 MODE designates the external interrupt bit associated with magnetic ink character reader. X'01' External line 7 X'04' External line 5 X'10' External line 3 X'02' External line 6 X'08' External line 4 X'20' External line 2

Figure 4. Macro Instructions for Supervisor Generation (Part 3 of 4)

Operation	Operand	Explanation
ASSGN		Assign LUBs to PUBs as standard system assignments.
	SYSnnn,X'cuu'	Symbolic unit is assigned a hexadecimal channel and unit number.
SEND	[n]	End of supervisor macro instructions. n = beginning address of the problem program area.

Figure 4. Macro Instructions for Supervisor Generation (Part 4 of 4)

Card Code	Actual Device	Device Type
2400T9	Nine Track Magnetic Tapes	Tapes
2400T7	Seven Track Magnetic Tapes	
1442N1	1442N1 Card Read Punch	Card Readers - Punches
2520B1	2520B1 Card Read Punch	
2501	2501 Card Reader	Card Readers
2540R	2540 Card Reader	
2540P	2540 Card Punch	Card Punches
2520B2	2520B2 Card Punch	
1442N2	1442N2 Card Punch	
2520B3	2520B3 Card Punch	
1403	1403 Printer	Printers
1403U	1403 Printer with UCS Feature	
1404	1404 Printer	
1443	1443 Printer	
1445	1445 Printer	
1050A	1052 Printer - Keyboard	1050 Control Unit
UNSP	Unsupported Device	Unsupported. No Burst Mode on Multiplexor Channel
UNSPB	Unsupported Device	Unsupported with Burst Mode on Multiplexor Channel
2260 (Local)	2260 Display Unit	Display Unit
2260 (Local)	A 1053 attached to a 2848. The mode operand must be entered as MODE=X'01'	Printer

Card Code	Actual Device	Device Type
2311	2311 Disk Drive	DASD
2314	2314 Disk Storage Facility	DASD
2321	2321 Data Cell Drive	DASD
2671	2671 Paper Tape Reader	Paper Tape Reader
2495TC	2495TC Tape Cartridge Reader	Tape Cartridge Reader
1412	1412 Magnetic Character Reader	Magnetic Character Reader
1419	1419 Magnetic Character Reader or 1259 Magnetic Reader	Magnetic Character Reader
1419P	Primary control unit address on 1419 Dual Address Adapter	Magnetic Character Reader
1419S	Secondary control unit address on 1419 Dual Address Adapter	Magnetic Character Reader
2701 *	2701 Line Adapter Unit	Teleprocessing Lines
2702	2702 Transmission Control Unit	
2703	2703 Transmission Control Unit	
2703	IBM System/360 Model 25 with the Integrated Communication attachment.	Teleprocessing Lines
7770	7770 Audio Response Unit	Audio Response
7772	7772 Audio Response Unit	
1285	1285 Optical Reader	Optical Readers
1287	1287 Optical Reader	

* Note: A 2701 Line Adapter Unit attached to an IBM System/360 Model 25 must be placed on the multiplexor channel.

Note: The codes used in DVCGEN macros are identical to those used in IPL statements.

Figure 5. Device Code

Figure 6 shows macro instructions and parameters that generate the supervisors supplied by IBM.

2311 DISK SYSTEM SUPERVISOR

Operation	Operand
SUPVR	SYSTEM=DISK
CONFG	
STDJC	
FOPT	PC=YES
PIOCS	BMPX=YES,TAPE=7
IOTAB	
SEND	6144
END	

2314 DIRECT ACCESS STORAGE SUPERVISOR

Operation	Operand
SUPVR	SYSTEM=DISK
CONFG	
STDJC	
FOPT	PC=YES,SYSFIL=2314
PIOCS	BMPX=YES,CHANSW=RWTAU,TAPE=7
IOTAB	
SEND	8192
END	

• Figure 6. IBM-Supplied Supervisor for 2311 Disk Systems and 2314 Direct Access Storage Facility

Two IBM 2311 Disk Drives

The system is supplied in two volumes. The first volume consists of a core image library and a relocatable library. The second volume is composed of a core image library and a source statement library. Depending upon the user's configuration, these packs are used in varying order.

The following is an example of one of the many possible methods for generating a system. Figure 7 illustrates the system configuration upon which Example One (Figure 8) is based. The following steps are keyed to Example One:

Step 1 Disk-only users receive the system on disk. Disk users with at least one tape unit available unit receive the system on tape.

Before generating a system, disk-only users should have at least one initialized disk pack (VTOC on cylinder 199).

Users with two disk drives and at least one tape unit should have at least two initialized disk packs (VTOC on cylinder 199).

Mount the IBM-supplied core image and relocatable library volume. Users with tape must also mount a disk pack to which the tape is to be restored.

Before restoring the tape, the user must take the necessary action to perform or bypass the initialize disk routine.

a. If the disk packs have not been initialized, the IBM-supplied volume, a self-loading tape, is capable of initializing the packs (see Figure 8). The DLBL cards used in the following steps must contain the volume serial number appearing in the VOL card of the initialize disk control cards. Any volume serial number used in the EXTENT cards for the following steps must agree with the volume serial number in the VOL card for initialize disk control cards.

b. If the disk packs have been previously initialized, the initialize disk routine can be bypassed (see Figure 8). Any volume serial number used in the EXTENT cards for the following steps must agree with the volume serial number used when the packs were initialized.

Step 2 After step 1 is completed, the user must restore the system from the self-loading tape to the initialized pack. This step is omitted when a system is received on disk.

Step 3 When the tape has been restored, dial the address of the system residence disk drive into the load unit switches, and IPL to pass control to the DOS supervisor. See Appendix A for the IPL control statements.

Step 4 Perform a DSERV to display the directories. After determining which components are never to be used, perform a DELETR to delete any relocatable component never to be used.

Step 5 Copy the relocatable library to a second initialized disk, defining it as a private relocatable library. During this copying, the user can allocate the private library to the desired size. When allocating the size of the private library, consider the workfile requirements discussed in step 9. To compute the minimum size of a library, see <u>Allocating Library Sizes for 2311 and 2314 Disk Systems</u>.

For techniques on copying libraries (MERGE), see <u>Maintenance Procedures</u>.

Step 6 Disk only users, remove the IBM-supplied core image and relocatable library volume, and mount the IBM-supplied core image and source statement library volume.

Disk users that received the IBM-supplied systems on tape should mount the IBM-supplied core image and source statement library system tape, and can either:

a. replace the system residence pack mounted in step 1, and initialize it, or

b. bypass initialization (because the system residence pack mounted in step 1 is already initialized).

Step 7 After the previous step is performed, the user must restore the system from the self-loading tape to the initialized pack. This step is omitted when a system is received on disk.

Step 8 When the tape has been restored, dial the address of the system residence disk drive into the load unit switches and IPL to pass control to the DOS supervisor.

Perform a DSERV to display the directories. Then SSERV (display and punch) all the desired sample problems from the source statement library into cards. Punched output includes the sample problems with BKEND and CATALS cards. Four other books in the source statement library that should be punched out at this time (Z.LINKEDIT, Z.DELETECL, Z.DELETERL, and Z.DELETESL) contain the necessary control statements to selectively linkage edit and delete all IBM components. The sample problem program names, and the linkage edit and the delete book names can be chosen and punched into the DSPCH statement(s). The sample problem program names and the linkage edit and delete book names are shown in <u>Appendix C</u>.

After these sample problems and books have been punched, they can be deleted from the source statement library, along with unwanted macros (e.g., those macros never to be used).

Step 9 Define workfiles for SYSLNK, SYS001, SYS002, and SYS003 to the second drive. The workfiles SYSLNK, SYS001, SYS002, and SYS003 are defined by use of the DLBL and EXTENT cards. These cards must be preceded by the OPTION STDLABEL or OPTION PARSTD card.

Step 10 Perform all necessary assemblies. The assemblies for the supervisor, IOCS modules, and emulator should be performed as separate jobs. For the information required to assemble Emulator Programs, refer to the <u>Emulator Program</u> manual listed in the Preface. The user must be careful to keep all assemblies in order.

Step 11 Assemble all user-required IOCS modules. By assigning SYSPCH to a tape unit, the IOCS modules can be cataloged to the relocatable library without punching them on cards. The IOCS modules required by COBOL, PL/I, and RPG, as defined in <u>Appendix B</u>, are supplied in the relocatable library by IBM.

Close the tape assigned to SYSPCH and reassign SYSPCH to its permanent assignment by using the CLOSE command. The assembly listings should be checked for errors before proceeding.

Step 12 Delete unwanted macros from the source statement library, and then create a private source statement library. If space allows, this copy could be made to a free area of the pack to which the relocatable library was copied.

For techniques on creating a new library, see <u>Maintenance Procedures</u>.

Step 13 This step defines the creation of the core image library of the user's operational pack.

The user can either delete the source statement library and reallocate the system to create a large core image library, or reallocate to create a large core image and a small system source statement library on the operational pack.

For techniques on creating a new library, see <u>Maintenance Procedures</u>.

Step 14 During this step the user must have the core image library of his operational pack that he is building on line, as well as his relocatable library.

If linkage edit work files are not assigned, they must be assigned now.

Linkage edit and catalog the assembled supervisor (object module from step 10) to the core image library. If the SEND address is larger than the one used by the supervisor being replaced, certain key programs must also be linkage edited and cataloged to the core image library in the same job step with the new supervisor. These key programs are IPL, linkage editor, and librarian. The LINKEDIT deck punched out in step 8 contains all of the necessary control statements to linkage edit all IBM components shipped on the system. If the SEND address is not exceeded, only the supervisor need be cataloged. <u>The new supervisor is not cataloged until a /& statement is read</u>. The user must not attempt any other operation from the time the supervisor and these preceding programs are cataloged until IPL time.

Step 15 Re-IPL and set the date (and clock if the Timer Feature is present).

Linkage edit and catalog any additional components desired to the core image library. See <u>Appendix C</u> for a complete list of control cards for all IBM components to be cataloged. Before the next step is performed, check the linkage editor listings, and make all necessary corrections.

Step 16 Reload the tape that was assigned to SYSPCH in step 11 and assign it to SYSIPT. With this tape the MAINT program catalogs the IOCS modules to the relocatable library by the control card // EXEC MAINT. The user may set new standard labels (OPTION STDLABEL), reallocate library sizes, and set automatic condense limits, if required. Backup for the operational disk can be obtained by copying the operational disk pack to tape by using the copy disk-to-tape utility program.

INPUT/OUTPUT DEVICE CONFIGURATION FOR EXAMPLE ONE (SEE FIGURE 7)

Device	Channel	Unit	Use
2540R	0	0C	Card Reader (SYSRDR, SYSIPT)
2540P	0	0D	Card Punch (SYSPCH)
1403	0	0E	Printer (SYSLST)
1052	0	1F	Printer-Keyboard (SYSLOG)
2311	1	91	Disk (SYSLNK, SYS001, SYS002, SYS003, SYS004, SYSRLB, SYSSLB)
2311	1	90	Disk (SYSRES)
2321	1	92	Data Cell (SYS007)
2400T7	1	80	Magnetic tape with the data conversion feature (switchable to channel 2)
2400T9	1	81	Magnetic tape (switchable to channel 2)

Example One: Results of System Generation

When system generation is completed for example one (see Figure 8), the operational disk of the installation contains: Assembler, Basic FORTRAN, COBOL, RPG, PL/I, Autotest, all utilities, and the disk sort/merge programs in its core image library together with the installation's tailored supervisor, job control, linkage editor, and librarian programs. The tape shipped by IBM is retained as a backup tape. It is a selfloading tape capable of being restored onto disk.

The sample problems are punched out during step 8.

The private relocatable and source statement libraries contain all modules and macro definitions shipped from IBM (except teleprocessing, OLTEP, Assembler F, Tape Sort/Merge, and the Vocabulary file utility).

The core image library of the operational pack is built to contain those IBM programs chosen by the user.

At this point, the private libraries are condensed, and user IOCS modules are assembled and cataloged to the private relocatable library.

When system generation is completed, the sample problems should be run against the operational pack to ensure correct creation of all system programs.

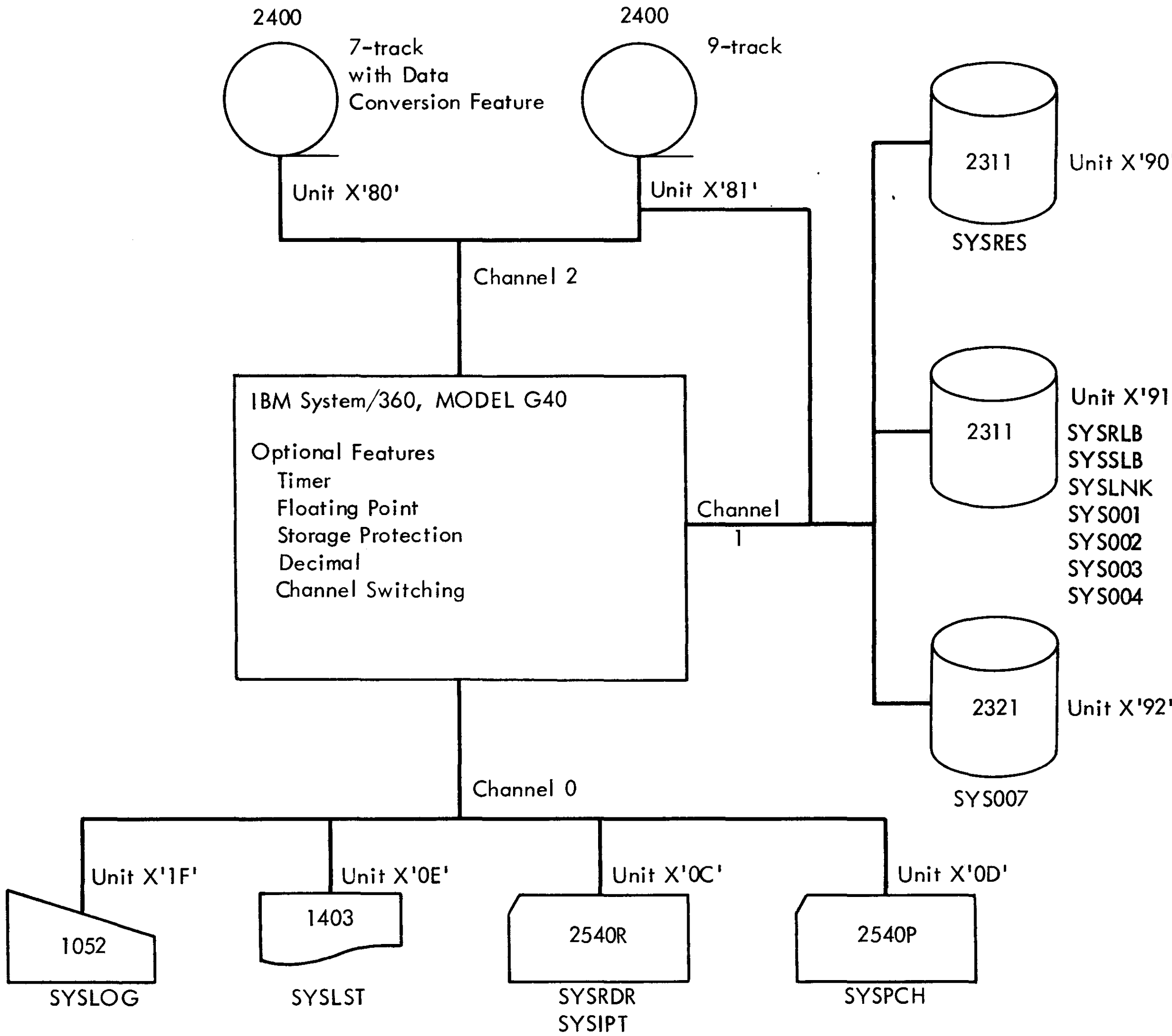


Figure 7. System Configuration for Example One

IBM — IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 1a

The user may optionally bypass the initialize routine. However, if initialization is required mount a pack on unit 190, mount a second pack on unit 191. Mount IBM-supplied tape on unit 181 (9-track drive). Place the following cards in the card reader. Dial 181 in the load unit switches and press load. When system enters the wait state press start and EOF on the card reader. To initialize the second pack, change the assignment for SYSLOG to // ASSGN SYSOPT,X'191',D1; rewind the IBM-supplied tape mounted on 181, reload the card reader with the following cards, and press load. When the system enters the wait state press start and EOF on the card reader. Use of multi-part forms is suggested as copies of the linkage editor maps and the supervisor listing will be required by your IBM customer engineer for maintenance purposes.

```
// JOB  INTDSK
// DATE 69032
// ASSGN SYSLOG,X'01F',C1
// ASSGN SYSOPT,X'190',D1
// EXEC
// UID  IR,C1
// VTOC STRTADR=(0199000),EXTENT=(9)
VOL1111111
// END
```

IBM — IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 1b

To bypass initialize disk program on IBM-supplied tape. Mount IBM-supplied tape on unit 181 (9-track drive). Mount initialized disk on unit 190 (VTOC on cyl 199). Mount work pack on 191 insert the following control cards in the reader. Dial 181 in load unit switches. Press load key. When system enters wait state, press start and EOF on the card reader. Job is complete when 00C and 4000A messages are logged.

```
// JOB  INTDSK
// DATE 69032
// ASSGN SYSLOG,X'01F'C1
// FILES SYSIPT,1
```

Figure 8. Example One (Part 1 of 14)

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 2

**** Caution **** Do not rewind unit 181. To load the tape onto disk, place the following cards in the reader. Dial 181 in load switches. Press load. When the system enters the wait state, press start and EOF on card reader. The following message appears on SYSLOG:
4444A. Type in 4(B) and press INTERRUPT to continue.

```
// JOB DISRST
// DATE 69032
// ASSGN SYS000,X'190',D1
// ASSGN SYSLOG,X'01F',C1
// ASSGN SYSLST,X'00E',L1
// EXEC
```

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 3

After the system is loaded onto disk generate the installation system. Dial 190 in load unit switches. Press load key on console. When system enters the wait state, press start on the card reader. The following cards are in the card reader (SYSRDR/ SYSIPT).

```
ADD X'00C',2540R
ADD X'00D',2540P
ADD X'00E',1403
ADD X'01F',1050A
ADD X'190',2311
ADD X'191',2311
SET DATE=02/01/69,CLOCK=00/00/00
ASSGN SYSLOG,X'01F'
LOG
ASSGN SYSRDR,X'00C'
ASSGN SYSIPT,X'00C'
ASSGN SYSPCH,X'00D'
ASSGN SYSLST,X'00E'
```

Figure 8. Example One (Part 2 of 14)

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 4

```
// JOB DSERVRLB
// EXEC DSERV
 DSPLY ALL
/*

// EXEC MAINT
 DELETR IJL.ALL,IJN.ALL
 DELETR IJP.ALL,IJY.ALL,IJZ.ALL

 DELETR IJKSYSA,IJKSYSI
 RENAMR IJXSYSA,IJKSYSA,IJXSYSI,IJKSYSI
/*
/&
```

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 5

If a small private relocatable library is to be built on the operational system, remove the pack from 191 and mount the initialized pack that the operational system is to be built on. Then, repeat this job with the proper EXTENT and NEWVOL statements.

```
// JOB PVTRLB
// ASSGN SYSRLB,X'191'
// DLBL IJSYSRL,'DOS PVT REL LIB',99/365,SD
// EXTENT SYSRLB,nnnnnn,1,1,nnnn,nnnn
// EXEC CORGZ                                          Create private relocatable library.
   NEWVOL RL=nnn(n)
   COPYR ALL
/*
/&
// PAUSE
```

Figure 8. Example One (Part 3 of 14)

IBM — IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES

PROGRAMMER: STEP 6a

The user may optionally bypass the initialize routine. However, if initialization is required mount a pack on unit 190, mount a second pack on unit 191. Mount IBM-supplied tape on unit 181 (9-track drive). Place the following cards in the card reader. Dial 181 in the load unit switches and press load. When system enters the wait state press start and EOF on the card reader. Use of multi-part forms is suggested as copies of the linkage editor maps and the supervisor listing will be required by your IBM customer engineer for maintenance purposes.

```
// JOB INTDSK
// DATE 69032
// ASSGN SYSLOG,X'01F',C1
// ASSGN SYSOPT,X'190',D1
// EXEC
// UID IR,C1
// VTOC STRTADR=(0199000),EXTENT=(9)
VOL1111111
// END
```

IBM — IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES

PROGRAMMER: STEP 6b

To bypass initialize disk program on IBM-supplied tape. Mount IBM-supplied tape on unit 181 (9-track drive). Mount initialized disk on unit 190 (VTOC on cyl 199). Mount work pack on 191 insert the following control cards in the reader. Dial 181 in load unit switches. Press load key. When system enters wait state, press start and EOF on the card reader. Job is complete when 00C and 4000A messages are logged.

```
// JOB INTDSK
// DATE 69032
// ASSGN SYSLOG,X'01F',C1
// FILES SYSIPT,1
```

Figure 8. Example One (Part 4 of 14)

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES

PROGRAMMER: STEP 7

**** Caution **** Do not rewind unit 181. To load the tape onto disk, place the following cards in the reader. Dial 181 in load switches. Press load. When the system enters the wait state, press start and EOF on card reader. The following message appears on SYSLOG:
4444A. Type in 4 B and press INTERRUPT to continue.

```
// JOB DISRST
// DATE 69032
// ASSGN SYS000,X'190',D1
// ASSGN SYSLOG,X'01F',C1
// ASSGN SYSLST,X'00E',L1
// EXEC
```

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES

PROGRAMMER: STEP 8

After the system is loaded onto disk, generate the installation system. Dial 190 in load unit switches. Press load key on console. When system enters the wait state, press start on the card reader. The following cards are in the card reader (SYSRDR/SYSIPT).

```
ADD X'00C',2540R
ADD X'00D',2540P
ADD X'00E',1403
ADD X'01F',1050A
ADD X'190',2311
ADD X'191',2311
ADD X'181',2400T9
```

Figure 8. Example One (Part 5 of 14)

IBM IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 8 (continued)

```
SET DATE=02/01/69,CLOCK=00/00/00
ASSGN SYSLOG,X'01F'
LOG
ASSGN SYSRDR,X'00C'
ASSGN SYSIPT,X'00C'
ASSGN SYSPCH,X'00D'
ASSGN SYSLST,X'00E'
ASSGN SYS001,X'191'
ASSGN SYS002,X'191'
ASSGN SYS003,X'191'
```

IBM IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 8 (continued)

```
// JOB DSERVSLB
// EXEC DSERV
 DSPLY ALL
/*


// EXEC SSERV
 DSPLY A.DOSCHLV               Display the change level.
 DSPCH Z.ALL                   Display and punch sample probelms, delete, and link edit books. See Appendix C for names.
/*
// PAUSE REMOVE CARDS FROM SYSPCH. PRESS EOB TO CONTINUE.


/&
```

Figure 8. Example One (Part 6 of 14)

IBM — IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 9

```
// JOB ASSEMSUP
// OPTION DECK,LIST,LOG,STDLABEL
// DLBL IJSYS01,'SYSTEM WORK FILE NO. 1',99/365,SD
// EXTENT SYS001,nnnnnn,1,n,nnnn,nnnn
// DLBL IJSYS02,'SYSTEM WORK FILE NO. 2',99/365,SD
// EXTENT SYS002,nnnnnn,1,n,nnnn,nnnn
// DLBL IJSYS03,'SYSTEM WORK FILE NO. 3',99/365,SD
// EXTENT SYS003,nnnnnn,1,n,nnnn,nnnn
// DLBL IJSYSRL,'DOS PVT REL LIB',99/365,SD
// EXTENT SYSRLB,nnnnnn,1,1,nnnn,nnnn
```

IBM — IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 10

```
// EXEC ASSEMBLY
         SUPVR SYSTEM=DISK,MPS=YES
         CONFG MODEL=40,SP=YES,DEC=YES,FP=YES,TIMER=YES
         STDJC LISTX=YES,LINES=46
         FOPT  IT=BG,PC=YES,OC=YES,CCHAIN=YES,DASDFP=(1,2,2321),        X
               SYSFIL=(2311),TEB=4
         PIOCS CHANSW=RWTAU,TAPE=7,BMPX=YES
         ALLOC F1=6K,F2=6K
         IOTAB JIB=10,CHANQ=10,F1PGR=8,F2PGR=8,IODEV=10,BGPGR=10
         DVCGEN CHUN=X'00C',DVCTYP=2540R
         DVCGEN CHUN=X'00D',DVCTYP=2540P
         DVCGEN CHUN=X'00E',DVCTYP=1403
         DVCGEN CHUN=X'01F',DVCTYP=1050A
         DVCGEN CHUN=X'190',DVCTYP=2311
         DVCGEN CHUN=X'191',DVCTYP=2311
         DVCGEN CHUN=X'192',DVCTYP=2321
         DVCGEN CHUN=X'180',DVCTYP=2400T7,CHANSW=YES
         DVCGEN CHUN=X'181',DVCTYP=2400T9,CHANSW=YES
```

Figure 8. Example One (Part 7 of 14)

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 10 (continued)

```
         ASSGN SYSRDR,X'00C'
         ASSGN SYSIPT,X'00C'
         ASSGN SYSPCH,X'00D'
         ASSGN SYSLST,X'00E'
         ASSGN SYSLOG,X'01F'
         ASSGN SYSLNK,X'191'
         ASSGN SYS001,X'191'
         ASSGN SYS002,X'191'
         ASSGN SYS003,X'191'
         ASSGN SYS004,X'191'
         ASSGN SYS007,X'192'
         SEND  10240
 END
/*
*    CHECK ASSEMBLY LISTING FOR ERRORS.  IF CORRECT
* REMOVE ASSEMBLED SUPERVISOR FROM SYSPCH.  INSERT IN READER
* FOLLOWING THE INCLUDE CARD OF JOB CATALSUP
// PAUSE TO CONTINUE PRESS EOB

/&
```

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 11

```
* AT THIS POINT ASSEMBLE IOCS MODULES TO BE CATALOGED TO THE
* RELOCATABLE LIBRARY
* UNIT 181 MUST BE UNASSIGNED AT THIS POINT.
* THE FOLLOWING IS AN EXAMPLE.  MODULES SHOULD BE ASSEMBLED ON AN
*    AS REQUIRED BASIS.
// JOB ASSEM
// OPTION DECK,LIST,LOG
// ASSGN SYSPCH,X'181'
// EXEC ASSEMBLY
    CDMOD          RECFORM=FIXUNB,                                     X
                   CTLCHR=ASA,                                         X
                   TYPEFLE=OUTPUT,                                     X
                   IOAREA2=YES,                                        X
                   DEVICE=2540R,                                       X
                   SEPASMB=YES
  END
/*
// EXEC ASSEMBLY
    MTMOD          RECFORM=FIXUNB,                                     X
                   READ=FORWARD,                                       X
                   CKPTREC=YES,                                        X
                   SEPASMB=YES
  END
/*
```

Figure 8. Example One (Part 8 of 14)

IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES

PROGRAMMER: STEP 11 (continued)

```
 CLOSE SYSPCH,X'00D'
/&
*  CHECK ASSEMBLY LISTINGS FOR ERRORS. RELOAD 181 WITH SAME TAPE.
//  PAUSE IF CORRECT PRESS EOB TO CONTINUE
```

IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK-DRIVES

PROGRAMMER: STEP 12

```
// JOB PVTSLB
// EXEC MAINT
```

Include DELETS statements for those macro definitions that are not wanted on the private source statement library. All DELETS statements are included in the Z.DELETESL book that was displayed and punched in step 8.

```
/*
// ASSGN SYSSLB,X'191'
// DLBL IJSYSSL,'DOS PVT SRC LIB',99/365,SD
// EXTENT SYSSLB,nnnnnn,1,1,nnnn,nnnn
* CREATE PRIVATE SOURCE STATEMENT LIBRARY ON DRIVE 191
// EXEC CORGZ
   NEWVOL SL=nnn(n)
   COPYS ALL
/*
/&

// PAUSE
```

Figure 8. Example One (Part 9 of 14)

IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 13

```
// JOB SYSRES
// EXEC MAINT
 DELETS A.ALL,Z.ALL
/*
// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',99/365,SD
// EXTENT SYSRES,111111,1,,0001,nnn9
// EXEC MAINT
   ALLOC CL=nnn(n),SL=0(0)    Number of cylinders allocated to CL must be at least two less than EXTENT for SYSRES.
/*

/&

// JOB LABELS
```

IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 13 (continued)

```
// OPTION DECK,LIST,LOG,STDLABEL
// DLBL IJSYSLN,'SYSTEM WORK FILE NO. 0',99/365,SD
// EXTENT SYSLNK,nnnnnn,1,n,nnnn,nnnn
// DLBL IJSYS01,'SYSTEM WORK FILE NO. 1',99/365,SD
// EXTENT SYS001,nnnnnn,1,n,nnnn,nnnn       Assign work files and private libraries.
// DLBL IJSYS02,'SYSTEM WORK FILE NO. 2',99/365,SD
// EXTENT SYS002,nnnnnn,1,n,nnnn,nnnn
// DLBL IJSYS03,'SYSTEM WORK FILE NO. 3',99/365,SD
// EXTENT SYS003,nnnnnn,1,n,nnnn,nnnn
// DLBL IJSYSRL,'DOS PVT REL LIB',99/365,SD
// EXTENT SYSRLB,nnnnnn,1,1,nnnn,nnnn
// DLBL IJSYSSL,'DOS PVT SRC LIB',99/365,SD
// EXTENT SYSSLB,nnnnnn,1,1,nnnn,nnnn

/&

// JOB CATALSUP
// ASSGN SYSRLB,X'191'
// ASSGN SYSLNK,X'191'
```

Figure 8. Example One (Part 10 of 14)

IBM — IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 14

```
// OPTION CATAL
   ACTION CLEAR
 INCLUDE
 Supervisor object deck here.
/*
// EXEC LNKEDT
 INCLUDE IJBLBP                          CSERV
// EXEC LNKEDT
 INCLUDE IJBIPL                          IPL
// EXEC LNKEDT
 INCLUDE IJBLE                           LINKAGE EDITOR
// EXEC LNKEDT
 INCLUDE IJBSL1                          DSERV
// EXEC LNKEDT
 INCLUDE IJBSL2                          MAINT
// EXEC LNKEDT
 INCLUDE IJBSL3                          RSERV
// EXEC LNKEDT
```

Steps 14 and 15 contain the coding for including the IBM components selected for this example. The LINKEDIT book displayed and punched in step 8 contains all the necessary job control statements to linkage edit any of the IBM components shipped on the system. The user is encouraged to use this book to tailor a job stream to include any IBM components desired.

Need not be linkage edited if the shipped supervisor is not exceeded.

IBM — IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 14 (continued)

```
 INCLUDE IJBSL4                          SSERV
// EXEC LNKEDT
 INCLUDE IJBSL5                          CORGZ
// EXEC LNKEDT
/&
```

Need not be linkage edited if the shipped supervisor is not exceeded.

At this point the system will indicate re-IPL is needed. IPL from SYSRES (190).

```
SET DATE=02/01/69,CLOCK=00/00/00
// JOB CONDSCLB
// EXEC MAINT
 DELETC ASSE.ALL
 CONDS CL
/*
/&
 ASSGN SYSRLB,X'191'
 ASSGN SYSSLB,X'191'
```

Figure 8. Example One (Part 11 of 14)

IBM System 360 Assembler Coding Form

PROGRAM EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER STEP 15

```
// JOB CATALCLB
* AT THIS POINT LINKAGE EDIT COBOL, BASIC FOTRAN, DISK SORT MERGE.
* UTILITIES, AUTOTEST,PL/I,RPG,14K ASSEMBLER
// OPTION CATAL
   ACTION CLEAR
 INCLUDE IJVPT                          AUTOTEST
// EXEC LNKEDT
 INCLUDE IJSCBD                         COBOL
// EXEC LNKEDT
 INCLUDE IJSDDB                         COBOL DEBUG
// EXEC LNKEDT
 INCLUDE IJTFO                          BASIC FORTRAN
// EXEC LNKEDT
 INCLUDE IJXPLID                        PL/I
// EXEC LNKEDT
 INCLUDE IJRRG                          RPG
// EXEC LNKEDT
 INCLUDE IJOSM                          SORT/MERGE
// EXEC LNKEDT
 INCLUDE IJQD32                         14K ASSEMBLER
// EXEC LNKEDT
```

IBM System 360 Assembler Coding Form

PROGRAM EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER STEP 15 (continued)

```
 INCLUDE IJWTM                          TAPE TO DATA CELL UTILITY
 PHASE TPDC5,IJWTMCS2,NOAUTO
 INCLUDE IJWLAB
// LBLTYP TAPE
// EXEC LNKEDT
 INCLUDE IJWTD                          TAPE TO DISK UTILITY
 PHASE TPDK5,IJWTDCS2,NOAUTO
 INCLUDE IJWLAB
// LBLTYP TAPE
// EXEC LNKEDT
 INCLUDE IJWTP                          TAPE TO PRINTER UTILITY
 PHASE TPPR5,IJWTPCS2,NOAUTO
 INCLUDE IJWLAB
// LBLTYP TAPE
// EXEC LNKEDT
 INCLUDE IJWTC                          TAPE TO CARD UTILITY
 PHASE TPCD5,IJWTCCS2,NOAUTO
 INCLUDE IJWLAB
// LBLTYP TAPE
// EXEC LNKEDT
```

Figure 8. Example One (Part 12 of 14)

IBM — IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 15 (continued)

```
 INCLUDE IJWCT                          CARD TO TAPE UTILITY
 PHASE CDTP5,IJWCTCS2,NOAUTO
 INCLUDE IJWLAB
// LBLTYP TAPE
// EXEC LNKEDT
 INCLUDE IJWCD                          CARD TO DISK UTILITY
 PHASE CDDK5,IJWCDCS2,NOAUTO
 INCLUDE IJWLAB
// EXEC LNKEDT
 INCLUDE IJWCP                          CARD TO PRINTER/PUNCH UTILITY
 PHASE CDPP5,IJWCPCS2,NOAUTO
 INCLUDE IJWLAB
// EXEC LNKEDT
 INCLUDE IJWDP                          DISK TO PRINTER UTILITY
 PHASE DKPR5,IJWDPCS2,NOAUTO
 INCLUDE IJWLAB
// EXEC LNKEDT
 INCLUDE IJWDC                          DISK TO CARD UTILITY
 PHASE DKCD5,IJWDCCS2,NOAUTO
 INCLUDE IJWLAB
// EXEC LNKEDT
```

IBM — IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 15 (continued)

```
 INCLUDE IJWDT                          DISK TO TAPE UTILITY
 PHASE DKTP5,IJWDTCS2,NOAUTO
 INCLUDE IJWLAB
// LBLTYP TAPE
// EXEC LNKEDT
 INCLUDE IJWDM                          DISK TO DATA CELL UTILITY
 PHASE DKDC5,IJWDMCS2,NOAUTO
 INCLUDE IJWLAB
// EXEC LNKEDT
 INCLUDE IJWDD                          DISK TO DISK UTILITY
 PHASE DKDK5,IJWDDCS2,NOAUTO
 INCLUDE IJWLAB
// EXEC LNKEDT
 INCLUDE IJWMD                          DATA CELL TO DISK UTILITY
 PHASE DCDK5,IJWMDCS2,NOAUTO
 INCLUDE IJWLAB
// EXEC LNKEDT
 INCLUDE IJWMT                          DATA CELL TO TAPE UTILITY
 PHASE DCTP5,IJWMTCS2,NOAUTO
 INCLUDE IJWLAB
// LBLTYP TAPE
// EXEC LNKEDT
```

Figure 8. Example One (Part 13 of 14)

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 15 (continued)

```
 INCLUDE IJWMP                                  DATA CELL TO PRINTER UTILITY
 PHASE DCPR5,IJWMPCS2,NOAUTO
 INCLUDE IJWLAB
// EXEC LNKEDT
 INCLUDE IJWMM                                  DATA CELL TO DATA CELL UTILITY
 PHASE DCDC5,IJWMMCS2,NOAUTO
 INCLUDE IJWLAB
// EXEC LNKEDT
 INCLUDE IJWCLD                                 CLEAR DISK UTILITY
// EXEC LNKEDT
 INCLUDE IJWCLM                                 CLEAR DATA CELL UTILITY
// EXEC LNKEDT
 PHASE TPCP,*,NOAUTO                            TAPE COMPARE UTILITY
 INCLUDE IJWTCP
 INCLUDE IJJCPD1
 INCLUDE IJWXIT
 INCLUDE IJWTPCP
// LBLTYP TAPE
// EXEC LNKEDT
/*
/&

// JOB CATALRLB
```

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISK DRIVES
PROGRAMMER: STEP 16

```
// ASSGN SYSIPT,X'181'
* CATALOG MODULES TO RELOCATABLE LIBRARY
// EXEC MAINT
/*
* THE FOLLOWING DELETIONS ARE OPTIONAL.
// PAUSE PLACE REMAINING CARDS IN READER. PRESS EOB TO CONTINUE
// EXEC MAINT

 Place any desired deletes here.

 CONDS RL,SL
/*
/&
* THE BASIC SYSGEN IS COMPLETE. THE USER MAY SET NEW STANDARD
* LABELS, RE-ALLOCATE LIBRARY SIZES, AND SET AUTOMATIC CONDENSE
* LIMITS IF REQUIRED. COPY THE OPERATIONAL PACK TO TAPE FOR BACK-UP.
* USE THE IDENTICAL UTILITY DECK SETUP AS AFTER FIRST SYSGEN JOB.
* THE FINAL CONDENSE OPERATION SUPPLIED THE CURRENT STATUS
* OF THE SYSTEM LIBRARIES.
// PAUSE EOJ SYSGEN-----.
```

Figure 8. Example One (Part 14 of 14)

One IBM 2311 Disk Drive

The system is supplied in two volumes. The first volume consists of a core image library and a relocatable library. The second volume is composed of a core image library and a source statement library. Depending upon the user's configuration, these packs are used in varying order.

IBM supplies the 10K background variant of the disk work file assembler in the core image library of the source statement library pack. The following procedure employs this assembler variant for system generation. If the user wishes to use another variant, he must:

- Linkage edit and catalog the desired assembler to the core image library of the relocatable library system.
- CSERV the assembler just cataloged.
- Delete the supplied assembler from the core image library of the source statement library system, and condense this library.
- Catalog the assembler obtained through the previous CSERV to the core image library of the source statement library volume.

The user can now perform system generation as described. Figure 9 illustrates the system configuration upon which Example Two (Figure 10) is based. The following steps are keyed to Example Two:

Step 1 Disk-only users receive the system on disk. Disk users with at least one tape unit available receive the system on tape.

Mount the IBM-supplied core image and source statement library volume. Users with tape must also mount a disk pack to which the tape is to be restored.

Before generating a system with at least one tape unit the user should have an initialized disk pack (VTOC on cylinder 199).

a. If the disk pack has not been initialized, the IBM-supplied volume, a self-loading tape, is capable of initializing the pack (see Figure 10). The DLBL cards used in the following steps must contain the volume serial number appearing in the VOL card of the initialize disk control cards. Any volume serial number used in the EXTENT cards for the following steps must agree with the volume serial number in the VOL card for the initialize disk control card.

b. If the disk pack has been previously initialized, the initialize disk routine can be bypassed (see Figure 10). Any volume serial number used in the EXTENT cards for the following steps must agree with the volume serial number used when the pack was initialized.

Step 2 After step 1 is completed, the user must restore the system from the self-loading tape to the initialized pack. This step is omitted when a system is received on disk.

Step 3 When the tape has been restored, dial the address of the system residence disk drive into the load unit switches, and IPL to pass control to the DOS supervisor. Then perform a DSERV to display the directories. See Appendix A for the IPL control statements.

Step 4 SSERV (display and punch) all the desired sample problems from the source statement library into cards. Punched output includes the sample problems with BKEND and CATALS cards. Four other books in the source statement library that should be punched out at this time (Z.LINKEDIT, Z.DELETECL, Z.DELETERL, and Z.DELETESL) contain the necessary control statements to selectively linkage edit and delete all IBM components. The sample problem program names, the linkage edit, and the delete book names can be chosen and punched using the DSPCH statement(s). The sample problem program names and the linkage edit and delete book names are shown in Appendix C.

After these sample problems and books have been punched, they can be deleted from the source statement library along with unwanted macros (e.g., those components never to be used).

Step 5 If additional space is needed for workfiles to perform assemblies, (SYS001, SYS002 and SYS003) the user can re-allocate the system to take advantage of the space gained by the deletes from the previous step.

Step 6 The workfiles SYSLNK, SYS001, SYS002, and SYS003 are defined by use of the DLBL and EXTENT cards. These cards must be preceded by the OPTION STDLABEL or OPTION PARSTD card.

Step 7 Perform all necessary assemblies. The assemblies for the supervisor, and IOCS modules should be performed as separate job steps. The user must be careful to keep all assemblies in order.

If the user's operational system is to contain a minimum source statement library with selected macros, they should be obtained through an SSERV. The macros obtained in this step will later be placed on SYSIPT and cataloged to a system source statement library.

Step 8 Disk users without tape must remove the IBM-supplied core image and source statement library pack, and replace it with the IBM-supplied core image and relocatable library pack. The pack removed should be retained for backup.

Users with tape can either replace the core image and source statement library pack, or they can restore the IBM-supplied core image and relocatable library tape to this same initilized disk pack.

Before restoring the tape, the user must take the necessary action to perform or bypass the initialize disk routine.

Step 9 After step 8 is completed, the user must restore the system from the self-loading tape to the initialized pack. This step is omitted when a system is received on disk.

Step 10 Dial the address of the system residence disk drive into the load unit switches, and IPL to pass control to the DOS supervisor.

Step 11 Perform a DSERV to display the directories. Delete all relocatable library components not to be used.

Step 12 Reallocate library sizes, assigning all free space with the exception of linkage editor work files, to the core image library. If room is available at this time, the user can also allocate the small source statement library to contain those macros punched in step 7. Otherwise, this allocation can be done in step 15. The linkage editor work files (SYSLNK and SYS001) are already defined on this pack. The user need only assign SYSLNK and SYS001 to this drive. (See step 10 in Figure 10.)

Step 13 Linkage edit and catalog the assembled supervisor (from step 7) to the core image library. Certain key programs must also be linkage edited and cataloged in the same job step with the new supervisor if the SEND address is larger than the one used by the supervisor being replaced. These key programs are IPL, linkage editor, and librarian. The control statements to linkage edit these programs are in the linkage edit deck punched out in step 4. The SSERV and assembler components illustrated in Figure 10, step 13, should be included as part of this job only if the user's operational system is to contain a minimum source statement library.

The new supervisor is not cataloged until a /& statement is read. The user must not attempt any other operations from the time when the supervisor and preceding programs are cataloged and the subsequent IPL.

Step 14 After these key programs have been cataloged, re-IPL, and set the date (set the clock if the timer feature is present).

Linkage edit and catalog any additional components desired to the core image library. See Appendix C for a complete, list of control cards for all components. The LINKEDIT deck punched out in step 4 contains all of the necessary control statements to linkage edit all components shipped on the system.

Before the next step is performed check the linkage editor listings and make all necessary corrections before deleting the modules from the relocatable library.

Step 15 If the components deleted in step 11 did not provide adequate space for the final allocation of system libraries, a new allocation can be performed by deleting components previously cataloged to the core image library. More then one deletion and allocation may be required during linkage edit jobs. Users desiring a small source statement library should now allocate space for one.

Catalog the modules assembled in step 7 to the system relocatable library. If your decision was to have a source statement library on this pack, catalog the source statement macros punched in step 7 to the system source statement library.

The user may set new standard labels, re-allocate library sizes, and set automatic condense limits. Backup for the operational system should also be obtained.

INPUT/OUTPUT DEVICE CONFIGURATION FOR EXAMPLE TWO (SEE FIGURE 9)

Device	Channel	Unit	Use
2540R	0	0C	Card Reader (SYSRDR/SYSIPT)
2540P	0	0D	Card Punch (SYSPCH)
1403	0	0E	Printer (SYSLST)
1052	0	1F	Printer-Keyboard (SYSLOG)
2400T7	1	81	Magnetic Tape with data conversion feature
2311	1	90	Disk (SYSRES, SYSLNK, SYS001, SYS002, SYS003)

Example Two: Results of System Generation

When system generation is completed for Example Two (see Figure 10), the operational disk contains COBOL, PL/I, Basic FORTRAN, Tape Sort/Merge and Utilities together with the installation's tailored supervisor, job control, linkage editor, and librarian programs in the core image library. The tape shipped by IBM is retained as a backup tape. The relocatable library contains all those components shipped by IBM except those that were deleted in steps 11 and 15. There is no source statement library on this operational pack. This is a user option, and was indicated as such in the example. Thus, the user may either utilize the source statement library shipped by IBM on the source statement library volume, or he may build a source statement library on the operational pack by allocating space and cataloging his desired macro definitions.

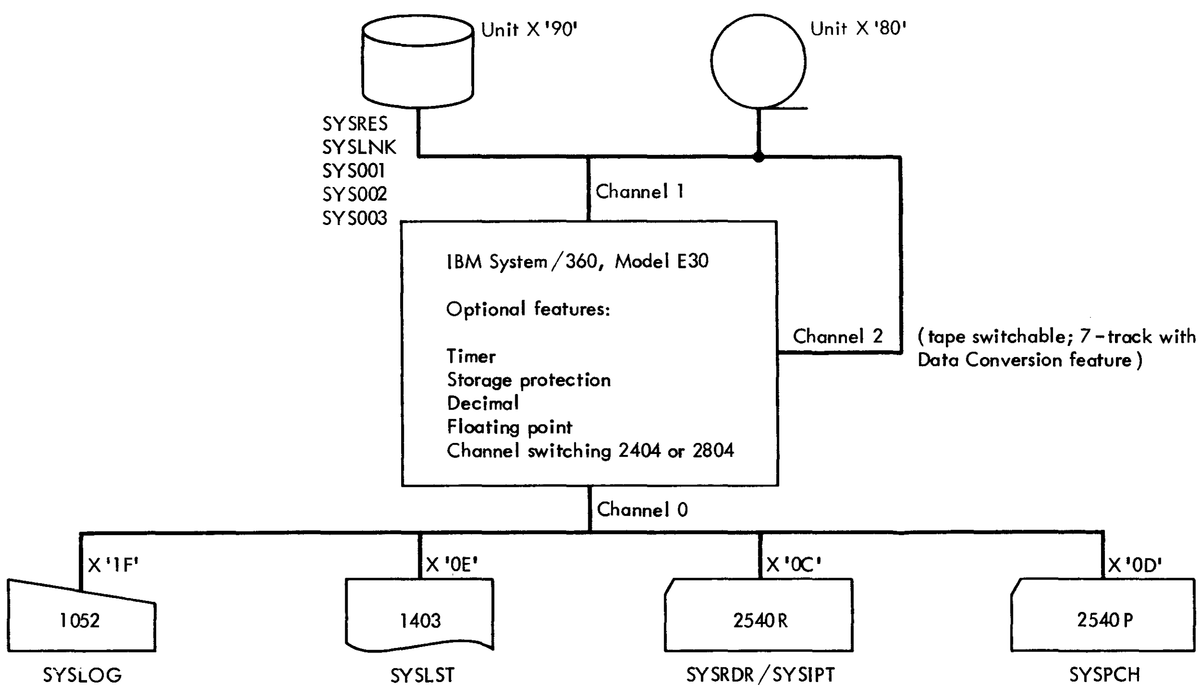


Figure 9. System Configuration for Example Two

IBM — IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 1a

The user may optionally bypass the initialize routine. However, if initialization is required mount a pack on unit 190. Mount IBM-supplied tape on unit 180 (7-track drive). Place the following cards in the card reader. Dial 180 in the load unit switches and press load. When system enters the wait state press start and EOF on the card reader. Use of multi-part forms is suggested as copies of the linkage editor maps and the supervisor listing will be required by your IBM customer engineer for maintenance purposes.

```
// LOG
// JOB INTDSK
// DATE 69032
// ASSGN SYSLOG,X'01F',C1
// ASSGN SYSOPT,X'190',D1
// EXEC
// UID IR,C1
// VTOC STRTADR=(0199000),EXTENT=(9)
VOL1111111
// END
```

IBM — IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 1b

To bypass initialize disk program on IBM-supplied tape, mount IBM-supplied tape on unit 180 (7-track drive). Mount initialized disk on unit 190 (VTOC on CYL 199). Insert the following control cards in the reader. Dial 180 in load unit switches. Press load key. When system enters wait state, press start and EOF on the card reader. Job is complete when 00C and 4000A messages are logged.

```
// LOG
// JOB INTDSK
// DATE 69032
// ASSGN SYSLOG,X'01F',C1
// FILES SYSIPT,1
```

Figure 10. Example Two (Part 1 of 12)

IBM — IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE

PROGRAMMER: STEP 2

**** Caution **** Do not rewind unit 180. To load the tape onto disk, place the following cards in the reader. Dial 180 in load switches. Press load. When the system enters the wait state, press start and EOF on card reader. The following message appears on SYSLOG:
4444A. Type in 4 B and press INTERRUPT to continue.

```
// JOB DISRST
// DATE 69032
// ASSGN SYS000,X'190',D1
// ASSGN SYSLOG,X'01F',C1
// ASSGN SYSLST,X'00E',L1
// EXEC
```

IBM — IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE

PROGRAMMER: STEP 3

After the system is loaded onto disk generate the installation system. Dial 190 in load unit switches. Press load key on console. When system enters the wait state, press start and EOF on the card reader. The following cards are in the card reader.

```
ADD X'00C',2540R
ADD X'00D',2540P
ADD X'00E',1403
ADD X'01F',1050A
ADD X'190',2311
ADD X'180',2400T7
```

Figure 10. Example Two (Part 2 of 12)

IBM — IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 3

```
SET DATE=02/01/69,CLOCK=00/00/00
ASSGN SYSLOG,X'01F'
LOG
ASSGN SYSRDR,X'00C'
ASSGN SYSIPT,X'00C'
ASSGN SYSLST,X'00E'
ASSGN SYSPCH,X'00D'
ASSGN SYS001,X'190'
ASSGN SYS002,X'190'
ASSGN SYS003,X'190'
// JOB DSERVSLB
// EXEC DSERV
   DSPLY ALL
/*
/&
```

IBM — IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 4

```
// JOB DELETE
* THIS JOB PUNCHES AND DELETES SAMPLE PROGRAMS,DELETE BOOKS AND THE LINK
* EDIT BOOK
* IT ALSO DELETES PAPER-TAPE IOCS MACROS AND EMULATOR MACROS.
// EXEC SSERV
   DSPCH Z.ALL
/*
// PAUSE REMOVE SAMPLE PROGRAM CARDS FROM SYSPCH,PRESS EOB TO CONT.
// EXEC MAINT
   DELETS Z.ALL
   DELETS A.DTFPT,A.PTMOD
   DELETS A.EU3CG,A.EU3DK,A.EU3EJ,A.EU3ER,A.EU3FT,A.EU3MS
   DELETS A.EU3OS,A.EU3PH,A.EU3PT,A.EU3RD,A.EU3TP,A.EU30
/*
```

See Appendix C for a list of all names.

Figure 10. Example Two (Part 3 of 12)

IBM — IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 5

```
// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',99/365,SD
// EXTENT SYSRES,111111,1,n,nnnn,nn9
// EXEC MAINT
   ALLOC CL=nn(n),SL=nn(n)
/*
/&
```

IBM — IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 6

```
* SYSRES NOW CONTAINS THE ORIGINAL CORE IMAGE LIBRARY, ORIGINAL SOURCE
* STATEMENT LIBRARY (LESS DELETED SAMPLE PROBLEM, PAPER TAPE
* MACROS AND EMULATOR)
* AT THIS POINT, GENERATE THE TAILORED SUPERVISOR
// JOB ASSEM
// OPTION DECK,LIST,LOG,STDLABEL
// DLBL IJSYS01,'SYSTEM WORK FILE NO. 1',99/365,SD
// EXTENT SYS001,111111,8,n,nnn0,nnn,2
// DLBL IJSYS02,'SYSTEM WORK FILE NO. 2',99/365,SD
// EXTENT SYS002,111111,8,n,nnn3,nnn,5
// DLBL IJSYS03,'SYSTEM WORK FILE NO. 3',99/365,SD
// EXTENT SYS003,111111,8,n,nnn6,nnn,9
```

Figure 10. Example Two (Part 4 of 12)

IBM IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 7

```
// EXEC ASSEMBLY
         SUPVR SYSTEM=DISK
         CONFG MODEL=30,SP=YES,DEC=YES,FP=YES,TIMER=YES
         STDJC LISTX=YES,LINES=46
         FOPT  IT=BG,PC=YES,OC=YES,TEB=6
         PIOCS CHANSW=RWTAU,TAPE=7,BMPX=YES
         IOTAB IODEV=10,JIB=8,CHANQ=6
         DVCGEN CHUN=X'00C',DVCTYP=2540R
         DVCGEN CHUN=X'00D',DVCTYP=2540P
         DVCGEN CHUN=X'00E',DVCTYP=1403
         DVCGEN CHUN=X'01F',DVCTYP=1050A
         DVCGEN CHUN=X'190',DVCTYP=2311
         DVCGEN CHUN=X'180',DVCTYP=2400T7,CHANSW=YES,MODE=X'90'
```

IBM IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 7 (continued)

```
         ASSGN SYSRDR,X'00C'
         ASSGN SYSIPT,X'00C'
         ASSGN SYSPCH,X'00D'
         ASSGN SYSLST,X'00E'
         ASSGN SYSLOG,X'01F'
         ASSGN SYSLNK,X'190'
         ASSGN SYS001,X'190'
         ASSGN SYS002,X'190'
         ASSGN SYS003,X'190'
         SEND  8192
         END
/*
* CHECK ASSEMBLY LISTINGS FOR ERRORS. IF CORRECT
* REMOVE ASSEMBLED SUPERVISOR FROM SYSPCH, INSERT IN READER
* FOLLOWING THE INCLUDE CARD IN STEP 13
/&
```

Figure 10. Example Two (Part 5 of 12)

IBM — IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 7 (continued)

```
*  AT THIS POINT ASSEMBLE IOCS MODULES TO BE CATALOGED TO THE
*  RELOCATABLE LIBRARY, MOUNT AND READY A SCRATCH TAPE ON 180.
*  THE FOLLOWING IS AN EXAMPLE. MODULES SHOULD BE ASSEMBLED ON AN
*  AS REQUIRED BASIS.
// PAUSE TO CONTINUE PRESS EOB
// JOB ASSEM2
// OPTION DECK,LIST,LOG
// ASSGN SYSPCH,X'180'
// EXEC ASSEMBLY
         CDMOD RECFORM=FIXUNB,CONTROL=YES,TYPEFLE=INPUT,DEVICE=2540,   X
               SEPASMB=YES
         END
/*
// EXEC ASSEMBLY
         MTMOD RECFORM=VARUNB,CKPTREC=YES,WORKA=YES,SEPASMB=YES
         END
/*
 CLOSE SYSPCH,X'00D'
/&
// PAUSE
```

IBM — IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 8a

The user may optionally bypass the initialize routine. However, if initialization is required mount a pack on unit 190. Mount IBM-supplied tape on unit 180 (7-track drive). Place the following cards in the card reader. Dial 180 in the load unit switches and press load. When system enters the wait state press start and EOF on the card reader. Use of multi-part forms is suggested as copies of the linkage editor maps and the supervisor listing will be required by your IBM customer engineer for maintenance purposes.

```
// LOG
// JOB INTDSK
// DATE 69032
// ASSGN SYSLOG,X'01F',C1
// ASSGN SYSOPT,X'190',D1
// EXEC
// UID IR,C1
// VTOC STRTADR=(0199000),EXTENT=(9)
VOL1111111
// END
```

Figure 10. Example Two (Part 6 of 12)

IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 8b

To bypass initialize disk program on IBM-supplied tape, mount IBM-supplied tape on unit 180 (7-track drive). Mount initialized disk on unit 190 (VTOC on CYL 199). Insert the following control cards in the reader. Dial 180 in load unit switches. Press load key. When system enters wait state, press start and EOF on the card reader. Job is complete when 00C and 4000A messages are logged.

```
// LOG
// JOB INTDSK
// DATE 69032
// ASSGN SYSLOG,X'01F',C1
// FILES SYSIPT,1
```

IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 9

**** Caution **** Do not rewind unit 180. To load the tape onto disk, place the following cards in the reader. Dial 180 in load switches. Press load. When the system enters the wait state, press start and EOF on card reader. The following message appears on SYSLOG:
4444A. Type in 4B and press INTERRUPT to continue.

```
// JOB DISRST
// DATE 69032
// ASSGN SYS000,X'190',D1
// ASSGN SYSLOG,X'01F',C1
// ASSGN SYSLST,X'00E',L1
// EXEC
```

Figure 10. Example Two (Part 7 of 12)

IBM

IBM System 360 Assembler Coding Form

PROGRAM EXAMPLE TWO: ONE DISK DRIVE

PROGRAMMER STEP 10

After the system is loaded onto disk generate the installation system. Dial 190 in load unit switches. Press load key on console. When system enters the wait state, press start and EOF on the card reader. The following cards are in the card reader.

```
ADD X'00C',2540R
ADD X'00D',2540P
ADD X'00E',1403
ADD X'01F',1050A
ADD X'190',2311
ADD X'180',2400T7
```

IBM

IBM System 360 Assembler Coding Form

PROGRAM EXAMPLE TWO: ONE DISK DRIVE

PROGRAMMER STEP 10

```
SET DATE=02/01/69,CLOCK=00/00/00
ASSGN SYSLOG,X'01F'
LOG
ASSGN SYSRDR,X'00C'
ASSGN SYSIPT,X'00C'
ASSGN SYSLST,X'00E'
ASSGN SYSPCH,X'00D'
ASSGN SYSLNK,X'190'
ASSGN SYS001,X'190'
```

Figure 10. Example Two (Part 8 of 12)

IBM — IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE

PROGRAMMER: STEPS 11 and 12

```
// JOB DSERVRLB
// EXEC DSERV
 DSPLY ALL
/*
* THIS JOB DELETES ALL UNWANTED RELOCATABLE MODULES

// EXEC MAINT
 DELETR IJL.ALL,IJZ.ALL
 DELETR IJH.ALL,IJY.ALL,IJN.ALL,IJV.ALL

 DELETR IJKSYSA,IJKSYSI
 RENAMR IJXSYSA,IJKSYSA,IJXSYSI,IJKSYSI
/*

// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE          1111111',99/365,SD
// EXTENT SYSRES,111111,1,n,0001,nn9
// EXEC MAINT
   ALLOC CL=nn(n),RL=nn(n)
/*
/&
// PAUSE PRESS EOB TO CONTINUE
```

IBM — IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE

PROGRAMMER: STEP 13

Steps 13 and 14 contain the coding for including the IBM components selected for this example. The LINKEDIT book displayed and punched in step 4 contains all the necessary job control statements to linkage edit any of the IBM components shipped on the system. The user is encouraged to use this book to tailor a job stream to include any IBM components desired.

```
// JOB CATAL
// OPTION CATAL
   ACTION CLEAR
 INCLUDE

 Supervisor object deck here.

/*
```

Figure 10. Example Two (Part 9 of 12)

IBM — IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 13 (continued)

```
// EXEC LNKEDT
 INCLUDE IJBLBP                          CSERV
// EXEC LNKEDT
 INCLUDE IJBIPL                           IPL
// EXEC LNKEDT
 INCLUDE IJBLE                           LINKAGE EDITOR
// EXEC LNKEDT
 INCLUDE IJBSL1                          DSERV
// EXEC LNKEDT
 INCLUDE IJBSL2                          MAINT
// EXEC LNKEDT
 INCLUDE IJBSL3                          RSERV
// EXEC LNKEDT
 INCLUDE IJBSL4                          SSERV
// EXEC LNKEDT
 INCLUDE IJQD16DW                        DISK WORK FILE ASSEMBLER
// EXEC LNKEDT
/&
```

Need be included only if the operational volume will contain a source statement library.

Need not be linkage edited if the shipped supervisor is not exceeded.

IBM — IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 14

At this point the system will indicate re-IPL is needed. IPL form SYSRES (190).

```
SET DATE=02/01/69,CLOCK=00/00/00
// JOB CATAL
* AT THIS POINT LINKAGE EDIT COBOL, BASIC FORTRAN, PL/I, TAPE SORT/MERGE,
* UTILITIES
// OPTION CATAL
   ACTION CLEAR
 INCLUDE IJSCBD                          COBOL
// EXEC LNKEDT
 INCLUDE IJSDDB                          COBOL DEBUG
// EXEC LNKEDT
 INCLUDE IJTFO                           BASIC FORTRAN
// EXEC LNKEDT
 INCLUDE IJXPLID                         PL/I
// EXEC LNKEDT
 INCLUDE IJPSM                           TAPE SORT/MERGE
// LBLTYP TAPE
// EXEC LNKEDT
```

Figure 10. Example Two (Part 10 of 12)

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE

PROGRAMMER: STEP 14 (continued)

```
   INCLUDE IJWTP                         TAPE-TO-PRINTER UTILITY
   PHASE TPPR5,IJWTPCS2,NOAUTO
   INCLUDE IJWLAB
// LBLTYP TAPE
// EXEC LNKEDT
   INCLUDE IJWTC                         TAPE-TO-CARD UTILITY
   PHASE TPCD5,IJWTCCS2,NOAUTO
   INCLUDE IJWLAB
// LBLTYP TAPE
// EXEC LNKEDT
```

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE

PROGRAMMER: STEP 14 (continued)

```
   INCLUDE IJWCT                         CARD-TO-TAPE UTILITY
   PHASE CDTP5,IJWCTCS2,NOAUTO
   INCLUDE IJWLAB
// LBLTYP TAPE
// EXEC LNKEDT
   INCLUDE IJWCD                         CARD-TO-DISK UTILITY
   PHASE CDDK5,IJWCDCS2,NOAUTO
   INCLUDE IJWLAB
// EXEC LNKEDT
   INCLUDE IJWCP                         CARD-TO-PRINTER/PUNCH UTILITY
   PHASE CDPP5,IJWCPCS2,NOAUTO
   INCLUDE IJWLAB
// EXEC LNKEDT
```

Figure 10. Example Two (Part 11 of 12)

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 14 (continued)

```
        INCLUDE  IJWDC                          DISK-TO-CARD UTILITY
        PHASE    DKCD5,IJWDCCS2,NOAUTO
        INCLUDE  IJWLAB
// EXEC LNKEDT
        PHASE    TPCP,*,NOAUTO                  TAPE COMPARE UTILITY
        INCLUDE  IJWTCP
        INCLUDE  IJJCPD1
        INCLUDE  IJWXIT
        INCLUDE  IJWTPCP
// LBLTYP TAPE
// EXEC LNKEDT
/*
// PAUSE    CHECK LINKEDIT LISTINGS FOR ERRORS. IF CORRECT PRESS EOB.
```

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE TWO: ONE DISK DRIVE
PROGRAMMER: STEP 15

```
*  AT THIS POINT, DELETE THE COMPONENT PROGRAMS FROM THE RELOCATABLE
*  LIBRARY: COBOL, PL/I AND FORTRAN OBJECT SUBROUTINES REMAIN IN
*  RELOCATABLE LIBRARY AS WELL AS PREASSEMBLED MODULES.
// EXEC MAINT
        DELETR   IJS.ALL,IJTFO,IJTFO1,IJTFO2,IJTFO3,IJTFO4
        DELETR   IJQ.ALL,IJW.ALL,IJP.ALL,IJB.ALL,IJX.ALL
        CONDS    RL
/*
/&
// JOB CATAL
*  RELOAD 180 WITH THE TAPE CREATED IN STEP 7.
*  CHECK ASSEMBLY LISTING FOR ERRORS. IF CORRECT,
// PAUSE TYPE EOB TO CONTINUE
// ASSGN SYSIPT,X'180'
*  CATALOG MODULES TO RELOCATABLE LIBRARY
// EXEC MAINT
/*
/&                SYSTEM GENERATION COMPLETE
*  THE BASIC SYSGEN IS COMPLETE. THE USER MAY SET NEW STANDARD
*  LABELS, RE-ALLOCATE LIBRARY SIZES, AND SET AUTOMATIC CONDENSE
*  LIMITS IF REQUIRED.
*  COPY THE OPERATIONAL PACK TO TAPE FOR BACK-UP.
// PAUSE EOJ SYSTEM GENERATION
```

Figure 10. Example Two (Part 12 of 12)

IBM 2314 Direct Access Storage Facility

The system is supplied on two magnetic tapes. These two tapes contain DOS in a format suitable for restoring them to one 2316 disk pack.

Example three is one of the many possible methods for generating a system. Figure 11 illustrates the system configuration upon which Example Three (Figure 12) is based. The following steps are keyed to Example Three.

Before generating a system, users should have at least one initialized disk pack (VTOC on cylinder 199).

Step 1 Mount the IBM-supplied magnetic tapes and a disk pack to which the tapes are to be restored.

Before restoring the first tape, the user must take the necessary action to perform or bypass the initialize disk routine.

a. If the disk pack has not been initialized, the IBM-supplied volume, a self-loading tape, is capable of initializing the packs (see Figure 12). The DLBL cards in the following steps must contain the volume serial number appearing in the VOL card of the initialize disk control cards. Any volume serial number used in the EXTENT cards for the following steps must agree with the volume serial number in the VOL card for the initialize disk control cards.

b. If the disk pack was previously initialized, the initialize disk routine can be bypassed (see Figure 12). Any volume serial number used in the EXTENT cards for the following steps must agree with the volume serial number used when the packs were initialized.

Step 2 After step 1 is completed, the user must restore the system from the self-loading tapes to the initialized pack.

Step 3 When the tapes are restored, dial the address of the system residence disk drive into the load unit switches, and IPL to pass control to the DOS supervisor. See Appendix A for the IPL control statements.

Step 4 Perform a DSERV to display the directories. Then SSERV (display and punch) all the desired sample problems from the source statement library into cards. Punched output includes the sample problems with BKEND and CATALS cards. Four other books in the source statement library that should be punched out at this time (Z.LINKEDIT, Z.DELETECL, Z.DELETERL, and Z.DELETESL) contain the necessary control statements to selectively linkage edit and delete all IBM components. The sample problem program names, and the linkage edit and the delete book names can be chosen and punched into the DSPCH statement(s). The sample problem program names and the linkage edit and delete book names are shown in Appendix C.

Step 5 Define workfiles for SYSLNK, SYS001, SYS002, and SYS003. The workfiles SYSLNK, SYS001, SYS002, and SYS003 are defined by use of the DLBL and EXTENT cards. These cards must be preceded by the OPTION STDLABEL or OPTION PARSTD card.

Step 6 Perform all necessary assemblies. The assemblies for the supervisor, IOCS modules, and emulators should be performed as separate jobs. For the information required to assemble Emulator Programs, refer to the Emulator Program manual listed in the

Preface. The user must be careful to keep all assemblies in order.

Step 7 Assemble all user-required IOCS modules. By assigning SYSPCH to a tape unit, the IOCS modules can be cataloged to the relocatable library without punching them on cards. The IOCS modules required by COBOL, PL/I, and RPG, as defined in <u>Appendix B</u>, are supplied in the relocatable library by IBM.

Close the tape assigned to SYSPCH and reassign SYSPCH to its permanent assignment by using the CLOSE command. The assembly listings should be checked for errors before proceeding.

Step 8 Linkage edit and catalog the assembled supervisor (object module from step 6) to the core image library. If the SEND address is larger than the one used by the supervisor being replaced, certain key programs must also be linkage edited and cataloged to the core image library in the same job step with the new supervisor. These key programs are IPL, linkage editor, and librarian. The LINKEDIT deck punched out in step 4 contains all the necessary control statements to linkage edit all IBM components shipped on the system. If the SEND address is not exceeded, only the supervisor is cataloged. <u>The new supervisor is not cataloged until a /& statement is read</u>. The user must not attempt any other operation from the time the supervisor and these preceding programs are cataloged until IPL time.

Step 9 Re-IPL and set the date (and clock if the Timer Feature is present).

Linkage edit and catalog any additional components desired to the core image library. See <u>Appendix C</u> for a complete list of control cards for all IBM components to be cataloged. Before the next step is performed, check the linkage editor listings, and make all necessary corrections.

Step 10 Reload the tape that was assigned to SYSPCH in step 7 and assign it to SYSIPT. With this tape the MAINT program catalogs the IOCS modules to the relocatable library by the control card // EXEC MAINT. The user may set new standard labels (OPTION STDLABEL), reallocate library sizes, and set automatic condense limits, if required. Backup for the operational disk can be obtained by copying the operational disk pack to tape by using the copy disk-to-tape utility program.

INPUT/OUTPUT DEVICE CONFIGURATION FOR EXAMPLE THREE (SEE FIGURE 11)

Device	Channel	Unit	Use
2540R	0	0C	Card Reader (SYSRDR, SYSIPT)
2540P	0	0D	Card Punch (SYSPCH)
1403	0	0E	Printer (SYSLST)
1052	0	1F	Printer-Keyboard (SYSLOG)
2314	1	30	Disk (SYSRES, SYSLNK, SYS001, SYS002, SYS003, SYS004)
2400T9	1	80	Magnetic Tape

Example Three: Results of System Generation

When system generation is completed for Example One (see Figure 12), the operational disk of the installation contains: Assembler, Basic FORTRAN, COBOL, RPG, PL/I, and the disk sort/merge programs in its core image library, together with the installation's tailored supervisor, job control, linkage editor, and librarian programs. The tapes shipped by IBM are retained as backup tapes. They are self-loading tapes capable of being restored on disk.

The sample problems are punched out during step 4.

Libraries contain all modules and macro definitions shipped from IBM.

The core image library is built to contain those IBM programs chosen by the user.

At this point, the libraries are condensed, and user IOCS modules are assembled and cataloged to the relocatable library.

When system generation is completed, the sample problems should be run against the operational pack to ensure correct creation of all system programs.

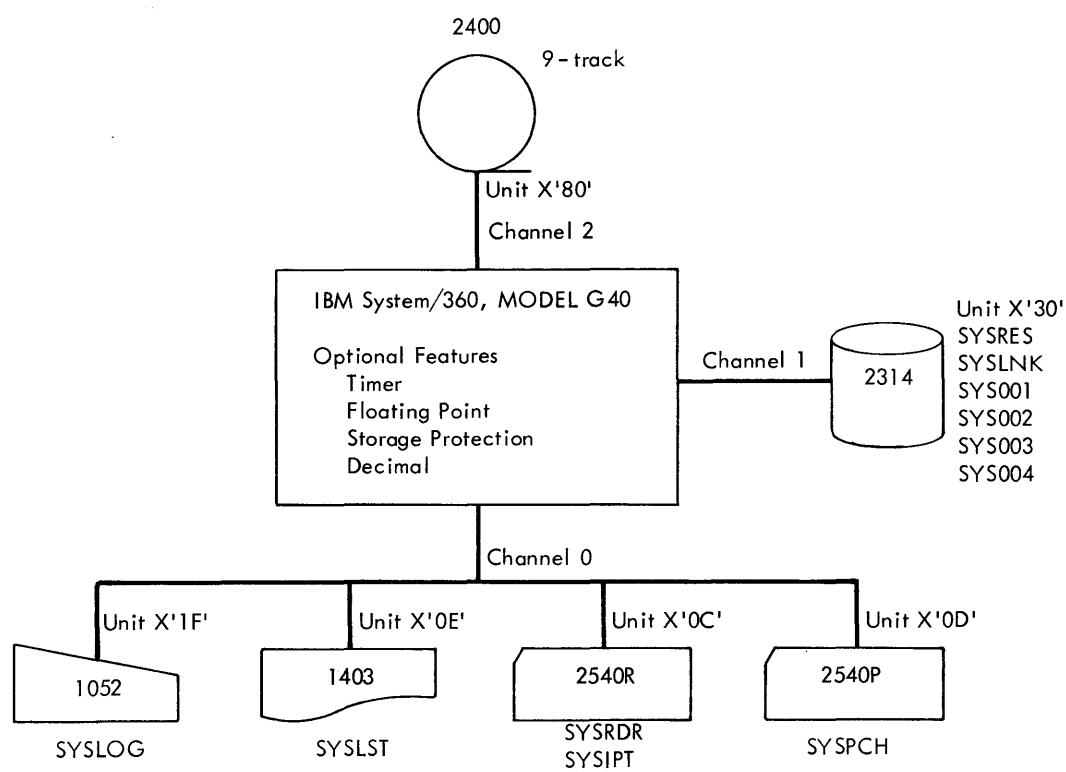


Figure 11. System Configuration for Example Three

IBM IBM System 360 Assembler Coding Form

PROGRAM EXAMPLE THREE: 2314 DIRECT ACCESS STORAGE

PROGRAMMER STEP 1a

The user may optionally bypass the initialized routine. However, if initialization is required mount a pack on unit 130. Mount IBM-supplied tape on unit 180 (9-track drive). Place the following cards in the card reader. Dial 180 in the load unit switches and press load. When system enters the wait state press start and EOF on the card reader. Use of multi-part forms is suggested as copies of the linkage editor maps and the supervisor listing will be required by your IBM customer engineer for maintenance purposes.

```
// JOB INTDSK
// DATE 69032
// ASSGN SYSLOG,X'01F',C1
// ASSGN SYSOPT,X'130',D3
// EXEC
// UID IR,C1
// VTOC STRTADR=(0199000),EXTENT=(20)
VOL1111111
//END
```

IBM IBM System 360 Assembler Coding Form

PROGRAM EXAMPLE THREE: 2314 DIRECT ACCESS STORAGE

PROGRAMMER STEP 1b

To bypass initialize disk program on IBM-supplied tape. Mount IBM-supplied tape on unit 130 (9-track drive). Mount initialized disk on unit 130 (VTOC on cyl 199). Insert the following control cards in the reader. Dial 130 in load unit switches. Press load key. When system enters wait state, press start and EOF on the card reader. Job is complete when 00C and 4000A messages are logged.

```
// JOB INTDSK
// DATE 69032
// ASSGN SYSLOG,X'01F',C1
// FILES SYSIPT,1
```

Figure 12. Example Three (Part 1 of 7)

IBM IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE THREE: 2314 DIRECT ACCESS STORAGE
PROGRAMMER: STEP 2

**** Caution **** Do not rewind unit 180. To load the tape onto disk, place the following cards in the reader. Dial 180 in load switches. Press load. When the system enters the wait state, press start and EOF on card reader. The following message appears on SYSLOG:
4444A. Type in 4 Ⓑ and press INTERRUPT to continue.

```
// JOB DISRST
// DATE 69032
// ASSGN SYS000,X'130',D3
// ASSGN SYSLOG,X'01F',C1
// ASSGN SYSLST,X'00E',L1
// EXEC
```

IBM IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE THREE: 2314 DIRECT ACCESS STORAGE
PROGRAMMER: STEP 3

After the system is loaded onto disk generate the installation system. Dial 130 in load unit switches. Press load key on console. When system enters the wait state, press start on the card reader. The following cards are in the card reader (SYSRDR/SYSIPT).

```
ADD X'00C',2540R
ADD X'00D',2540P
ADD X'00E',1403
ADD X'01F',1050A
ADD X'130',2314
ADD X'180',2400T9
SET DATE=02/01/69,CLOCK=00/00/00
ASSGN SYSLOG,X'01F'
LOG
ASSGN SYSRDR,X'00C'
ASSGN SYSIPT,X'00C'
ASSGN SYSPCH,X'00D'
ASSGN SYSLST,X'00E'
ASSGN SYS001,X'130'
ASSGN SYS002,X'130'
ASSGN SYS003,X'130'
ASSGN SYSLNK,X'130'
```

Figure 12. Example Three (Part 2 of 7)

IBM — IBM System 360 Assembler Coding Form

PROGRAM EXAMPLE THREE: 2314 DIRECT ACCESS STORAGE
PROGRAMMER STEP 4

```
// JOB DSERV
// EXEC DSERV
 DSPLY ALL
/*


// EXEC MAINT
 DELETR IJL.ALL,IJN.ALL
 DELETR IJP.ALL,IJY.ALL,IJZ.ALL

 DELETR IJKSYSA,IJKSYSI
 RENAMR IJXSYSA,IJKSYSA,IJXSYSI,IJKSYSI
/*
/&
```

IBM — IBM System 360 Assembler Coding Form

PROGRAM EXAMPLE THREE: 2314 DIRECT ACCESS STORAGE
PROGRAMMER STEP 5

```
// JOB LABEL
// OPTION DECK,LIST,LOG,STDLABEL
// DLBL IJSYSLN,'SYSTEM WORK FILE NO. 0',99/365,SD
// EXTENT SYSLNK,nnnnnn,1,n,nnnn,nnnn
// DLBL IJSYS01,'SYSTEM WORK FILE NO. 1',99/365,SD
// EXTENT SYS001,nnnnnn,1,n,nnnn,nnnn                      Assign work files.
// DLBL IJSYS02,'SYSTEM WORK FILE NO. 2',99/365,SD
// EXTENT SYS002,nnnnnn,1,n,nnnn,nnnn
// DLBL IJSYS03,'SYSTEM WORK FILE NO. 3',99/365,SD
// EXTENT SYS003,nnnnnn,1,n,nnnn,nnnn


/&
```

Figure 12. Example Three (Part 3 of 7)

IBM IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE:THREE: 2314 DIRECT ACCESS STORAGE

PROGRAMMER: STEP 6

```
// JOB SUPVR
// EXEC ASSEMBLY
         SUPVR SYSTEM=DISK,MPS=YES
         CONFG MODEL=40,SP=YES,DEC=YES,FP=YES,TIMER=YES
         STDJC LISTX=YES,LINES=46
         FOPT  IT=BG,PC=YES,OC=YES,CCHAIN=YES,DASDFP=(1,2,2314),          X
               SYSFIL=(2314),TEB=4
         PIOCS TAPE=9,BMPX=YES
         ALLOC F1=6K,F2=6K
         IOTAB JIB=10,F1PGR=8,F2PGR=8,IODEV=10,BGPGR=10
         DVCGEN CHUN=X'00C',DVCTYP=2540R
         DVCGEN CHUN=X'00D',DVCTYP=2540P
         DVCGEN CHUN=X'00E',DVCTYP=1403
         DVCGEN CHUN=X'01F',DVCTYP=1050A
         DVCGEN CHUN=X'130',DVCTYP=2314
         DVCGEN CHUN=X'180',DVCTYP=2400T9
```

IBM IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE THREE: 2314 DIRECT ACCESS STORAGE

PROGRAMMER: STEP 6 (continued)

```
         ASSGN SYSRDR,X'00C'
         ASSGN SYSIPT,X'00C'
         ASSGN SYSPCH,X'00D'
         ASSGN SYSLST,X'00E'
         ASSGN SYSLOG,X'01F'
         ASSGN SYSLNK,X'130'
         ASSGN SYS001,X'130'
         ASSGN SYS002,X'130'
         ASSGN SYS003,X'130'
         SEND  10240
 END
/*
*    CHECK ASSEMBLY LISTING FOR ERRORS. IF CORRECT
*  REMOVE ASSEMBLED SUPERVISOR FROM SYSPCH. INSERT IN READER
*  FOLLOWING THE INCLUDE CARD OF JOB CATALSUP
// PAUSE TO CONTINUE PRESS EOB

/&
```

Figure 12. Example Three (Part 4 of 7)

IBM — IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE THREE: 2314 DIRECT ACCESS STORAGE
PROGRAMMER: STEP 7

```
* AT THIS POINT ASSEMBLE IOCS MODULES TO BE CATALOGED TO THE
* RELOCATABLE LIBRARY
* UNIT 180 MUST BE UNASSIGNED AT THIS POINT.
* THE FOLLOWING IS AN EXAMPLE. MODULES SHOULD BE ASSEMBLED ON AN
*     AS REQUIRED BASIS.
// JOB ASSEM
// OPTION DECK,LIST,LOG
// ASSGN SYSPCH,X'181'
// EXEC ASSEMBLY
    CDMOD         RECFORM=FIXUNB,                                      X
                  CTLCHR=ASA,                                          X
                  TYPEFLE=OUTPUT,                                      X
                  IOAREA2=YES,                                         X
                  DEVICE=2540R,                                        X
                  SEPASMB=YES
  END
/*
// EXEC ASSEMBLY
     MTMOD        RECFORM=FIXUNB,                                      X
                  READ=FORWARD,                                        X
                  CKPTREC=YES,                                         X
                  SEPASMB=YES
  END
/*
```

IBM — IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE THREE: 2314 DIRECT ACCESS STORAGE
PROGRAMMER: STEP 7 (continued)

```
 CLOSE SYSPCH,X'00D'
/&
* CHECK ASSEMBLY LISTINGS FOR ERRORS. RELOAD 180 WITH SAME TAPE.
// PAUSE IF CORRECT PRESS EOB TO CONTINUE
```

Figure 12. Example Three (Part 5 of 7)

IBM — IBM System 360 Assembler Coding Form

PROGRAM EXAMPLE THREE: 2314 DIRECT ACCESS STORAGE
PROGRAMMER STEP 8

```
// JOB CATALSUP
// OPTION CATAL
  ACTION CLEAR
 INCLUDE

 Supervisor object deck here.

/*
// EXEC LNKEDT
 INCLUDE IJBLBP                    CSERV
// EXEC LNKEDT
 INCLUDE IJBIPL                    IPL
// EXEC LNKEDT
 INCLUDE IJBLE                     LINKAGE EDITOR
// EXEC LNKEDT
 INCLUDE IJBSL1                    DSERV
// EXEC LNKEDT
 INCLUDE IJBSL2                    MAINT
// EXEC LNKEDT
 INCLUDE IJBSL3                    RSERV
// EXEC LNKEDT
```

Steps 8 and 9 contain the coding for including the IBM components selected for this example. The LINKEDIT book displayed and punched in step 4 contains all the necessary job control statements to linkage edit any of the IBM components shipped on the system. The user is encouraged to use this book to tailor a job stream to include any IBM components desired.

Need not be linkage edited if the shipped supervisor is not exceeded.

IBM — IBM System 360 Assembler Coding Form

PROGRAM EXAMPLE THREE: 2314 CIRECT ACCESS STORAGE
PROGRAMMER STEP 9

```
 INCLUDE IJBSL4                    SSERV
// EXEC LNKEDT
 INCLUDE IJBSL5                    CORGZ
// EXEC LNKEDT
/&

 At this point the system will indicate re-IPL is needed. IPL from SYSRES (130)

 SET DATE=02/01/69,CLOCK=00/00/00
```

Need not be linkage edited if the shipped supervisor is not exceeded.

Figure 12. Example Three (Part 6 of 7)

IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE THREE: 2314 DIRECT ACCESS STORAGE

PROGRAMMER: STEP 9 (continued)

```
// JOB CATALCLB
* AT THIS POINT LINKAGE EDIT COBOL, BASIC FORTRAN,
* PL/I, RPG, DISK SORT MERGE, 14K ASSEMBLER
// OPTION CATAL
 ACTION CLEAR
 INCLUDE IJSCBD                    COBOL
// EXEC LNKEDT
 INCLUDE IJSDDB                    COBOL DEBUG
// EXEC LNKEDT
 INCLUDE IJTFO                     BASIC FORTRAN
// EXEC LNKEDT
 INCLUDE IJXPLID                   PL/I
// EXEC LNKEDT
 INCLUDE IJRRG                     RPG
// EXEC LNKEDT
 INCLUDE IJOSM                     SORT/MERGE
// EXEC LNKEDT
 INCLUDE IJQD32                    14K ASSEMBLER
// EXEC LNKEDT
/*
/&
```

IBM System 360 Assembler Coding Form

PROGRAM: EXAMPLE THREE: 2314 DIRECT ACCESS STORAGE

PROGRAMMER: STEP 10

```
// JOB CATALRLB
// ASSGN SYSIPT,X'180'
* CATALOG MODULES TO RELOCATABLE LIBRARY
// EXEC MAINT
/*
* THE FOLLOWING DELETIONS ARE OPTIONAL.
// PAUSE PLACE REMAINING CARDS IN READER. PRESS EOB TO CONTINUE
// EXEC MAINT
 Place any desired deletes here.

 CONDS RL,SL
/*
/&
* THE BASIC SYSGEN IS COMPLETE. THE USER MAY SET NEW STANDARD
* LABELS, RE-ALLOCATE LIBRARY SIZES, AND SET AUTOMATIC CONDENSE
* LIMITS IF REQUIRED. COPY THE OPERATIONAL PACK TO TAPE FOR BACK-UP
* USE THE IDENTICAL UTILITY DECK SETUP AS AFTER FIRST SYSGEN JOB.
* THE FINAL CONDENSE OPERATION SUPPLIED THE CURRENT STATUS
* OF THE SYSTEM LIBRARIES.
// PAUSE EOJ SYSGEN....
```

Figure 12. Example Three (Part 7 of 7)

Maintenance Procedures

The maintenance of Disk Operating System libraries can be performed with the MERGE function of the CORGZ librarian program. The MERGE function allows complete libraries to be merged into one library, or entire components, selected individual phases, modules, or books to be transferred from one existing library to another. When the MERGE function is used, space availability is always a consideration.

Using the MERGE Function of the CORGZ Librarian Program

The MERGE function provides the facility to replace identical entries of a library, or add unique entries to a library. In either case, executing a copy (MERGE) requires that space be available in the library to accommodate the added information.

REPLACING IDENTICAL ENTRIES USING MERGE FUNCTION

When transferring information that is named identically as existing entries of a library, the old entry (phase, module, or book name) is deleted from the library's directory, and the name of the new entry is added to the end of the library's directory. The new entry is the name of the new information. The information itself (phase, module, or book) is added to the end of the library.

ADDING UNIQUE ENTRIES USING MERGE FUNCTION

When uniquely named entries (phase, module, or book name) are transferred to an existing library, the names of the entries are added to the end of the library's directory, and the information itself is added to the end of the library.

General Library Updating Techniques Using the MERGE Function of the CORGZ Librarian Program

The techniques presented here apply to the examples given. They are intended as a guide for the user, and do not necessarily satisfy all user requirements. The method that a user applies to maintain his libraries will depend upon the user's library structure, and his special requirements concerning the maintenance of his library.

COPYING SELECTIVELY TO MERGE LIBRARIES

When there are many more entries desired in an existing library than another, it is usually faster to selectively copy (MERGE) to the library containing the most number of desired entries. Thus, the number of entries transferred are kept to a minimum to save time. Copying to a library necessarily expands the library. If there is insufficient space in a library to accommodate additional entries, unwanted entries can be deleted, the library condensed, and the new entries then added to the library.

DELETING UNWANTED ENTRIES AND MERGING AN ENTIRE LIBRARY WITH ANOTHER LIBRARY

A second technique is to apply the IBM-supplied DELETERL or DELETESL book to delete all unwanted entries from a library and copy the entire library to another library, thus merging the two libraries. Using this technique requires that the library being copied to contains enough space to accommodate the entire library being copied. If there is insufficient space, the library being copied to can be condensed or reallocated.

COPYING SELECTIVELY, MERGING TWO LIBRARIES TO CREATE A THIRD LIBRARY

If there is insufficient space to accommodate a merge of two libraries, a third library can be created to contain selected entries from the two libraries being merged. This technique eliminates the need for condensing or reallocating an existing library. Note that this technique can be applied by using only two disk drives.

DELETING UNWANTED ENTRIES FROM TWO LIBRARIES AND MERGING BOTH LIBRARIES TO CREATE A THIRD LIBRARY

Another technique for merging two libraries is to delete unwanted entries from the two existing libraries, and merging the two libraries, in their entirety, by copying them to create a third library. Note that this technique can be applied by using only two disk drives.

Conventions for Merge Examples

The following conventions are used in the maintenance examples given here. If a macro definition called m is an entry in a library called L (L being any library and m being any macro definition entry in any library L) then any macro definition entry m in any library L can be designated as Lm.

Assume that there are two libraries L called A and B, and that the macro definition entries 1, 2, and 3 in library A are IBM-supplied, and 8 and 9 are user-defined macro definitions. Further, assume that the user receives a new release of DOS with a source statement library, called library B, containing IBM-supplied macro definitions 1, 2, 3, 4, and 5. Thus, library A and library B can be represented as follows:

Library A-- A1, A2, A3, A8, A9

Library B-- B1, B2, B3, B4, B5

For the libraries defined, macro-definition entries 1, 2, and 3 in library B are identical in name to 1, 2 and 3 in library A, but a later version or update. Macro definitions 4 and 5 in library B are new IBM-supplied macro definitions for the release received, and macro definitions 8 and 9 are user defined. The structures of the libraries defined here are used in the examples that follow.

Maintenance Examples Using MERGE Function

For each example that follows:

- The macro definitions updated are selected arbitrarily,
- Library A is assumed to reside on disk drive A, and library B on disk drive B,
- Library A is assumed to be the existing user library, and library B the IBM-supplied maintenance update (new release),
- The techniques described can be applied by using only two disk drives.

The following examples illustrate several ways to update an existing library.

EXAMPLE 1A: COPYING SELECTIVELY TO MERGE TWO LIBRARIES

If the majority of the macro definitions desired are in library A, and it is desired to update macro definitions A2 and A3, selectively copy (merge) macro definitions B2 and B3 from IBM-supplied library B to library A. Thus, when there are many more macro definitions in one library than the other, it is faster to selectively copy to the library with the greatest number of entries, if there is enough space in the library being enlarged. If there is not enough space in library A, the unwanted macro definitions A2 and A3 must first be deleted from library A, and library A condensed before the transfer can be attempted.

A representation of the results of selectively copying macro definitions B2 and B3 from library B to library A follows:

Library A-- A1, __, __, A8, A9, B2, B3

Library B-- B1, B2, B3, B4, B5

Note that the macro definitions B2 and B3 are added to the end of the library updated (library A).

EXAMPLE 1B: VARIATION OF SELECTIVE COPYING TO MERGE TWO LIBRARIES

A variation of the technique described in Example 1A demonstrates the capability of transferring in any direction. Thus, if the majority of macro definitions are in library B, and it is desired to update macro definitions A1, A2, and A3, selectively copy macro definitions A8 and A9 from library A to library B.

A representation of selectively copying macro definitions A8 and A9 from library A to library B follows:

Library A-- A1, A2, A3, A8, A9

Library B-- B1, B2, B3, B4, B5, A8, A9

Note that macro definitions A8 and A9 are added to the end of the library updated (library B).

EXAMPLE 2A: DELETING UNWANTED ENTRIES FROM A LIBRARY AND MERGING LIBRARIES BY COPYING ONE TO THE OTHER

Assuming that the majority of macro definitions are in library A, an alternate method of accomplishing an update similar to that described in Example 1A follows.

Delete macro definitions B1, B4, and B5 from library B, and completly copy library B to library A. Thus, the IBM-supplied DELETESL book can delete all unwanted macro definitions from the source statement library, and the library transferred in its entirety.

A representation of the results of deleting macro definitions B1, B4, and B5 from library B, and copying library B follows:

Library A-- A1, __, __, A8, A9, B2, B3

Library B-- __, B2, B3, __, __

Note that library B is added to the end of library A, and that library A in Example 1A is identical to library A in this example.

EXAMPLE 2B: VARIATION OF DELETING UNWANTED ENTRIES FROM A LIBRARY AND COPYING (MERGING) ONE TO THE OTHER

A variation of the technique described in Example 2A that demonstrates the capability of transferring in any direction follows.

Thus, to add macro definitions A8 and A9 to library B, delete macro definitions A1, A2, and A3 in library A and then completely copy library A to library B. Again, the IBM-supplied DELETESL book can delete all unwanted IBM-supplied macro definitions from the source statement library.

A representation of the results of deleting macro definitions A1, A2, and A3 from library A and copying library A to library B follows.

Library A-- __, __, __, A8, A9

Library B-- B1, B2, B3, B4, B5, A8, A9

Note that user macro-definitions A8 and A9 are added to the end of the library being updated (library B), and library B in Example 1B is identical to library B in this example.

EXAMPLE 3: COPYING SELECTIVELY, MERGING TWO LIBRARIES TO CREATE A THIRD LIBRARY

If space is a problem, a third library called C can be created to be a combination of selected macro definitions from both library A and library B. Thus, if the desired macro definitions in library A are A8 and A9 (the user's macro definitions), and B1, B2, and B3 in library B (IBM-supplied), then user macro definitions A8 and A9 can be selectively copied (merged) from library A to library C, and IBM-supplied macro definitions B1, B2, and B3 can be selectively copied (merged) from library B to library C. The advantage of this technique is that the need for a condense or reallocation of an existing library is eliminated, and time is saved. However, the time required to initialize a third disk pack must be taken into consideration. A representation of the results of selectively copying library A and library B (merging) to create library C follows:

Library A-- A1, A2, A3, A8, A9

Library B-- B1, B2, B3, B4, B5

Library C-- B1, B2, B3, A8, A9

Note that library C requires less space than if either library A or library B were merged with each other.

EXAMPLE 4: DELETING UNWANTED ENTRIES FROM TWO LIBRARIES AND MERGING BOTH LIBRARIES TO CREATE A THIRD LIBRARY

A variation of selectively copying from two libraries to form a third library follows.

To retain user macro definitions A8 and A9 in library A, and update IBM-supplied macro definitions A1, A2, and A3 in library A, delete A1, A2, and A3 from library A, and B4 and B5 from library B. Then completely copy (merge) library A and library B to create a third library C.

A representation of the results of deleting unwanted macro definitions from library A and library B, and copying (merging) both libraries to create a third library C follows:

Library A-- __, __, __, A8, A9

Library B-- B1, B2, B3, __, __

Library C-- B1, B2, B3, A8, A9

Note that library C requires less space than if either library A or B were merged with each other. Also note that library C in Example 3 is identical to library C in this example.

Considerations for Merging

The DIBL and EXTENT file definition statements must precede the MERGE control statement. When defining files, use the following rules:

In merging to or from a modified, or duplicate system residence file, the modified or duplicate file name must be IJSYSRS, the logical unit must be SYS002, and the file ID (identification) must be identical to the ID supplied when the file was created.

In merging to a private relocatable library file, the file name must be IJSYSRL, the logical unit must be SYSRLB, and the file ID must be identical to the ID supplied when the file was created.

In copying from a private relocatable library file, the file name must be IJSYSPR, the logical unit must be SYS001, and the file ID must be identical to the ID supplied when the file was created.

In merging to a private source statement library file, the file name must be IJSYSSL, the logical unit must be SYSSLB, and the file ID must be identical to the ID supplied when the file was created.

In copying from a private source statement library file, the file name must be IJSYSPS, the logical unit must be SYS000, and the file ID must be identical to the ID supplied when the file was created. The file name, logical unit, and direction of transfer for each of the MERGE operations is indicated in the Figure 13. Any combination of the indicated operations can be performed in one job step.

File Name Logical unit	IJSYSRS SYSRES	IJSYSRS SYS002	IJSYSRL SYSRLB	IJSYSPR SYS001	IJSYSSL SYSSLB	IJSYSPS SYS000
Merge RES to NRS	from	to				
Merge NRS to RES	to	from				
Merge RES to PRV	from		to		to	
Merge NRS to PRV		from	to		to	
Merge PRV to RES	to			from		from
Merge PRV to NRS		to		from		from
Merge PRV to PRV			to	from	to	from

Figure 13. Merge Operation File Name and Logical Unit Identification

Diagnostic messages for erroneous assignments, file definitions, etc., are provided on SYSLST.

The following is an example of a job set up for the MERGE function. The sections of the job are bracketed and numbered 1 through 8. The explanations that follow the job are keyed to the job sections. The example assumes two disk drives with addresses of 190 and 191 that:

1. the MERGE function of the CORGZ librarian program is on SYSRES which is on 190 by virtue of an IPL

2. SYSRLB and SYSSLB are assigned to 191, SYS000 and SYS001 to 190, and SYS002 to 191.

```
// JOB EXAMPLE

// ASSGN SYSRLB,X'191'
```

```
  // ASSGN SYSSLB,X'191'

  // ASSGN SYS000,X'190'

  // ASSGN SYS001,X'190'

  // ASSGN SYS002,X'191'

1 [// DLBL IJSYSRL,'PRIVATE RL',69/365
  [// EXTENT SYSRLB,111111,1,0,1500,10

2 [// DLBL IJSYSSL,'PRIVATE SL',69/365
  [// EXTENT SYSSLB,111111,1,0,1600,10

3 [// DLBL IJSYSPR,'PRIVATE RL TEST',69/365
  [// EXTENT SYS001,111111,1,0,1300,10

4 [// DLBL IJSYSPS,'PRIVATE SL TEST',69/365
  [// EXTENT SYS000,111111,1,0,1400,10

5 [// DLBL IJSYSRS,'SYSTEM RESIDENCE',69/365
  [// EXTENT SYS002,111111,1,0,1,170

  // EXEC CORGZ

  [NEWVOL RL=10(2),SL=10(2)
6 [COPYR ALL
  [COPYS ALL

  [MERGE PRV,PRV
7 [COPYR ALL
  [COPYS ALL

  [MERGE NRS,PRV
8 [COPYR ALL
  [COPYS ALL

  /*
  /&
```

The following explanations are keyed to the sections of the job:

1. File definition statements for a private relocatable library file which is created and updated.

2. File definition statements for a private source statement library file which is created and updated.

3. File definition statements for a private relocatable library file from which modules are copied.

4. File definition statements for a private source statement library file from which books are copied.

5. File definition statements for a modified, or duplicate system residence file from which modules and books are copied. Note that this file is the old SYSRES (with user programs).

6. Creates private relocatable and source statement libraries on SYSRLB and SYSSLB and copies the relocatable and source statement libraries from the system residence file on SYSRES into them.

7. Merges all modules and books from private relocatable and source statement libraries on SYS001 and SYS000 into the appropriate private libraries created on SYSRLB and SYSSLB.

8. Merges all modules and books from the relocatable and source statem libraries of a modified, or duplicate system residence file on SYS0 into private libraries created on SYSRLB and SYSSLB.

For a more detailed description of the MERGE function see the System Control and System Service manual listed on the front cover.

Disk Operating System Sample Problems

The sample problems provided with the Disk Operating System (DOS) demonstrate to the user, particularly the operator, each component of the programming system. Although the problems are general and illustrative rather than detailed and exhaustive, they nevertheless serve as a minimal test of each user's programming system. The user's programming system is built from the more general system supplied by IBM.

The sample problems are designed to be run on a minimum system configuration including at least one 2311 disk drive (the system residence volume), a card reader/punch, printer, and a 1052 Printer-Keyboard. The minimum background partition storage capacity required for each sample problem is given in <u>Background Partitions Storage Requirements for Disk Operating System IBM-Supplied Programs</u>.

No data is required for any of the sample problems, except RPG, PL/I, Autotest, and multiprogramming. The requirements for each problem are described in the examples that follow.

All sample problems, except those for multiprogramming, are included in the IBM-supplied core image and source statement library volume. The multiprogramming sample problem is prepared by the user. Those sample problems included in the source statement library volume are retrieved as card decks by the SSERV librarian program. Each card deck is either a source program or a set of control cards. Individual decks are preceded by a CATALS card and a BKEND card and followed by a BKEND card. Once the sample problems have been retrieved, they can be deleted from the user's operational system disk during system generation.

Retrieving the Sample Problems

The sample problems are retrieved from the second volume (core image and source statement library volume) during system generation. Although the order for retrieving the problems from the disk is optional, it is recommended that they be retrieved in the order of intended execution. This facilitates the task of preparing the job stream. The following job steps are necessary to retrieve all of the sample problems:

```
// JOB SAMPLPRB
// EXEC SSERV
   DSPCH Z.FO1,Z.CB1,Z.RG1,Z.PL1,Z.AS1
   DSPCH Z.SM1,Z.UTTPPR1,Z.AS2,Z.SM2
   DSPCH Z.UTDKPR1,Z.AS3,Z.UTDCPR1
   DSPCH Z.AS4,Z.SM4,Z.UTTPPR2
   DSPCH Z.AS5,Z.SM5,Z.UTDKPR2
   DSPCH Z.AS6,Z.SM6,Z.UTDKPR3
   DSPCH Z.MCR1,Z.MCR2,Z.EU3SPRGM,Z.EU4SPRGM
   DSPCH Z.AT1,Z.ORDC,Z.ORJT,Z.VFU1
/&
```

If you do not wish to retrieve a particular problem from the source statement library volume, delete the appropriate operand from the preceding DSPCH statement. All of the sample problems are in the Z sub-library and all of that sub-library can be retrieved with the DSPCH Z.ALL statement.

Card columns 73-80 in each sample problem deck (except RPG) contain an identification number and a sequence number. The identification number for the RPG sample problem deck is punched in columns 75-78. Card columns 1-5 contain the sequence number for RPG. Although these numbers can be used to identify individual sample problems, it is recommended that the cards be machine-interpreted.

Figure 14 shows the arrangement and identification and sequence numbers of each of the sample problem card decks produced by the SSERV librarian program during system generation. These decks are punched in the order in which they are retrieved from the source statement library volume (preferably the intended order of program execution). To prepare each sample problem for execution, the user must:

- Remove all CATALS and BKEND cards shown in Figure 14. If the RPG or PL/I sample problem is retrieved, remove the *END SOURCE DECK/BEGIN INPUT CARDS card from the programs.

- Punch and insert the necessary job control cards for each program to be run, as shown in the examples.

The programs are compiled or assembled, linkage edited, and executed with a minimum of operator intervention.

Cards Retrieved		Card Cols. 73 - 76	Card Cols. 77 - 80
CATALS Z.FO1 BKEND Z.FO1 FORTRAN Source Deck (22 cards) BKEND	FORTRAN Sample Problem	$451	0001 - 0022
CATALS Z.ILFSAMPL BKEND Z.ILFSAMPL FORTRAN Source Deck (22 Cards) BKEND	FORTRAN IV Sample Problem	$479	0001 - 0022
CARALS Z.CB1 BKEND Z.CB1 COBOL Source Deck (81 cards) BKEND	COBOL Sample Problem	$452	0001 - 0081
CATALS Z.RG1 BKEND Z.RG1 RPG Source Deck (45 Cards) * END SOURCE DECK/BEGIN INPUT DATA RPG Data (13 Cards) BKEND	RPG Sample Problem	$460	
CATALS Z.PL1 BKEND Z.PL1 PL/1 Source Deck (38 Cards) * END SOURCE DECK/BEGIN INPUT DATA PL/I Data (1 Card) BKEND	PL/I Sample Problem	$464	0001 - 0038
CATALS Z.AS1 BKEND Z.AS1 Assembler Source Deck (23 Cards) BKEND	Assembler Sample Problem 1 (Generate input for Tape Sort/Merge	$465	0001 - 0023
CATALS Z.SM1 BKEND Z.SM1 S/M Control Cards (6 Cards) BKEND	Sort/Merge Sample Problem (Generate input for Tape-to-Printer Utility)	$400	0001 - 0006
CATALS Z.UTTPPR1 BKEND Z.UTTPPR1 T-P Control Cards (2 Cards) BKEND	Tape-to-Printer Utility Sample Problem	$462	0001 - 0002
CATALS Z.AS2 BKEND Z.AS2 Assembler Source Deck (23 Cards) BKEND	Assembler Sample Problem 2 (Generate input for Disk Sort/Merge)	$465	0101 - 0123
CATALS Z.SM2 BKEND Z.SM2 S/M Control Cards (6 Cards) BKEND	Disk Sort/Merge Sample Problem (Generate input for Disk-to-Printer Utility)	$450	0001 - 0006
CATALS Z.UTDKPR1 BKEND Z.UTDKPR1 D-P Control Cards (2 Cards) BKEND	Disk-to-Printer Utility Sample Problem	$461	0001 - 0002

Figure 14. Sample Problem Card Decks Retrieved from IBM-Supplied Core Image and Source Statement Library Volumes (Part 1 of 3)

Cards Retrieved		Card Cols. 73 - 76	Card Cols. 77 - 80
CATALS Z.AS3 BKEND Z.AS3 Assembler Source Deck (23 Cards) BKEND	Assembler Sample Problem 3 (Generate input for Data Cell-to-Printer Utility)	$465	0201 - 0223
CATALS Z.UTDCPR1 BKEND Z.UTDCPR1 DC-P Control Cards (2 Cards) BKEND	Data Cell-to-Printer Utility Sample Problem	$463	0001 - 0002
CATALS Z.AS4 BKEND Z.AS4 Assembler Source Deck (23 Cards) BKEND	Assembler Sample Problem 4 (Generate input for Tape and Disk Sort/Merge)	$465	0402 - 0423
CATALS Z. SM4 BKEND Z.SM4 S/M Control Cards (6 Cards) BKEND	Tape and Disk Sort/Merge 2400 Sample Problem (Generate input for Tape-to-Printer Utility)	$483	0001 - 0006
CATALS Z.UTTPPR2 BKEND Z.UTTPPR2 T - P Control Cards (2 Cards) BKEND	Tape-to-Printer Utility Sample Problem	$462	0101 - 0102
CATALS Z.AS5 BKEND Z.AS5 Assembler Source Deck (23 Cards) BKEND	Assembler Sample Problem 5 (Generate input for Tape and Disk Sort/Merge	$465	0502 - 0523
CATALS Z.SM5 BKEND Z.SM5 S/M Control Cards (6 Cards) BKEND	Tape and Disk Sort/Merge 2311 Sample Problem (Generate input for Disk-to-Printer Utility)	$483	0101 - 0106
CATALS Z.UTDKPR2 BKEND Z.UTDKPR2 D - P Control Cards (2 Cards) BKEND	Disk-to-Printer Utility Sample Problem	$461	0101 - 0102
CATALS Z.AS6 BKEND Z.AS6 Assembler Source Deck (23 Cards) BKEND	Assembler Sample Problem 6 (Generate input for Tape and Disk Sort/Merge)	$465	0602 - 0623
CATALS Z.SM6 BKEND Z.SM6 S/M Control Cards (6 Cards) BKEND	Tape and Disk Sort/Merge 2314 Sample Problem (Generate input for Disk-to-Printer Utility)	$483	0201 - 0206
CATALS Z.UTDKPR3 BKEND Z. UTDKPR3 D - P Control Cards (2 Cards) BKEND	Disk-to-Printer Utility Sample Problem	$461	0201 - 0202
CATALS Z.AT1 BKEND Z.AT1 Autotest Source Deck (27 Cards) Autotest Data Deck (5 Cards) BKEND	Autotest Sample Problem	$459	0001 - 0027

Figure 14. Sample Problem Card Decks Retrieved from IBM-Supplied Core Image and Source Statement Library Volumes (Part 2 of 3)

Cards Retrieved		Card Cols. 73-76	Card Cols. 77-80
CATALS Z.VFU1 BKEND Z.VFU1 VFU1 Source Deck (65 Cards) BKEND	Vocabulary File Utility Sample Problem	Refer to the Vocabulary File Utility Program publication listed on the front cover of this manual.	
CATALS Z.MCR1 BKEND Z.MCR1 MCR1 Source Deck (71 Cards) BKEND	Magnetic Character Recognition Devices (1412/1419)	S477	0001 - 0071
CATALS Z.MCR2 BKEND Z.MCR2 MCR2 Source Deck (115 Cards) BKEND		S477	0001 - 0115
CATALS Z.ORDC BKEND Z.ORDC Optical Reader Source Deck (179 Cards) BKEND	1287 Sample Problem Document Mode	S478	0001 - 9179
CATALS Z.ORJT BKEND Z.ORJT Source Deck (81 Cards) BKEND	1285/1287 Sample Problem for Journal Tape	S478	1001 - 1081
CATALS Z.EU3SPRGM BKEND Z.EU3SPRGM EU 1400 object decks and source decks BKEND	1401/1440/1460 Emulator Programs for IBM Model 2025/2030		
CATALS Z.EU4SPRGM BKEND Z.EU4SPRGM EU 1400 object decks and source decks BKEND	1401/1440/1460 Emulator Programs for IBM Model 2040		

Figure 14. Sample Problem Card Decks Retrieved from IBM-Supplied Core Image and Source Statement Library Volumes (Part 3 of 3)

The sample problems can be run as separate jobs; but, when possible should be run as successive job steps within an operating system environment. A PAUSE card placed at the end of each sample problem to be run in successive job steps allows the operator to make any necessary changes in device assignments. The order for running the sample problems is not completely arbitrary. The assembler, sort/merge, and utility programs should be run consecutively; the output of one program becomes the input to the next program. The FORTRAN, Basic FORTRAN COBOL, PL/I, VFU1, MICR, OCR, and RPG sample problems can either precede or follow the other programs. The sample problems for Autotest and multiprogramming should be run last when the programs are run as successive job steps.

Physical and Logical I/O Assignments

It is assumed that the user has made assignments for these logical I/O functions:

SYSLOG	SYSPCH	SYS001
SYSRDR	SYSLST	SYS002
SYSIPT	SYSLNK	SYS003

The preceding assignments are for the background problem program area only. The assignments necessary for the multiprogramming sample problem are included in the input test data for that problem.

Unique disk extents must be assigned initially to SYSLNK, SYS001, SYS002, and SYS003.

These extents can be assigned to the system residence pack as follows:

```
SYSLNK
   // XTENT 1,0,000166000,000170009,'111111',SYSLNK

SYS001
   // XTENT 128,0,000171000,000198002,'111111',SYS001

SYS002
   // XTENT 128,0,000171003,000198006,'111111',SYS002

SYS003
   // XTENT 128,0,000171007,000198009,'111111',SYS003
```

If standard assignments are missing, the necessary ASSGN cards should be inserted at the beginning of the job stream. After each job step of a sample problem, certain additional logical I/O assignments or reassignments may be required. Examples of such reassignments are shown in Figures 15 and 16. When reassignments are necessary, the user must furnish:

- The channel and unit number (X'cuu') for each tape or disk extent.
- The proper disk pack serial number and data cell volume number in all DLAB and XTENT cards.

Tape, Disk, and Data Cell Configurations

In addition to the minimum system configuration, three tape drives are required to run the Tape Sort/Merge sample problem, and five tape drives are required to run the Tape and Disk Sort/Merge sample problem (2400 application). A 2314 direct access storage facility is required to run the Tape and Disk Sort/Merge sample problem (2314 application). For installations with 7-track drives, convert feature OFF and translate ON (X'B8' as third operand of ASSGN) must be specified. Where 9-track tape drives are used, the third operand (X'B8') can be omitted. The logical I/O assignments for tape are shown in Figure 15 and those for disk and data cell are shown in Figure 16.

In addition to checking the proper function of each sample problem, the programs also test job control and linkage editor functions. They are helpful in verifying the correct generation of the user's operational system. The main purposes of the sample problems, however, are demonstration and instruction.

A listing of the source program and job control cards is written on SYSLST for each problem. If LOG is keyed into the 1052 Printer-Keyboard at the beginning of the job, a listing of all job control cards and operator messages is written on SYSLOG (Figures 17, 18, 19, and 20). Detailed setup procedures, including job control cards, are given for each sample problem.

Note that the SYSLOG output for the 1401/1440/1460 Emulator Programs can be found in the Emulator Program manual listed in the Preface.

Sample Problem	Disk Extent No. 1	Tape Drive No. 1[1]	Tape Drive No. 2[1]	Tape Drive No. 3[1]	Tape Drive No. 4	Tape Drive No. 5
Tape Sort/Merge Execute	SYSRES	SYS004	SYS001[2] SYS003[3]	SYS002[2] SYS005[3]		
Tape and Disk Sort/Merge (2400 Application)	SYSRES	SYS001	SYS002	SYS003	SYS004	SYS005
Tape-to-Printer Utility Execute	SYSRES	------	SYS004	------		

1. Tape drive number refers to X'cuu' operand in ASSGN card.

2. If a 7-track tape drive is used, the third operand (X'B8') is required to turn byte convert off.

3. For Tape Sort/Merge, SYS001 and SYS003 must be assigned to the same tape drive, as must SYS002 and SYS005.

Figure 15. Sequence of Logical I/O Assignments for Tape Sort/Merge and Tape-to-Printer Utility Sample Problems

Sample Problem	Disk Extent No. 1	Disk Extent No. 2	Disk Extent No. 3	Disk Extent No. 4
FORTRAN or Basic FORTRAN:				
Compile	SYSLNK	SYS001	SYS002**	------
Link Edit	SYSLNK	SYS001	------	------
Execute	-------	------	------	------
COBOL, RPG, Assembler:				
Compile	SYSLNK	SYS001	SYS002	SYS003
Link Edit	SYSLNK	SYS001	------	------
Execute	-------	------	------	------
AUTOTEST:				
Assemble	SYSLNK	SYS001	SYS002	SYS003
Link Edit	SYSLNK	SYS001	------	------
Execute	SYSLNK*	------	------	------
Disk Sort/Merge Execute	SYS002	SYS004	------	------
Tape and Disk Sort/Merge (2311 Application)	SYS001	SYS002	SYS003	------
Tape and Disk Sort/Merge (2314 Application)	SYS001	SYS002	SYS003	------
Disk-to-Printer Utility Execute	SYS004	SYS005	------	------
Data Cell-to-Printer Utility Execute	SYS004	------	------	------
1285, 1287, 1412/1419:				
Assemble	SYSLNK	SYS001	SYS002	SYS003
Link Edit	SYSLNK	SYS001	------	------
Execute	-------	------	------	------

* This is the Autotest work file (IJSYSAT).
** FORTRAN IV only.

Figure 16. Sequence of Logical I/O Assignments for Disk/Data-Cell Sample Problems

```
// JOB FORTRAN
EOJ FORTRAN
// JOB FORTRAN IV
EOJ FORTRAN4
// JOB COBOL
EOJ COBOL
// JOB RPG
EOJ RPG
// JOB PL/I
EOJ PL/I
// JOB ASSEMBLE
// PAUSE  ASSGN SYS002 TO A 9 TRACK TAPE DRIVE
EOJ ASSEMBLE
// JOB SORT
// PAUSE  ASSGN SYS002 TO SAME TAPE DRIVE AS IN PREVIOUS JOB
// PAUSE  ASSGN SYS001,SYS003,SYS004,SYS005 TO DIFF TAPES
EOJ SORT
// JOB TPPR
// PAUSE  ASSGN SYS004 TO SYS001 OF PREVIOUS JOB
// PAUSE  ASSGN SYS005 TO THE PRINTER
8001D IS IT EOF
EOJ TPPR
// JOB ASSEMBLE
// PAUSE  ASSGN SYS002 TO SCRATCH PACK
EOJ ASSEMBLE
// JOB DSORT
// PAUSE  ASSGN SYS002 AND SYS004 TO SCRATCH PACK
EOJ DSORT
// JOB DKPR
// PAUSE  ASSGN SYS004 TO SCRATCH PACK
// PAUSE  ASSGN SYS005 TO THE PRINTER
EOJ DKPR
// JOB ASSEMBLE
// PAUSE  ASSGN SYS004 TO DATA CELL
// PAUSE  MOUNT CELL nnnnnn ON STATION 3
EOJ ASSEMBLE
// JOB DCPR
// PAUSE  ASSGN SYS004 TO DATA CELL
// PAUSE  ASSGN SYS005 TO THE PRINTER
EOJ DCPR
BG // JOB ASSEMBLY SAMPLE 4
BG // PAUSE ASSGN SYS002 TO A 9 TRACK TAPE DRIVE
BG
BG EOJ ASSEMBLY
BG // JOB SORT 2400
BG // PAUSE ASSGN SYS002 TO SAME DRIVE AS PREVIOUS JOB
BG
BG // PAUSE ASSGN SYS001,SYS003,SYS004,SYS005 TO DIFF TAPES
BG
BG 7905I RCD IN 0002000, OUT 0002000, ESTIMATED 0000000
BG 7101I END SORT PH
BG 7905I RCD IN 0002000, OUT 0002000, ESTIMATED 0000000
BG 7201I END MERGE PH
```

Figure 17. SYSLOG Output for all Sample Problems Except Multiprogramming and Vocabulary File Utility Problems (Part 1 of 3)

```
BG 7905I RCD IN 0002000, OUT 0002000, ESTIMATED 0000000
BG 7302I EOJ
BG EOJ SORT
BG // JOB TPPR 2400 TO PRINTER
BG // PAUSE ASSGN SYS004 TO SYS001 OF PREVIOUS JOB
BG
BG // PAUSE ASSGN SYS005 TO THE PRINTER
BG
BG 8001D IS IT EOF
Y
BG EOJ TPPR
BG 1C00A ATTN.  0 0C.
BG
BG // JOB ASSEMBLY SAMPLE 5
BG // PAUSE ASSGN SYS002 TO SCRATCH PACK
BG
BG EOJ ASSEMBLY
BG // JOB SORT 2311
BG // PAUSE ASSGN SYS001,SYS002,SYS003 TO SCRATCH PACK
BG
BG 7905I RCD IN 0002000, OUT 0002000, ESTIMATED 0002000
BG 7101I END SORT PH
BG 7905I RCD IN 0002000, OUT 0002000, ESTIMATED 0002000
BG 7201I END MERGE PH
BG 7905I RCD IN 0002000, OUT 0002000, ESTIMATED 0002000
BG 7302I EOJ
BG EOJ SORT
BG // JOB DKPR 2311 TO PRINTER
BG // PAUSE ASSGN SYS004 TO SCRATCH PACK
BG
BG // PAUSE ASSGN SYS005 TO THE PRINTER
BG
BG 1C00A ATTN.  0 0C.
BG
BG // JOB ASSEMBLY SAMPLE 6
BG // PAUSE ASSGN SYS002 TO SCRATCH PACK
BG
BG EOJ ASSEMBLY
BG // JOB SORT 2314
BG // PAUSE ASSGN SYS001,SYS002,SYS003 TO SCRATCH PACK
BG
BG 7905I RCD IN 0003000, OUT 0003000, ESTIMATED 0003000
BG 7101I END SORT PH
BG 7905I RCD IN 0003000, OUT 0003000, ESTIMATED 0003000
BG 7201I END MERGE PH
BG 7905I RCD IN 0003000, OUT 0003000, ESTIMATED 0003000
BG 7302I EOJ
BG EOJ SORT
BG // JOB DKPR 2314 TO PRINTER
BG // PAUSE ASSGN SYS004 TO SCRATCH PACK
BG
BG // PAUSE ASSGN SYS005 TO THE PRINTER
BG
BG 1C00A ATTN.  0 0C.
BG
```

Figure 17. SYSLOG Output for All Sample Problems Except Multiprogramming and Vocabulary File Utility Problems (Part 2 of 3)

```
// JOB AUTOTEST
EOJ AUTOTEST
BG // JOB MCR SAMPLE PROBLEMS
BG // PAUSE   END OF MCR1 SAMPLE PROBLEM
BG
BG  INTERVENTION REQUIRED ON FILE2
BG EOJ MCR
BG // PAUSE   END OF MCR SAMPLE PROBLEMS
BG
// JOB OPTICAL READER SAMPLE PROBLEM
EOJ OPTICAL
// PAUSE END OF OPTICAL READER SAMPLE PROBLEM
Note:  If a multiprogramming system, the message will be preceded by
       the appropriate prefix (BG, F1, F2 or AR).
```

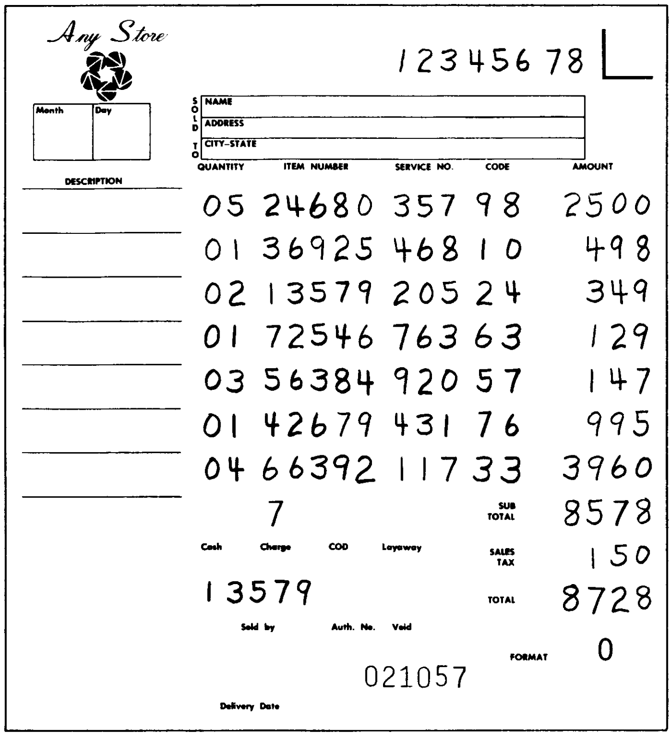
• Figure 17. SYSLOG Output for All Sample Problems except Multiprogramming and Vocabulary File Utility Problems (Part 3 of 3)

The SYSLST output reproduces much of the SYSLOG output. In addition, SYSLST displays the source programs, storage maps, sort/merge and utility program control cards, problem results, and other information. If SYSLST is a 1403 Printer equipped with the Universal Character Set (UCS) feature, see UCS command in the Disk Operating System Operating Guide. If this specification is not made, the issuance of a control command by job control causes a command reject, resulting in job cancellation.

```
BG // JOB CATALOG SAMPLE FOREGROUND/BACKGROUND PROGRAM
BG EOJ CATALOG

BG // JOB EXECUTE SAMPLE FOREGROUND/BACKGROUND PROGRAM
BG // PAUSE CARD READER END-OF-FILE SIGNALS END OF CARD INPUT
BG
BG 0P08A     INTERV REQ SYSRDR=00C
        CCSW=0210002240020000000 SNS=400000000000 CCB=002220
BG EOJ EXECUTE
BG // PAUSE REFER TO PRINTER FOR ADDITIONAL INSTRUCTIONS
BG assgn sysrdr,ua
BG assgn sysipt,ua
BG assgn syslst,ua
BG stop
AR 1I60A  READY FOR COMMUNICATIONS.
AR start f1
F1 assgn sys001,x'00c'
F1 assgn sys00[illegible],x'00e'
F1 exec cardprnt
F1 0S10I PROGRAM CARDPRNT COMPLETED
```

Figure 18. SYSLOG Output for Multiprogramming Sample Problem



Any Store

12345678

Month	Day

SOLD TO: NAME / ADDRESS / CITY-STATE

DESCRIPTION	QUANTITY	ITEM NUMBER	SERVICE NO.	CODE	AMOUNT
	05	24680	357	98	2500
	01	36925	468	10	498
	02	13579	205	24	349
	01	72546	763	63	129
	03	56384	920	57	147
	01	42679	431	76	995
	04	66392	117	33	3960
	7			SUB TOTAL	8578

Cash Charge COD Layaway

SALES TAX 150

13579

TOTAL 8728

Sold by Auth. No. Void

FORMAT 0

021057

Delivery Date

Figure 19. Optical Reader Sample Input Document for Document Mode Processing

```
012 3456 789C
123 4567 890S
234 5678 901T
345 6789 012N
456 7890 123S
567 8901 234X
678 9012 345C
789 0123 456T
890 1234 567Z
901 2345 678/
012 3456 789C
123 4567 890S
234 5678 901T
345 6789 012N
456 7890 123S
567 8901 234X
678 9012 345C
789 0123 456T
890 1234 567Z
901 2345 678/
012 3456 789C
123 4567 890S
234 5678 901T
345 6789 012N
456 7890 123S
567 8901 234X
678 9012 345C
789 0123 456T
890 1234 567Z
901 2345 678/
```

Figure 20. Optical Reader Sample Input for Journal Tape Mode Processing

Examples of Sample Problem Execution and Output

Basic FORTRAN

Program Name is Z.F01. The Basic FORTRAN sample problem generates all prime numbers between 0 and 1000. To execute the Basic FORTRAN sample problem the following job control cards are needed:

```
// JOB FORTRAN SAMPLE
// OPTION LINK,LIST,LOG
// EXEC FORTRAN
   Basic FORTRAN SOURCE DECK (22 cards)
/*
```

```
// EXEC LNKEDT
// EXEC
/&
```

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 21:

- Job control cards
- Source program listing
- Object program storage map
- Linkage editor storage map
- List of prime numbers generated by the program

```
// JOB FORTRAN SAMPLE
// OPTION LINK,LIST,LOG
// EXEC FORTRAN
```

```
                          DISK OPERATING SYSTEM/360 FORTRAN   360N-FO-451 30

      C     PRIME NUMBER PROBLEM                                    $4510001
        100 WRITE (3,8)                                             $4510002
          8 FORMAT (52H FOLLOWING IS A LIST OF PRIME NUMBERS FROM 1 TO 1000/  $4510003
           /19X,1H1/19X,1H2/19X,1H3)                                $4510004
        101 I=5                                                     $4510005
          3 A=I                                                     $4510006
        102 A=SQRT(A)                                               $4510007
        103 J=A                                                     $4510008
        104 DO 1 K=3,J,2                                            $4510009
        105 L=I/K                                                   $4510010
        106 IF(L*K-I)1,2,4                                          $4510011
          1 CONTINUE                                                $4510012
        107 WRITE (3,5)I                                            $4510013
          5 FORMAT (I20)                                            $4510014
          2 I=I+2                                                   $4510015
        108 IF(1000-I)7,4,3                                         $4510016
          4 WRITE (3,9)                                             $4510017
          9 FORMAT (14H PROGRAM ERROR)                              $4510018
          7 WRITE (3,6)                                             $4510019
          6 FORMAT (31H THIS IS THE END OF THE PROGRAM)             $4510020
        109 STOP                                                    $4510021
            END                                                     $4510022
```

```
      01/01/68          FORTMAIN                                        0002

                                        SCALARS

SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
I          006C       A          0070       J          0074       K          0078       L          007C

                                        CALLED SUBROUTINES

IJTAPST     IJTACOM     IJTSSQT     SQRT

LABEL    LOCATION     LABEL    LOCATION     LABEL    LOCATION     LABEL    LOCATION     LABEL    LOCATION
00100      0078       00008      0088       00101      00D8       00003      00E0       00102      0100
00103      010E       00104      012C       00105      0134       00106      0144       00001      015E
00107      0172       00005      0190       00002      019A       00108      01A6       00004      01BC
00009      01DC       00007      01E8       00006      01FC       00109      0226
      COMPILATION COMPLETE    AMOUNT OF COMMON 000000     AMOUNT OF CORE 000696    ADDRESS BASE TABLE    0200
```

```
// EXEC LNKEDT
```

Figure 21. SYSLST Output for Basic FORTRAN Sample Problems (Part 1 of 2)

```
JOB  FORTRAN              DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN    MAP
LIST   AUTOLINK   IJTACOM
LIST   AUTOLINK   IJTACON
LIST   AUTOLINK   IJTAPST
LIST   AUTOLINK   IJTFIOS
LIST   AUTOLINK   IJTSSQT
LIST   ENTRY
```

```
01/01/68   PHASE   XFR-AD  LOCORE  HICORE  DSK-AD   ESD TYPE  LABEL     LOADED  REL-FR

        PHASE***   002000  002000  C0429F  15 9 1   CSECT     FORTMAIN  002000  002000

                                                    CSECT     IJTAPST   003510  003510

                                                    CSECT     IJTACOM   0022B8  0022B8
                                                       ENTRY  IJTSAVE   002820

                                                    CSECT     IJTSSQT   0041F8  0041F8
                                                       ENTRY  SQRT      0041FE

                                                    CSECT     IJTACON   002B80  002B80
                                                    *  ENTRY  FCVFI     002B80
                                                    *  ENTRY  FCVFO     002B84
                                                    *  ENTRY  FCVEI     002B88
                                                    *  ENTRY  FCVEO     002B8C
                                                    *  ENTRY  FCVII     002B90
                                                    *  ENTRY  FCVIO     002B94
                                                    *  ENTRY  FCVDI     002D3E
                                                    *  ENTRY  FCVDO     002F46

                                                    CSECT     IJTFIOS   0035A0  0035A0
                                                       ENTRY  UNITABE   003F86
                                                    *  ENTRY  DOIOXXE   003DBA
                                                    *  ENTRY  GETUNTE   00396A
                                                    *  ENTRY  OPENUNE   0039B8
                                                    *  ENTRY  SETLGUE   003C26
                                                    *  ENTRY  CCWNO1E   003F58
                                                    *  ENTRY  DSKWTME   003E60
                                                    *  ENTRY  ASNBUFE   003FF4
                                                    *  ENTRY  FILTABE   003EE8
                                                    *  ENTRY  BUFINNT   004142
                                                    *  ENTRY  NRFSWE    003DB8
                                                    *  ENTRY  CLOSUNE   003B56
```

```
// EXEC

FOLLOWING IS A LIST OF PRIME NUMBERS FROM 1 TO 1000
                 1
                 2
                 3
                 5
                 7
                11
```

```
               977
               983
               991
               997
THIS IS THE END OF THE PROGRAM
```

```
EOJ FORTRAN
```

Figure 21. SYSLST Output for Basic FORTRAN Sample Problems (Part 2 of 2)

FORTRAN

Program Name is Z.ILFSAMPL. This sample problem generates all prime numbers between 2 and 1000. To execute it, the following job control statements are needed:

```
// JOB FORTRAN4 SAMPLE
// OPTION LINK,LIST,LOG
// EXEC FFORTRAN
```

```
FORTRAN IV SOURCE DECK (22 cards)
/*
// EXEC LNKEDT
// EXEC
/*
```

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 22.

- Job control cards
- Source program listing
- Object program storage map
- Linkage editor storage map
- List of prime numbers generated by the program

```
// JOB FORTRAN4  SAMPLE                                                00.03.48
// OPTION LINK,LIST,LOG
// EXEC FFORTRAN

     MAP     YES

     LOAD    YES

     DECK    NO

     LIST    YES

     LISTX   NO

     EBCDIC
```

```
 DOS FORTRAN IV 360N-FO-479 3-0          MAINPGM          DATE   08/22/68      TIME     00.00.12       PAGE 0001

                C PRIME NUMBER GENERATOR
  0001                WRITE (3,1)
  0002              1 FORMAT ('1FOLLOWING IS A LIST OF PRIME NUMBERS FROM 2 TO 1000'/
                     *19X,'2'/19X,'3')
  0003                DO 4 I=5,1000,2
  0004                K = SQRT(FLOAT(I))
  0005                DO 2 J=3,K,2
  0006                IF (MOD(I,J) .EQ. 0) GO TO 4
  0007              2 CONTINUE
  0008                WRITE (3,3) I
  0009              3 FORMAT (I20)
  0010              4 CONTINUE
  0011                WRITE (3,5)
  0012              5 FORMAT (' THIS IS THE END OF THE PROGRAM')
  0013                STOP
  0014                END
```

```
 DOS FORTRAN IV 360N-FO-479 3-0          MAINPGM          DATE   08/22/68      TIME     00.00.12       PAGE 0002

                                     SCALAR MAP
SYMBOL       LOCATION         SYMBOL    LOCATION         SYMBOL      LOCATION        SYMBOL     LOCATION         SYMBOL     LOCATION
I                  A0         K              A4          J                 A8

                                  SUBPROGRAMS CALLED
SYMBOL       LOCATION         SYMBOL    LOCATION         SYMBOL      LOCATION        SYMBOL     LOCATION         SYMBOL     LOCATION
IBCOM=             AC         SQRT           B0

                                  FORMAT STATEMENT MAP
SYMBOL       LOCATION         SYMBOL    LOCATION         SYMBOL      LOCATION        SYMBOL     LOCATION         SYMBOL     LOCATION
     1             C4              3       108                5         10C

    TOTAL MEMORY REQUIREMENTS 000282 BYTES
```

```
    00.00.33,TOTAL COMPILATION TIME,00.00.21
```

Figure 22. SYSLST Output for FORTRAN 4 Sample Problems (Part 1 of 3)

```
// EXEC LNKEDT
```

```
JOB  FORTRAN4  08/22/68   DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN     MAP
LIST    AUTOLINK    ILFIBCOM
LIST    AUTOLINK    ILFADCON
LIST    AUTOLINK    ILFFINT
LIST    AUTOLINK    ILFFIOCS
LIST    AUTOLINK    IJJCPD1N
LIST    AUTOLINK    ILFDIOCS
LIST    AUTOLINK    ILESSORT
LIST    AUTOLINK    ILFUNTAB
LIST    ENTRY
```

```
08/22/68   PHASE    XFR-AD  LOCORE  HICORE  DSK-AD   ESD TYPE  LABEL     LOADED  REL-FR

        PHASE***   002800  002800  005FEF  3B 05 2  CSECT     MAINPGM   002800  002800

                                                    CSECT     ILFIBCOM  002A88  002A88
                                                       ENTRY  IBCOM=    002A88
                                                    *  ENTRY  OPSYS     00372C
                                                       ENTRY  INTSW     003881
                                                       ENTRY  PDPAR     0036A0
                                                       ENTRY  DUMPSW=   003636
                                                    *  ENTRY  IJTINTSW  003881

                                                    CSECT     ILESSORT  005F40  005F40
                                                       ENTRY  SORT      005F40

                                                    CSECT     ILFFINT   004AA0  004AA0
                                                       ENTRY  SAVERP    004F80

                                                    CSECT     ILFADCON  0039D0  0039D0
                                                       ENTRY  ILFFCVEO  0044AF
                                                       ENTRY  ILFFCVLO  003C62
                                                       ENTRY  ILFFCVIO  003FAC
                                                       ENTRY  ILFFCVCO  0046C8
                                                       ENTRY  ILFFCVAO  003BD2
                                                       ENTRY  ILFFCVZO  003B2C
                                                       ENTRY  INT6SW    004A88

                                                    CSECT     ILFFIOCS  005070  005070
                                                       ENTRY  ILFFBORG  0058C4
                                                       ENTRY  ILFBEORG  0058C0
                                                       ENTRY  IJSYSLO   0059A0
                                                       ENTRY  ADIOCR=   005828

                                                    CSECT     ILFUNTAB  005FF0  005FF0

                                                    CSECT     ILFDIOCS  005C08  005C08
                                                       ENTRY  ILFDIOCR  005C92
                                                    *  ENTRY  DIOCS=    005C08

                                                    CSECT     IJJCPD1N  005A78  005A78
                                                    *  ENTRY  IJJCPD3   005A78

                                                    CSECT     IJ2L0005  005A08  005070
```

```
// EXEC
```

Figure 22. SYSLST Output for FORTRAN 4 Sample Problems (Part 2 of 3)

```
FOLLOWING IS A LIST OF PRIME NUMBERS FROM 2 TO 1000
                    2
                    3
                    5
                    7
                   11
                   13
                   17
                   19
                   23
                   29
                   31
                   37
                   41
                   43
                   47
```

```
                  313
                  317
                  331
                  337
                  347
                  349
                  353
                  359
                  367
                  373
                  379
                  383
                  389
                  397
                  401
                  409
                  419
                  421
                  431
                  433
                  439
                  443
                  449
                  457
                  461
                  463
                  467
                  479
                  487
                  491
                  499
                  503
                  509
                  521
```

```
                  859
                  863
                  877
                  881
                  883
                  887
                  907
                  911
                  919
                  929
                  937
                  941
                  947
                  953
                  967
                  971
                  977
                  983
                  991
                  997
 THIS IS THE END OF THE PROGRAM
```

```
 EOJ FORTRAN4                                   00.02.17,DURATION 00.02.11
```

Figure 22. SYSLST Output for FORTRAN 4 Sample Problems (Part 3 of 3)

COBOL

Program Name is Z.CB1. The COBOL sample problem generates and prints a table of weekly, monthly, and annual salaries based on ten dollar increments in monthly salary from $500 to $1000. To execute the COBOL sample problem the following job control cards are needed:

```
// JOB COBOL SAMPLE
// OPTION LINK,LIST,LOG,LISTX,DUMP,SYM,ERRS
// EXEC COBOL
   COBOL SOURCE DECK (81 cards)
/*
// EXEC LNKEDT
// ASSGN SYS004,X'cuu' (SYS004 must be assigned to printer designated as
   SYSLST during system generation)
// EXEC
/&
```

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 23:

- Job control cards
- Source program listing
- Data division storage map
- Procedure division storage map
- Diagnostics
- Linkage editor storage map
- Table of salaries generated by the program

Note: Output is on SYS004 (same as printer).

```
// JOB COBOL SAMPLE
// OPTION LINK,LIST,LOG,LISTX,DUMP,SYM,ERRS
// EXEC COBOL
```

```
LINE NO. SEQ. NO.          SOURCE STATEMENT                                       CBD CL3-0

     1   001010 IDENTIFICATION DIVISION.                                                      $4520001
     2   001020 PROGRAM-ID. 'C360SAMP'.                                                       $4520002
     3   001030 REMARKS.                                                                      $4520003
     4   001040      EXAMPLE OF A 360 COBOL PROGRAM TO                                        $4520004
     5   001050      COMPUTE SALARIES.                                                        $4520005
     6   001060 ENVIRONMENT DIVISION.                                                         $4520006
     7   001070 CONFIGURATION SECTION.                                                        $4520007
     8   001080   SOURCE-COMPUTER. IBM-360.                                                   $4520008
     9   001090   OBJECT-COMPUTER. IBM-360.                                                   $4520009
    10   001100 INPUT-OUTPUT SECTION.                                                         $4520010
    11   001110 FILE-CONTROL.                                                                 $4520011
    12   001120      SELECT SALARY-FILE                                                       $4520012
    13   001130      ASSIGN TO 'SYS004' UNIT-RECORD 1403.                                     $4520013
    14   001140                                                                               $4520014
    15   001150 DATA DIVISION.                                                                $4520015
    16   001160 FILE SECTION.                                                                 $4520016
    17   001170  FD SALARY-FILE                                                               $4520017
    18   001180      LABEL RECORDS ARE OMITTED                                                $4520018
    19   001185      RECORDING MODE IS F                                                      $4520019
    20   001190      DATA RECORD IS SALARY-RECORD.                                            $4520020
    21   001200  01 SALARY-RECORD                                                             $4520021
    22   001210      PICTURE X(100).                                                          $4520022
    23   001220 WORKING-STORAGE SECTION.                                                      $4520023
    24   001230  77 TOTAL-A       PICTURE 9(6)V99     VALUE ZERO.                             $4520024
    25   001240  77 TOTAL-B       PICTURE 9(6)V99     VALUE ZERO.                             $4520025
    26   001250  77 TOTAL-C       PICTURE 9(6)V99     VALUE ZERO.                             $4520026
    27   002010  77 WEEKLY-PAY    PICTURE 999V99.                                             $4520027
    28   002020  77 MONTHLY-PAY   PICTURE 9999V99.                                            $4520028
    29   002030  77 ANNUAL-PAY    PICTURE 99999V99.                                           $4520029
    30   002040  77 CON-A  PICTURE 9(6)V99     VALUE IS 008826.69.                            $4520030
    31   002050  77 CON-B  PICTURE 9(6)V99     VALUE IS 038250.00 .                           $4520031
    32   002060  77 CON-C  PICTURE 9(6)V99     VALUE IS 459000.00 .                           $4520032
    33   002070  01 SALARIES.                                                                 $4520033
    34   002080      02 FILLER     PICTURE A(46)  VALUE SPACE.                                $4520034
    35   002090      02 WEEKLY     PICTURE ZZZ.99 .                                           $4520035
    36   002100      02 FILLER     PICTURE AAA    VALUE SPACE.                                $4520036
    37   002110      02 MONTHLY    PICTURE ZZZZ.99 .                                          $4520037
    38   002120      02 FILLER     PICTURE AAA    VALUE SPACE.                                $4520038
    39   002130      02 ANNUAL     PICTURE ZZZZZ.99 .                                         $4520039
    40   002140      02 FILLER     PICTURE A(27)  VALUE SPACE.                                $4520040
    41   002150  01 MESG.                                                                     $4520041
    42   002160      02 FILLER     PICTURE A(40)  VALUE SPACES.                               $4520042
    43   002170      02 SHOW       PICTURE A(26).                                             $4520043
    44   002180  01 DSPY.                                                                     $4520044
    45   002190      02 FILLER     PICTURE A(40)  VALUE SPACES.                               $4520045
    46   002200      02 PRSNT      PICTURE A(33).                                             $4520046
    47   002210  01 HEADING.                                                                  $4520047
    48   002220      02 FILLER     PICTURE A(46)  VALUE SPACES.                               $4520048
    49   002230      02 WEEKLY     PICTURE A(6)   VALUE IS 'WEEKLY'.                          $4520049
    50   002240      02 FILLER     PICTURE A(3)   VALUE IS SPACES.                            $4520050
    51   002250      02 MONTHLY    PICTURE A(7)   VALUE IS 'MONTHLY'.                         $4520051
    52   003010      02 FILLER     PICTURE A(3)   VALUE IS SPACES.                            $4520052
    53   003020      02 ANNUAL     PICTURE A(6)   VALUE IS 'ANNUAL'.                          $4520053
    54   003030      02 FILLER     PICTURE A(29)  VALUE IS SPACES.                            $4520054
```

Figure 23. SYSLST Output for COBOL Sample Problems (Part 1 of 5)

```
LINE NO. SEQ. NO.          SOURCE STATEMENT

  55    003040                                                                       S4520055
  56    003050 PROCEDURE DIVISION.                                                   S4520056
  57    003060 START.                                                                S4520057
  58    003070     OPEN OUTPUT SALARY-FILE.                                          S4520058
  59    003080     WRITE SALARY-RECORD FROM HEADING AFTER ADVANCING 0 LINES.         S4520059
  60    003090     PERFORM CALCULATIONS                                              S4520060
  61    003100       VARYING MONTHLY-PAY FROM 500 BY 10                              S4520061
  62    003110       UNTIL MONTHLY-PAY IS GREATER THAN 1000.                         S4520062
  63    003120       IF TOTAL-A = CON-A AND TOTAL-B = CON-B AND TOTAL-C = CON-C      S4520063
  64    003130       MOVE 'TABLE VALUES ARE CORRECT' TO SHOW                         S4520064
  65    003140       WRITE SALARY-RECORD FROM MESG AFTER ADVANCING 2 LINES           S4520065
  66    003150       ELSE                                                            S4520066
  67    003160       MOVE 'TABLE VALUES ARE NOT CORRECT' TO PRSNT                    S4520067
  68    003170       WRITE SALARY-RECORD FROM DSPY AFTER ADVANCING 2 LINES.          S4520068
  69    003180     CLOSE SALARY-FILE.                                                S4520069
  70    003190     STOP RUN.                                                         S4520070
  71    003200                                                                       S4520071
  72    003210 CALCULATIONS.                                                         S4520072
  73    003220     COMPUTE WEEKLY-PAY = 3 * MONTHLY-PAY / 13                         S4520073
  74    003230     COMPUTE ANNUAL-PAY = 12 * MONTHLY-PAY                             S4520074
  75    003240     MOVE WEEKLY-PAY TO WEEKLY IN SALARIES                             S4520075
  76    003250     MOVE MONTHLY-PAY TO MONTHLY IN SALARIES                           S4520076
  77    004010     MOVE ANNUAL-PAY TO ANNUAL IN SALARIES                             S4520077
  78    004020     ADD WEEKLY-PAY TO TOTAL-A                                         S4520078
  79    004030     ADD MONTHLY-PAY TO TOTAL-B                                        S4520079
  80    004040     ADD ANNUAL-PAY TO TOTAL-C                                         S4520080
  81    004050     WRITE SALARY-RECORD FROM SALARIES  AFTER ADVANCING 1 LINES.       S4520081


                              DATA DIVISION MAP

          TYPE      LOCATION    DATA NAME

          FILE                  SALARY-FILE
          REC       0000000     SALARY-RECORD
                    0000000     TOTAL-A
                    0000008     TOTAL-B
                    0000010     TOTAL-C
                    0000018     WEEKLY-PAY
                    000001D     MONTHLY-PAY
                    0000025     ANNUAL-PAY
                    000002A     CON-A
                    0000032     CON-B
                    000003A     CON-C
          REC       0000048     SALARIES
                    0000076     WEEKLY
                    000007F     MONTHLY
                    0000089     ANNUAL
          REC       00000B0     MESG
                    00000D8     SHOW
          REC       00000F8     DSPY
                    0000120     PRSNT
          REC       0000148     HEADING
                    0000176     WEEKLY
                    000017F     MONTHLY
                    0000189     ANNUAL
```

Figure 23. SYSLST Output for COBOL Sample Problems (Part 2 of 5)

```
                                            PROCEDURE DIVISION MAP

LINE/POS  ADDR        INSTRUCTION         LINE/POS  ADDR        INSTRUCTION         LINE/POS  ADDR        INSTRUCTION

00057 01 0004D8   ENTRY               00057 01 0004FE   58 00 3 190         00057 01 000508   0A 10
00058 01 0005CA   D7 B6 3 010 3 010   00058 01 000510   41 00 A 1C8         00058 01 000514   50 00 3 010
00058 01 000518   D7 03 A 1C8 A 1C8   00058 01 00051E   41 00 A 200         00058 01 000522   50 00 A 1CC
00058 01 000526   D2 06 A 1D0 A 328   00058 01 00052C   92 0A 3 014         00058 01 000530   41 10 A 320
00058 01 000534   41 00 3 010         00058 01 000538   0A 02               00058 01 00053A   41 10 A 310
00058 01 00053E   41 00 3 010         00058 01 000542   0A 02               00058 01 000544   58 90 A 1E0
00059 01 000548   D2 63 9 000 A 148   00059 01 00054E   92 F1 9 000         00059 01 000552   D2 02 3 011 A 1F1
00059 01 000558   D2 02 3 015 A 1E1   00059 01 00055E   58 10 3 010         00059 01 000566   50 10 3 018
00059 01 00056A   D2 02 A 1F1 3 019   00059 01 000570   58 10 3 014         00059 01 000578   50 10 3 018
00059 01 00057C   D2 02 A 1E1 3 019   00059 01 000582   48 10 A 1FC         00059 01 000586   06 10
00059 01 000588   40 10 A 1F6         00059 01 00058C   58 00 3 194         00059 01 000590   50 00 A 1D8
00059 01 000594   96 20 A 1DD         00059 01 000598   92 01 A 1F0         00059 01 00059C   41 10 A 1C8
00059 01 0005A8   41 00 3 286         00059 01 0005B0   47 10 3 286         00059 01 0005B4   0A 07
00059 01 0005B6   D2 02 A 1E1 3 011   00059 01 0005BC   D2 02 A 1F1 3 015   00059 01 0005C2   58 90 A 1E0
00060 01 0005C6   D2 01 3 1A6 3 636   00060 01 0005CC   D2 01 3 636 3 1A4   00060 01 0005D2   92 F0 A 01D
00060 01 0005D6   D2 04 A 01E A 01D   00060 01 0005DC   F3 31 A 01D A 2D8   00060 01 0005E2   D3 00 A 022 A 020
00060 01 0005E8   96 F0 A 020         00060 01 0005EC   47 F0 3 2EC         00062 01 0005F0   F2 35 3 038 A 01D
00062 01 0005F6   D7 03 3 010 3 010   00062 01 0005FC   D3 02 3 010 A 2D0   00062 01 000602   D1 01 3 010 A 2D0
00062 01 000608   D1 00 3 013 A 2D2   00062 01 00060E   F9 33 3 038 3 010   00062 01 000614   47 20 3 326
00060 01 000618   47 F0 3 2F0         00060 01 00061C   47 F0 3 2C0         00060 01 000620   47 F0 3 4C0
00060 01 000624   F2 F5 3 168 A 01D   00060 01 00062A   D7 0F 3 000 3 000   00060 01 000630   D3 01 3 00D A 2DA
00060 01 000636   D1 00 3 00D A 2DA   00060 01 00063C   D1 00 3 00F A 2DB   00060 01 000642   FA FF 3 168 3 000
00060 01 000648   F3 5F A 01D 3 168   00060 01 00064E   96 F0 A 022         00060 01 000652   47 F0 3 2C0
00060 01 000656   D2 01 3 636 3 1A6   00063 01 00065C   F2 47 3 03E A 000   00063 01 000662   F2 47 3 054 A 02A
00063 01 000668   F9 44 3 03E 3 054   00063 01 00066E   47 70 3 40A         00063 01 000672   F2 47 3 03E A 008
00063 01 000678   F2 47 3 054 A 032   00063 01 00067E   F9 44 3 03E 3 054   00063 01 000684   47 70 3 40A
00063 01 000688   F2 47 3 03E A 010   00063 01 00068E   F2 47 3 054 A 03A   00063 01 000694   F9 44 3 03E 3 054
00063 01 00069A   47 70 3 4CA         00064 01 00069E   D2 17 A 0D8 A 2DC   00064 01 0006A4   92 40 A 0F0
00064 01 0006A8   D2 00 A 0F1 A 0F0   00065 01 0006AE   D2 41 9 000 A 0B0   00065 01 0006B4   92 40 9 042
00065 01 0006B8   D2 20 9 043 9 042   00065 01 0006BE   92 F0 9 000         00065 01 0006C2   D2 02 3 011 A 1F1
00065 01 0006C8   D2 02 3 015 A 1E1   00065 01 0006CE   58 10 3 010         00065 01 0006D6   50 10 3 018
00065 01 0006DA   D2 02 A 1F1 3 019   00065 01 0006E0   58 10 3 014         00065 01 0006E8   50 10 3 018
00065 01 0006EC   D2 02 A 1E1 3 019   00065 01 0006F2   48 10 A 1FC         00065 01 0006F6   06 10
00065 01 0006F8   40 10 A 1F6         00065 01 0006FC   58 00 3 198         00065 01 000700   50 00 A 1D8
00065 01 000704   96 20 A 1DD         00065 01 000708   92 01 A 1F0         00065 01 00070C   41 10 A 1C8
00065 01 000718   41 00 3 3F6         00065 01 000720   47 10 3 3F6         00065 01 000724   0A 07
00065 01 000726   D2 02 A 1E1 3 011   00065 01 00072C   D2 02 A 1F1 3 015   00065 01 000732   58 90 A 1E0
00063 01 000736   47 F0 3 4A2         00067 01 00073A   D2 1B A 120 A 2F4   00067 01 000740   92 40 A 13C
00067 01 000744   D2 03 A 13D A 13C   00068 01 00074A   D2 48 9 000 A 0F8   00068 01 000750   92 40 9 049
00068 01 000754   D2 19 9 04A 9 049   00068 01 00075A   92 F0 9 000         00068 01 00075E   D2 02 3 011 A 1F1
00068 01 000764   D2 02 3 015 A 1E1   00068 01 00076A   58 10 3 010         00068 01 000772   50 10 3 018
00068 01 000776   D2 02 A 1F1 3 019   00068 01 00077C   58 10 3 014         00068 01 000784   50 10 3 018
00068 01 000788   D2 02 A 1E1 3 019   00068 01 00078E   48 10 A 1FC         00068 01 000792   06 10
00068 01 000794   40 10 A 1F6         00068 01 000798   58 00 3 19C         00068 01 00079C   50 00 A 1D8
00068 01 0007A0   96 20 A 1DD         00068 01 0007A4   92 01 A 1F0         00068 01 0007A8   41 10 A 1C8
00068 01 0007B4   41 00 3 492         00068 01 0007BC   47 10 3 492         00068 01 0007C0   0A 07
00068 01 0007C2   D2 02 A 1E1 3 011   00068 01 0007C8   D2 02 A 1F1 3 015   00068 01 0007CE   58 90 A 1E0
00069 01 0007D2   D7 B6 3 010 3 010   00069 01 0007D8   41 00 A 1C8         00069 01 0007DC   50 00 3 010
00069 01 0007E0   92 0A 3 014         00069 01 0007E4   41 10 A 318         00069 01 0007E8   41 00 3 010
00069 01 0007EC   0A 02               00070 01 0007EE   0A 0E               00073 01 0007F0   F2 F5 3 178 A 01D
00073 01 0007F6   FC F0 3 178 A 2D3   00073 01 0007FC   D7 06 3 168 3 168   00073 01 000802   F2 00 3 16D 3 187
```

```
                                            PROCEDURE DIVISION MAP

LINE/POS  ADDR        INSTRUCTION         LINE/POS  ADDR        INSTRUCTION         LINE/POS  ADDR        INSTRUCTION

00073 01 000808   F1 5E 3 168 3 178   00073 01 00080E   94 0F 3 168         00073 01 000812   D1 00 3 16E 3 187
00073 01 000818   FD 61 3 168 A 2D4   00073 01 00081E   F3 54 3 020 3 168   00073 01 000824   D3 00 3 024 3 025
00073 01 00082A   D2 04 A 018 3 02C   00073 01 000830   96 F0 A 01C         00074 01 000834   F2 F5 3 168 A 01D
00074 01 00083A   FC F1 3 168 A 2D6   00074 01 000840   F3 6F A 023 3 168   00074 01 000846   96 F0 A 029
00075 01 00084A   F2 24 3 010 A 018   00075 01 000850   D2 06 3 038 A 1AC   00075 01 000856   DE 06 3 038 3 010
00075 01 00085C   D2 05 A 076 3 039   00076 01 000862   F2 35 3 010 A 01D   00076 01 000868   D2 08 3 038 A 1B3
00076 01 00086E   DE 08 3 038 3 010   00076 01 000874   D2 06 A 07F 3 03A   00077 01 00087A   F2 36 3 010 A 023
00077 01 000880   D2 08 3 038 A 1B3   00077 01 000886   DE 08 3 038 3 010   00077 01 00088C   D2 07 A 089 3 039
00078 01 000892   F2 F7 3 168 A 000   00078 01 000898   F2 F4 3 000 A 018   00078 01 00089E   FA FF 3 168 3 000
00078 01 0008A4   F3 7F A 000 3 168   00078 01 0008AA   96 F0 A 007         00079 01 0008AE   F2 F7 3 168 A 008
00079 01 0008B4   F2 F5 3 000 A 01D   00079 01 0008BA   FA FF 3 168 3 000   00079 01 0008C0   F3 7F A 008 3 168
00079 01 0008C6   96 F0 A 00F         00080 01 0008CA   F2 F7 3 168 A 010   00080 01 0008D0   F2 F6 3 000 A 023
00080 01 0008D6   FA FF 3 168 3 000   00080 01 0008DC   F3 7F A 010 3 168   00080 01 0008E2   96 F0 A 017
00081 01 0008E6   D2 63 9 000 A 048   00081 01 0008EC   92 40 9 000         00081 01 0008F0   D2 02 3 011 A 1F1
00081 01 0008F6   D2 02 3 015 A 1E1   00081 01 0008FC   58 10 3 010         00081 01 000904   50 10 3 018
00081 01 000908   D2 02 A 1F1 3 019   00081 01 00090E   58 10 3 014         00081 01 000916   50 10 3 018
00081 01 00091A   D2 02 A 1E1 3 019   00081 01 000920   48 10 A 1FC         00081 01 000924   06 10
00081 01 000926   40 10 A 1F6         00081 01 00092A   58 00 3 1A0         00081 01 00092E   50 00 A 1D8
00081 01 000932   96 20 A 1DD         00081 01 000936   92 01 A 1F0         00081 01 00093A   41 10 A 1C8
00081 01 000946   41 00 3 624         00081 01 00094E   47 10 3 624         00081 01 000952   0A 07
00081 01 000954   D2 02 A 1E1 3 011   00081 01 00095A   D2 02 A 1F1 3 015   00081 01 000960   58 90 A 1E0
00000 00 000964   47 F0 3 638
```

Figure 23. SYSLST Output for COBOL Sample Problems (Part 3 of 5)

```
                                   DIAGNOSTICS

LINE/POS ER CODE   CLAUSE                                   MESSAGE

  21- 1  IJS063I W ALIGNMENT      TO ALIGN BLOCKED RECORDS ADD 4 BYTES TO THE 01 CONTAINING DATANAME SALARY-RECORD.
  33- 1  IJS053I W ALIGNMENT      FOR PROPER ALIGNMENT, A 6 BYTE LONG FILLER ENTRY IS INSERTED PRECEDING SALARIES.
  41- 1  IJS053I W ALIGNMENT      FOR PROPER ALIGNMENT, A 4 BYTE LONG FILLER ENTRY IS INSERTED PRECEDING MESG.
  44- 1  IJS053I W ALIGNMENT      FOR PROPER ALIGNMENT, A 6 BYTE LONG FILLER ENTRY IS INSERTED PRECEDING DSPY.
  47- 1  IJS053I W ALIGNMENT      FOR PROPER ALIGNMENT, A 7 BYTE LONG FILLER ENTRY IS INSERTED PRECEDING HEADING.
```

```
                                   DIAGNOSTICS

     5 LEVEL W DIAGNOSTICS
```

```
     // EXEC LNKEDT
```

```
JOB  COBOL     01/01/68   DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN    MAP
LIST   INCLUDE IHD03300                                               C3600001
LIST   AUTOLINK    IJDFAPIZ
LIST   ENTRY
```

```
01/01/68   PHASE   XFR-AD  LOCORE  HICORE  DSK-AD   ESD TYPE  LABEL     LOADED  REL-FR

        PHASE***  0025C0  002000  00285B  1D 2 1   CSECT     IHD03300  002000  002000
                                                   * ENTRY   IHD03301  002054
                                                     ENTRY   IHD03302  002000

                                                   CSECT     C360SAMP  0020E8  0020E8

                                                   CSECT     IJDFAPIZ  002A50  002A50
                                                   * ENTRY   IJDFAZIZ  002A50
```

```
// ASSGN SYS004,X'00E'
// EXEC
```

Figure 23. SYSLST Output for COBOL Sample Problems (Part 4 of 5)

WEEKLY	MONTHLY	ANNUAL
115.38	500.00	6000.00
117.69	510.00	6120.00
120.00	520.00	6240.00
122.30	530.00	6360.00
124.61	540.00	6480.00
126.92	550.00	6600.00
129.23	560.00	6720.00
131.53	570.00	6840.00
133.84	580.00	6960.00
136.15	590.00	7080.00
138.46	600.00	7200.00
140.76	610.00	7320.00
143.07	620.00	7440.00
145.38	630.00	7560.00
147.69	640.00	7680.00
150.00	650.00	7800.00
152.30	660.00	7920.00
154.61	670.00	8040.00
156.92	680.00	8160.00
159.23	690.00	8280.00
161.53	700.00	8400.00
163.84	710.00	8520.00
166.15	720.00	8640.00
168.46	730.00	8760.00
170.76	740.00	8880.00
173.07	750.00	9000.00
175.38	760.00	9120.00
177.69	770.00	9240.00
180.00	780.00	9360.00
182.30	790.00	9480.00
184.61	800.00	9600.00
186.92	810.00	9720.00
189.23	820.00	9840.00
191.53	830.00	9960.00
193.84	840.00	10080.00
196.15	850.00	10200.00
198.46	860.00	10320.00
200.76	870.00	10440.00
203.07	880.00	10560.00
205.38	890.00	10680.00
207.69	900.00	10800.00
210.00	910.00	10920.00
212.30	920.00	11040.00
214.61	930.00	11160.00
216.92	940.00	11280.00
219.23	950.00	11400.00
221.53	960.00	11520.00
223.84	970.00	11640.00
226.15	980.00	11760.00
228.46	990.00	11880.00
230.76	1000.00	12000.00

TABLE VALUES ARE CORRECT

EOJ COBOL

Figure 23. SYSLST Output for COBOL Sample Problems (Part 5 of 5)

RPG

Program Name is Z.RG1. The RPG sample problem produces an accounts receivable register. Both the RPG source deck and data deck are retrieved from the maintenance volume under Retrieving the Sample Problems. To execute the RPG sample problem the following job control cards are needed:

```
// JOB RGP SAMPLE
// OPTION LINK,DUMP
// EXEC RPG
   RPG SOURCE DECK (45 cards)
/*
// EXEC LNKEDT
// EXEC
   RPG DATA (13 cards)
/*
/&
```

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 24.

- Job control cards
- Source program listing
- Symbol tables
- Memory map
- Linkage editor storage map
- Accounts receivable register generated by program

```
// JOB RPG SAMPLE
// OPTION LINK,DUMP
// EXEC RPG
```

```
DOS/360*RPG*CL 3-0                  RPG

       01 000 H                                                            $460
001    01 010 FINPUT   IPE F   80  80                 READ40 SYSIPT        $460
002    01 020 FOUTPUT  O   F  132 132      OF         PRINTERSYSLST        $460
003    01 010 IINPUT   AA  01   1 Z                                        $460
004    01 020 I                                          8  29 NAME        $460
005    01 030 I                                         30  31MONTH        $460
006    01 040 I                                         32  330DAY         $460
007    01 050 I                                         34  380INVNO       $460
008    01 060 I                                         39  430CUSTNOL1    $460
009    01 070 I                                         44  450STATE       $460
010    01 080 I                                         46  480CITY        $460
011    01 090 I                                         74  802INVAMT      $460
012    01 010 C   01       INVAMT    ADD  TOTAL      TOTAL  72             $460
013    01 020 C   01       INVAMT    ADD  GRPTOT     GRPTOT 72             $460
014    01 010 OOUTPUT  H  201   1P                                         $460
015    01 020 O       OR        OF                                         $460
016    01 030 O                                  53 '      A C C O U N T S  R'  $460
017    01 040 O                                  77 ' E C E I V A B L E  R E '  $460
018    01 050 O                                  88 'G I S T E R'          $460
019    01 060 O         H  1    1P                                         $460
020    01 070 O       OR        OF                                         $460
021    01 080 O                                  25 'CUSTOMER'             $460
022    01 090 O                                  80 'LOCATION        INVOICE'  $460
023    01 100 O                                 109 'INVOICE DATE     INVOICE'  $460
024    01 110 O         H  2    1P                                         $460
025    01 120 O       OR        OF                                         $460
026    01 130 O                                  42 'NUMBER          CUSTOMER '  $460
027    01 140 O                                  46 'NAME'                 $460
028    01 150 O                                  79 ' STATE    CITY      NUMBER'  $460
029    01 160 O                                 108 ' MO     DAY      AMOUNT'  $460
030    02 010 O         D  2    01                                         $460
031    02 020 O                        CUSTNOZ   [illegible]               $460
032    02 030 O                        NAME      [illegible]               $460
033    02 040 O                        STATE Z   59                        $460
034    02 050 O                        CITY  Z   67                        $460
035    02 060 O                        INVNO Z   79                        $460
036    02 070 O                        MONTH Z   90                        $460
037    02 080 O                        DAY   Z   96                        $460
038    02 090 O                        INVAMT   109 '$  ,   0.  '          $460
039    02 100 O         T  2    L1                                         $460
040    02 110 O                        GRPTOT B 109 '$  ,   0.  '          $460
041    02 120 O                                 110 '*'                    $460
042    02 130 O         T  2    LR                                         $460
043    02 140 O                        TOTAL    109 '$  ,   0.  '          $460
044    02 150 O                                 111 '**'                   $460
```

Figure 24. SYSLST Output for RPG Sample Problems (Part 1 of 3)

```
    DOS/360*RPG*CL 3-0                 RPG

                                              SYMBOL   TABLES
RESULTING  INDICATORS

ADDRESS RI      ADDRESS RI      ADDRESS RI      ADDRESS RI      ADDRESS RI      ADDRESS RI      ADDRESS RI

 000011 OF       000014 1P       000015 LR       000016 00       000017 01       00007A L0       00007B L1
 000085 H0       000086 H1       000087 H2       000088 H3       000089 H4       00008A H5       00008B H6
 00008C H7       00008D H8       00008E H9
FIELD  NAMES

ADDRESS FIELD          ADDRESS FIELD          ADDRESS FIELD          ADDRESS FIELD          ADDRESS FIELD

000123  NAME           [illegible]  MONTH     [illegible]  DAY       [illegible]  INVNO     [illegible]  [illegible]
000143  STATE          [illegible]  CITY      [illegible]  INVAMT    [illegible]  TOTAL     [illegible]  [illegible]


LITERALS

ADDRESS LITERAL                     ADDRESS LITERAL                     ADDRESS LITERAL

000153         A C C O U N T S  R   00016B   E C E I V A B L E  R E    000183  G I S T E R
00018E  CUSTOMER                    000196  LOCATION       INVOICE     0001AC  INVOICE DATE      INVOICE
0001C3  NUMBER          CUSTOMER    0001DB  NAME                       0001DF   STATE    CITY     NUMBER
0001F7   MO     DAY     AMOUNT      00020C       ,   .                 000217  *
000218  **


                                        MEMORY MAP

INPUT/OUTPUT INTERCEPT                                     [illegible]
TABLE (INPUT AND OUTPUT)                                   [illegible]
DETERMINE RECORD TYPE                                      [illegible]
DATA SPECIFICATION                                         [illegible]
GET INPUT RECORD                                           [illegible]
DETAIL CALCULATIONS                                        [illegible]
TOTAL CALCULATIONS                                         [illegible]
DETAIL LINES                                               [illegible]
TOTAL LINES                                                [illegible]
INPUT/OUTPUT REQUEST BLOCKS POINTER                        0015A0
LOCATION OF DTF TABLE POINTERS                             000DF0
INPUT/OUTPUT INTERFACE ROUTINES                            000ECC
WORK AREA POINTER                                          0016BC
OVERFLOW BYPASS                                            000BA6
CONTROL LEVEL                                              000668
TABLE(ASSEMBLE 4)                                          000C8C
TEST ZONE (BCD)                                            [illegible]
OVERFLOW LINES                                             [illegible]
LINKAGE PROGRAM                                            [illegible]


PROGRAM LENGTH  [illegible]


    DOS/360*RPG*CL 3-0                 RPG

*END OF COMPILATION*


// EXEC LNKEDT
```

Figure 24. SYSLST Output for RPG Sample Problems (Part 2 of 3)

```
JOB  RPG                  DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN    MAP
LIST           INCLUDE       IJCFCIZO                                   $4600119
LIST           INCLUDE       IJDFYPZZ                                   $4600120
LIST   ENTRY
```

```
01/01/68   PHASE   XFR-AD  LOCORE  HICORE  DSK-AD   ESD TYPE  LABEL     LOADED  REL-FR

        PHASE***   003518  C02000  0038EB  19 4 1   CSECT     $460      002000  002000

                                                    CSECT     IJCFCIZO  003708  003708
                                                    * ENTRY   IJCFZIZO  003708

                                                    CSECT     IJDFYPZZ  003870  003870
                                                    * ENTRY   IJDFYZZZ  003870
```

```
// EXEC
```

```
                          A C C O U N T S   R E C E I V A B L E   R E G I S T E R

         CUSTOMER                             LOCATION      INVOICE       INVOICE DATE      INVOICE
          NUMBER         CUSTOMER NAME       STATE  CITY     NUMBER          MO    DAY       AMOUNT

          10712       AMALGAMATED CORP         33    61      11603           11    10    $   389.25

                                                                                         $   389.25*

          11315       BROWN WHOLESALE          30   231      12324           12    28    $   802.08

          11315       BROWN WHOLESALE          30   231      99588           12    14    $   261.17

                                                                                         $ 1,063.25*

          11897       FARM IMPLEMENTS          47    77      10901           10    18    $    27.63

                                                                                         $    27.63*

          18530       BLACK OIL                16    67      11509           11     8    $   592.95

          18530       BLACK OIL                16    67      12292           12    23    $   950.97

                                                                                         $ 1,543.92*

          20716       LEATHER BELT CO          36   471      11511           11     8    $   335.63

          20716       LEATHER BELT CO          36   471      12263           12    17    $   121.75

                                                                                         $   457.38*

          29017       GENERAL MFG CO            6    63      11615           11    14    $   440.12

          29017       GENERAL MFG CO            6    63      11676           11    23    $   722.22

                                                                                         $ 1,162.34*

          29054       A-B-C DIST CO            25    39       9689            9    11    $   645.40

          29054       A-B-C DIST CO            25    39      11605           11    11    $   271.69

          29054       A-B-C DIST CO            25    39      12234           12    14    $   559.33

                                                                                         $ 1,476.42*

                                                                                         $ 6,120.19**
```

```
EOJ RPG
```

Figure 24. SYSLST Output for RPG Sample Problems (Part 3 of 3)

PL/I

Program Name is Z.PL1. The PL/I sample problem produces a table of mathematical functions: SQRT(x^2+1),x^2,x^3,SIN(x),COS(x),SQRT(x), and SQRT(x^3). To execute the PL/I sample problem the following job control cards are needed:

```
// JOB PL/I SAMPLE
// OPTION LINK,NODECK,SYM,LISTX,48C,XREF
// EXEC PL/I
// PROCESS STMT
   PL/I source deck (38 cards)
/*
// EXEC LNKEDT
// EXEC
   PL/I data (1 card)
/*
/&
```

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 25.

- Job control cards
- Source program listing
- Symbol table listing
- Offset table listing
- Object program listing
- External symbol table
- Block table (automatic storage requirements)
- Linkage editor storage map
- Table produced by object program

```
// JOB PL/I SAMPLE
// OPTION LINK,LIST,LISTX,SYM,NODECK,48C
// EXEC PL/I
```

```
     DOS PL/I COMPILER  360N-PL-464 CL3-4            PL/I              02/26/69           PAGE 001

                                              O P T I O N S   L I S T

* PROCESS STMT,OPT,XREF

OPTIONS TAKEN ARE LIST,LISTX,SYM,XREF,ERRS,48C,OPT,STMT.
```

```
     DOS PL/I COMPILER  360N-PL-464 CL3-4            PL/I              02/26/69           PAGE 002

/*PL/I SAMPLE PROGRAM FOR DOS/TOS */

                    /*PL/I SAMPLE PROGRAM FOR DCS/TCS */                                 $4640001
    1               PL1..PROCEDURE OPTIONS (MAIN) ,.                                     $4640002
    2                           DECLARE  ( X, BEGIN, END, STEP, A(5), B(2) )             $4640003
                                     FLOAT BINARY (21) ,.                                $4640004
                    /*          THIS PROGRAM COMPUTES A SMALL TABLE CONTAINING A FEW     $4640005
                                MATH. FUNCTIONS.                                         $4640006
                                INPUT DATA.. BEGIN - START POINT OF TABLE ,              $4640007
                                             END   - END POINT TABLE   ,                 $4640008
                                             STEP  - STEP WIDTH IN TABLE  ,              $4640009
                                THE ABSOLUTE VALUES OF BEGIN AND END MAY NOT BE GREATER  $4640010
                                THAN  999.999, THE ABSOLUTE VALUE CF STEP MAY NOT BE LESS $4640011
                                THAN  0.001 .                                            $4640012
                                                                                      */ $4640013
    3                           GET EDIT (BEGIN, END, STEP)(3 F(8,3)) ,.                 $4640014
    4                           PUT EDIT ('X      SQRT(X**2+1)       X**2','X**3','SIN(X) $4640015
                     COS(X)        SQRT(X)      SQRT(X**3)') (X(4),A(41),A(14),A(61)) ,. $4640016
    5                           IF ABS(BEGIN) GE 1000 OR ABS(STEP) LT .000999            $4640017
                                   THEN GO TO INPUTERROR ,.                              $4640018
    6                           DO X=BEGIN TO END+STEP/1E3 BY STEP ,.                    $4640019
    7                           A(2)=X*X ,.                                              $4640020
    8                           A(1)=SQRT(A(2)+1),.                                      $4640021
    9                           A(3)=A(2)*X  ,.                                          $4640022
   10                           A(4)=SIN(X) ,.                                           $4640023
   11                           A(5)=COS(X) ,.                                           $4640024
   12                           PUT EDIT (X,A) (F(8,3),F(13,3),2 E(16,5),2 F(13,6)),.    $4640025
   13                           IF X LT 0 THEN  GOTO NEG ,.                              $4640026
   14                           B(1)=SQRT(X)     ,.                                      $4640027
   15                           B(2)=SQRT(A(3)) ,.                                       $4640028
   16                           PUT EDIT (B,' ') (F(12,4), E(16,5),A(13)) ,.             $4640029
   17                           GOTO ENDLOOP   ,.                                        $4640030
   18               NEG..       PUT EDIT ('--','--') (X(7), A(15), A(19)) ,.             $4640031
   19               ENDLOOP.. END  /* DO LOOP */ ,.                                      $4640032
   20               PUT SKIP EDIT (' NORMAL END')(A),.                                   $4640033
   21                        RETURN /* NORMAL END */ ,.                                  $4640034
   23               INPUTERROR..                                                         $4640035
                                PUT EDIT ('INPUT DATA ERROR') (A(12)) ,.                 $4640036
   24                           STOP /* END ON WRONG INPUT */ ,.                         $4640037
   25                           END  /* TABLE */  ,.                                     $4640038
```

```
     DOS PL/I COMPILER  360N-PL-464 CL3-4            PL/I              02/26/69           PAGE 003

                    S Y M B O L    T A B L E    L I S T I N G

PL1                          0100 00 0 ENTRY     ARITHM. DECIMAL FLOAT   6            EXT
X                            0101 01 1           ARITHM. BINARY  FLOAT  21    AUTOM.  INT
BEGIN                        0102 01 1           ARITHM. BINARY  FLOAT  21    AUTOM.  INT
END                          0103 01 1           ARITHM. BINARY  FLOAT  21    AUTOM.  INT
STEP                         0104 01 1           ARITHM. BINARY  FLOAT  21    AUTOM.  INT
A                            0105 01 1 ARRAY     ARITHM. BINARY  FLOAT  21    AUTOM.  INT
B                            0106 01 1 ARRAY     ARITHM. BINARY  FLOAT  21    AUTOM.  INT
NEG                          0107 01 1           LABEL   CONST.                       INT
ENDLOOP                      0108 01 1           LABEL   CONST.                       INT
INPUTERROR                   0109 01 1           LABEL   CONST.                       INT
```

Figure 25. SYSLST Output for PL/I Sample Programs (Part 1 of 4)

```
        DOS PL/I COMPILER  360N-PL-464 CL3-4             PL/I              02/26/69          PAGE 004

                       C R O S S    R E F E R E N C F    L I S T I N G

 A                             0105 01    7    8    8    9    9   10   11   12   15
 B                             0106 01   14   15   16
 BEGIN                         01C2 01    3    5    6
 END                           0103 01    3    6
 ENDLCOP                       0108 01   17   19
 INPUTERRCR                    0109 01    5   23
 NEG                           0107 01   13   18
 PL1                           0100 00    1
 STEP                          0104 01    3    5    6    6
 X                             0101 01    6    7    7    9   10   11   12   13   14
```

```
        DOS PL/I COMPILER  360N-PL-464 CL3-4             PL/I              02/26/69          PAGE 005

         INTERNAL NAME  OFFSET         TYPE            MODULE OFFSET        OFFSET TABLE

         0100           001C           STATIC          00042C
         0101           011C           AUTOMATIC
         0102           0120           AUTOMATIC
         0103           0124           AUTOMATIC
         0104           0128           AUTOMATIC
         0105           0130           AUTOMATIC
         0106           0148           AUTOMATIC
```

```
        DOS PL/I COMPILER  360N-PL-464 CL3-4             PL/I              02/26/69          PAGE 006

 LOC.     OBJECT CODE          LABEL    OP.    OPERANDS

 000000   05F0                          BALR   F,0
 000002                        L'0100'  BEGIN  OF BLOCK  01
 000002   0700                          BCR    0,0
 000004   45E0 F00A                     BAL    E,X'00A'(F)
 000008   00000410                      DC     A(N'FFFF')
 00000C   58C0 E000                     L      C,X'000'(E)
 000010   189F                          LR     9,F
 000012   1831                          LR     3,1
 000014   58F0 C05C                     L      F,N'0011'
 000018   05E0                          BALR   E,0
 00001A   41E0 E00E                     LA     E,X'00E'(E)
 00001E   051F                          BALR   1,F
 000020   83                            DC     X'83'
 000021   000428                        DC     AL3(N'010A')
 000024   00000150                      DC     LENGTH OF DSA OF BLOCK 01
 000028   4110 C02C                     LA     1,X'02C'(C)
 00002C   58F0 C054                     L      F,N'0016'
 000030   05EF                          BALR   E,F
 000032   D203 D050 3000                MVC    X'050'(04,D),X'000'(3)
 000038   4700 0001                     BC     0,X'001'
                                                                             STATEMENT NUMBER  1
```

```
        DOS PL/I COMPILER  360N-PL-464 CL3-4             PL/I              02/26/69          PAGE 014

  SYMBOL    TYPE   ESID    ADDR     LENGTH  ESID       EXTERNAL SYMBOL TABLE

  PL1        SD    0001    000000   000568
 IJKSZCA     ER    0002
 IJKSZCM     ER    0003
 IJKSZCT     ER    0004
 IJKSZLM     ER    0005
 IJKTFDM     ER    0006
 IJKTFPM     ER    0007
 IJKTSTM     ER    0008
 IJKVTCM     ER    0009
 IJKVECM     ER    000A
 IJKTSTR     ER    000B
 IJKVCTM     ER    000C
 IJKVCEM     ER    000D
 IJKSYSI     ER    000E
 IJKSYSA     ER    000F
 IJKQQSM     ER    0010
 IJKQSSD     ER    0011
 IJKQSSB     ER    0012
 IJKSZBA     ER    0013
```

Figure 25. SYSLST Output for PL/I Sample Programs (Part 2 of 4)

```
     DOS PL/I COMPILER  360N-PL-464 CL3-4          PL/I          02/26/69          PAGE 015

BLOCK     LENGTH OF DSA      BLOCKTABLE
 01         0150
```

```
     DOS PL/I COMPILER  360N-PL-464 CL3-4          PL/I          02/26/69          PAGE 016

5W01I    SUCCESSFUL    COMPILATION
```

```
// EXEC LNKEDT      .
```

```
JOB  PL/I      02/26/69   DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN      MAP
LIST   AUTOLINK    IJKQQSM
LIST   AUTOLINK    IJKQSSD
LIST   AUTOLINK    IJKSYSA
LIST   AUTOLINK    IJJCPIN
LIST   AUTOLINK    IJKSYSI
LIST   AUTOLINK    IJKSZBA
LIST   AUTOLINK    IJKSZLM
LIST   AUTOLINK    IJKTFDM
LIST   AUTOLINK    IJKTFMM
LIST   AUTOLINK    IJKTSTM
LIST   AUTOLINK    IJKTXCF
LIST   AUTOLINK    IJKVCEM
LIST   AUTOLINK    IJKVCTM
LIST   AUTOLINK    IJKVECM
LIST   AUTOLINK    IJKVTCM
LIST   AUTOLINK    IJKXTBM
LIST   ENTRY
```

Figure 25. SYSLST Output for PL/I Sample Programs (Part 3 of 4)

```
02/26/69   PHASE   XFR-AD  LOCORE  HICORE  DSK-AD   ESD TYPE  LABEL    LOADED  REL-FR

        PHASE***   002800  002800  004717  28 06 2  CSECT     PL1      002800  002800

                                                    CSECT     IJKSZLM  003878  003878

                                                    CSECT     IJKTFDM  003888  003888

                                                    CSECT     IJKTFMM  003A98  003A98
                                                       ENTRY  IJKTFMR  003AAC

                                                    CSECT     IJKTSTM  003B60  003B60
                                                       ENTRY  IJKTSTR  003D64
                                                       ENTRY  IJKTSTN  003BDA

                                                    CSECT     IJKVTCM  004578  004578

                                                    CSECT     IJKVECM  0043E0  0043E0

                                                    CSECT     IJKVCTM  004258  004258

                                                    CSECT     IJKVCEM  003E80  003E80

                                                    CSECT     IJKSYSI  0030F0  0030F0

                                                    CSECT     IJKSYSA  002F48  002F48

                                                    CSECT     IJKQQSM  002D68  002D68

                                                    CSECT     IJKQSSA  002E18  002E18
                                                       ENTRY  IJKQSSD  002E40
                                                       ENTRY  IJKQSSB  002E20
                                                    *  ENTRY  IJKQSSC  002E38

                                                    CSECT     IJJCP1N  003018  003018
                                                       ENTRY  IJJCP3   003018

                                                    CSECT     IJKTXCF  003E08  003E08
                                                    *  ENTRY  IJKTXCR  003E30
                                                    *  ENTRY  IJKTXCW  003E42

                                                    CSECT     IJKSZCN  003188  003188
                                                       ENTRY  IJKSZCA  0033B6
                                                       ENTRY  IJKSZCM  0033BE
                                                       ENTRY  IJKSZCT  0033DA
                                                       ENTRY  IJKSZBA  0033BA
                                                       ENTRY  IJKZWSA  003468
                                                       ENTRY  IJKSZCS  00323E
                                                    *  ENTRY  IJKSZCI  00321A
                                                    *  ENTRY  IJKSZCP  003348
                                                    *  ENTRY  IJKTCLM  0033FC
                                                       ENTRY  IJKTOPM  0033EC
                                                    *  ENTRY  IJKZWSI  00358C

                                                    CSECT     IJKXTBM  004698  004698
```

```
// EXEC
```

```
    X     SQRT(X**2+1)     X**2          X**3          SIN(X)      COS(X)     SQRT(X)    SQRT(X**3)
   .000      1.000     .00000E+00    .00000E+00     .000000    1.000000     .0000     .00000E+00
   .100      1.005     9.99999E-03   9.99999E-04    .099833     .995004     .3162     3.16228E-02
   .200      1.020     4.00000E-02   7.99999E-03    .198669     .980067     .4472     8.94426E-02
   .300      1.044     8.99999E-02   2.70000E-02    .295520     .955337     .5477     1.64317E-01
   .400      1.077     1.60000E-01   6.39999E-02    .389418     .921061     .6325     2.52982E-01
   .500      1.118     2.50000E-01   1.25000E-01    .479425     .877583     .7071     3.53553E-01
   .600      1.166     3.60000E-01   2.16000E-01    .564642     .825336     .7746     4.64758E-01

  2.500      2.693     6.24996E+00   1.56248E+01    .598479    -.801139    1.5811     3.95283E+00
  2.600      2.786     6.75995E+00   1.75758E+01    .515509    -.856884    1.6124     4.19235E+00
  2.700      2.879     7.28995E+00   1.96828E+01    .427389    -.904068    1.6432     4.43653E+00
  2.800      2.973     7.83994E+00   2.19518E+01    .334998    -.942219    1.6733     4.68527E+00
  2.900      3.068     8.40994E+00   2.43887E+01    .239260    -.970956    1.7029     4.93849E+00
  3.000      3.162     8.99993E+00   2.69997E+01    .141131    -.989991    1.7320     5.19612E+00
NORMAL END

EOJ PL/I                                                 14.06.55,DURATION 00.02.19
```

Figure 25. SYSLST Output for PL/I Sample Programs (Part 4 of 4)

Assembler 1

Program Name is Z.AS1. The Assembler 1 sample problem generates numbers from 2000 to 0001 in descending order and writes them on tape as 15 character unblocked records (11 blanks followed by 4 zone digits, unsigned). To execute the Assembler 1 sample problem the following job control cards are needed:

```
// JOB ASSEMBLE SAMPLE 1.
// OPTION LINK,LIST,LOG,NOXREF
// EXEC ASSEMBLY
   ASSEMBLER SOURCE DECK (23 cards)
/*
// EXEC LNKEDT
// PAUSE ASSGN SYS002 TO A 9 TRACK TAPE DRIVE
// ASSGN SYS002,X'cuu'
// MTC REW,SYS002
// EXEC
/&
```

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 26:

- Job control cards
- External symbol dictionary
- Source program listing
- Relocation dictionary
- Linkage editor storage map

```
// JOB ASSEMBLE  SAMPLE 1
// OPTION LINK,LIST,LOG,NOXREF
// EXEC ASSEMBLY
```

```
                                                  EXTERNAL SYMBOL DICTIONARY
  SYMBOL  TYPE ID  ADDR   LENGTH LD ID

           PC  01 000000 000124
 IJFFZZZZ  ER  02
 IJFFZZZZ  SD  03 000128 0002BA
```

```
SMPL    DOS ASSEMBLER TEST

  LOC   OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT                                              DOS CL3-0

                                        2          PRINT NOGEN                                               $4650002
000000                                  3          START 0                                                   $4650003
000000 0580                             4 TESTGEN  BALR  8,0                    LOAD BASE REG                $4650004
000002                                  5          USING *,8                                                 $4650005
000002 41D0 80C6              000C8     6          LA    13,SAVEAREA                                         $4650006
                                        7          OPEN  OUTFILE                OPEN OUTPUT FILE             $4650007
000016 4130 07D0              007D0    15          LA    3,2000                 GET STARTING RECORD VALUE    $4650008
00001A 4E30 80BE              000C0    16 NEXT     CVD   3,DWD                  CONVERT VALUE TO DEC.        $4650009
00001E F337 200B 80BE 0000B 000C0      17          UNPK  11(4,2),DWD            UNPACK IT                    $4650010
000024 96F0 200E        0000E          18          OI    14(2),X'F0'            MASK SIGN BIT                $4650011
                                       19          PUT   OUTFILE                WRITE RECORD ON TAPE         $4650012
000034 4630 8018              0001A    24          BCT   3,NEXT                 TEST FOR TASK COMPLETE       $4650013
                                       25          CLOSE OUTFILE                CLOSE TAPE FILE              $4650014
                                       33          EOJ                          RETURN TO CONTROL PROGRAM    $4650015
                                       36 OUTFILE  DTFMT BLKSIZE=15,DEVADDR=SYS002,FILABL=NO,IOAREA1=A1,    X$4650016
                                                         IOAREA2=A2,IOREG=(2),TYPEFLE=OUTPUT                 $4650017
                                       68          MTMOD RECFORM=FIXUNB                                      $4650018
00009E 404040404040404040             292 A1       DC    15C' '                                              $4650019
0000AD 404040404040404040             293 A2       DC    15C' '                                              $4650020
0000C0                                294 DWD      DS    D                                                   $4650021
0000C8                                295 SAVEAREA DS    9D                                                  $4650022
000000                                296          END   TESTGEN                                             $4650023
000110 5B5BC2D6D7C5D540               297                =C'$$BOPEN '
000118 5B5BC2C3D3D6E2C5               298                =C'$$BCLOSE'
000120 00000048                       299                =A(OUTFILE)
```

```
                                                      RELOCATION DICTIONARY

    POS.ID   REL.ID   FLAGS   ADDRESS

      01       01      0C     00001C
      01       01      0C     000040
      01       01      0C     000050
      01       02      18     000059
      01       01      C8     00006D
      01       01      08     000081
      01       01      0C     000088
      01       01      0C     00008C
      01       01      0C     000094
      01       01      0C     000120

   NO STATEMENTS FLAGGED IN THIS ASSEMBLY
```

Figure 26. SYSLST Output for Assembler (AS1) Sample Programs (Part 1 of 2)

```
// EXEC LNKEDT
```

```
JOB  ASSEMBLE              DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN     MAP
LIST   ENTRY
```

```
          PHASE   XFR-AD  LOCORE  HICORE  DSK-AD    ESD TYPE  LABEL     LOADED  REL-FR

       PHASE***   002000  002000  0023E1  20 3 1    CSECT               002000  002000

                                                    CSECT     IJFFZZZZ  002128  002000
```

```
// PAUSE  // ASSGN SYS002 TO A 9 TRACK TAPE DRIVE
// ASSGN SYS002,X'180'
// MTC REW,SYS002
// EXEC
```

```
EOJ ASSEMBLE
```

Figure 26. SYSLST Output for Assembler (AS1) Sample Programs (Part 2 of 2)

Tape Sort/Merge

Program Name is Z.SM1. The tape sort/merge sample problem rearranges the records produced by the Assembler 1 program in ascending sequence and writes then on another tape, five records per block. To execute the Tape Sort/Merge sample problem the following job control cards are needed:

```
// JOB SORT
// PAUSE ASSGN SYS002 TO SAME TAPE DRIVE AS IN PREVIOUS JOB
// ASSGN SYS002,X'cuu'
// PAUSE ASSGN SYS001,SYS003,SYS004,SYS005 TO DIFF TAPES
// ASSGN SYS001,X'cuu'
// ASSGN SYS003,X'cuu'
// ASSGN SYS004,X'cuu'
// ASSGN SYS005,X'cuu'
// EXEC TSRT
   S/M CONTROL CARDS (6 cards)
/*
/&
```

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 27.

- Job control cards
- Sort control cards
- Sort parameters
- Computed constants

```
// JOB SORT
// PAUSE  // ASSGN SYS002 TO SAME TAPE DRIVE AS IN PREVIOUS JOB
// ASSGN SYS002,X'180'
// PAUSE // ASSGN SYS001///SYS003///SYS004///SYS005 TO DIFF TAPES
// ASSGN  SYS001,X'183'
// ASSGN SYS003,X'184'
// ASSGN SYS004,X'185'
// ASSGN SYS005,X'186'
// EXEC TSRT
```

```
7T01I                  **TAPE AND DISK OPERATING SYSTEM SORT/MERGE PROGRAM**


7T05I                            **CONTROL CARD LISTING       **

**CONTROL CARD  01 SORT     FORMAT=BI,FIELDS=(12,4,A),WORK=3
**CONTROL CARD  02 RECORD   TYPE=F,LENGTH=15
**CONTROL CARD  03 INPFIL   VOLUME=1,BLKSIZE=(15,X),OPEN=RWD
**CONTROL CARD  04 OUTFIL   BLKSIZE=75,CLOSE=RWD
**CONTROL CARD  05 OPTION   PRINT,LABEL=(U,U,U)
**CONTROL CARD  06 END

7T06I                              **SORT  PARAMETERS**

INPUT 1 VOLUME IS         1        THE OUTPUT BLOCKSIZE IS   75      THE INPUT BLOCKSIZE IS    15
INPUT BLOCK FORMAT IS     X        OPEN INPUT FILE           RWD     CLOSE OUTPUT FILE         RWD
THE RECORD TYPE IS        F        THE RECORD LENGTH 1 IS    15
THE 01 CF LOCATION IS     12       THE 01 CF LENGTH IS       4       THE 01 CF SEQUENCE IS     A
WORK DRIVES AVAILABLE     3        THE DATA FORMAT IS        BI      INPUT LABELS ARE          U
WORK LABELS ARE           U        OUTPUT LABELS ARE         U

7T08I **COMPUTED CONSTANTS**
MFS =  1471025
BLZ =  0010875
N   =  0000005
C   =  0001730
A   =  0000002

7T08I ** END OF ASSIGNMENT PHASE **
```

```
EOJ SORT
```

Figure 27. SYSLST Output for Tape Sort/Merge Sample Problems

Tape-to-Printer Utility

Program Name is Z.UTTPPR1. The Tape-to-Printer Utility sample problem lists the numbers handled by the assembler 1 and tape sort/merge sample problems, allowing the user to check for proper functioning of all three programs. To execute the tape to printer utility sample problem, the following job control cards are needed:

```
// JOB TPPR
// PAUSE ASSGN SYS004 TO SYS001 OF PREVIOUS JOB
// ASSGN SYS004,X'cuu'
// PAUSE ASSGN SYS005 TO THE PRINTER
// ASSGN SYS005,X'cuu'
// UPSI 1000
// EXEC TPPR
   T-P CONTROL CARDS (2 cards)
/*
/&
```

The user must reply IT IS EOF to the logged message by entering Ⓑ (alternate code 5) from the IBM 1052 printer-keyboard.

SYSLOG output is shown in Figure 17.

A summary of the SYSLST output shown in Figure 28 follows:

- Job control cards
- Tape-to-printer control cards
- Utility parameter statements (11 lines)
- Numbers 0001-2000 (400 lines)
- IS IT EOF
- REPLY Y (Ⓑ is not printed)
- END OF DATA
- NUMBER OF INPUT BLOCKS PROCESSED 000400
- NUMBER OF OUTPUT BLOCKS PROCESSED 000400
- END OF JOB

```
// JOB TPPR
// PAUSE  // ASSGN SYS004 TO SYS001 OF PREVIOUS JOB
// ASSGN SYS004,X'183'
// PAUSE  // ASSGN SYS005 TO THE PRINTER
// ASSGN SYS005,X'00E'
// UPSI 1000
// EXEC TPPR
```

```
// UTP TL,FF,A=(75,75),B=(120),IR,OC,S1,PY                              $4620001
// END                                                                  $4620002
TAPE TO PRINT UTILITY
INPUT RECORD LENGTH 0075
OUTPUT RECORD LENGTH  0120
INPUT BLOCK LENGTH 00075
OUTPUT BLOCK LENGTH 00120
INPUT OPTION REWIND
OUTPUT OPTION PRINT CHARACTER
2 INPUT,2 OUTPUT AREAS ASSIGNED
RECORD FORMAT FIXED
TYPE LIST
STARTING RECORD NUMBER  00000001
```

```
0001      0002      0003      0004      0005
0006      0007      0008      0009      0010
0011      0012      0013      0014      0015
0016      0017      0018      0019      0020
0021      0022      0023      0024      0025
0026      0027      0028      0029      0030
0031      0032      0033      0034      0035
0036      0037      0038      0039      0040
0041      0042      0043      0044      0045
0046      0047      0048      0049      0050
0051      0052      0053      0054      0055
0056      0057      0058      0059      0060
0061      0062      0063      0064      0065
0066      0067      0068      0069      0070
0071      0072      0073      0074      0075
0076      0077      0078      0079      0080
0081      0082      0083      0084      0085
0086      0087      0088      0089      0090
0091      0092      0093      0094      0095
0096      0097      0098      0099      0100
0101      0102      0103      0104      0105
0106      0107      0108      0109      0110
0111      0112      0113      0114      0115
0116      0117      0118      0119      0120
0121      0122      0123      0124      0125
0126      0127      0128      0129      0130
0131      0132      0133      0134      0135
0136      0137      0138      0139      0140
0141      0142      0143      0144      0145
0146      0147      0148      0149      0150
0151      0152      0153      0154      0155
0156      0157      0158      0159      0160
0161      0162      0163      0164      0165
0166      0167      0168      0169      0170
0171      0172      0173      0174      0175
0176      0177      0178      0179      0180
0181      0182      0183      0184      0185
0186      0187      0188      0189      0190
```

```
1911      1912      1913      1914      1915
1916      1917      1918      1919      1920
1921      1922      1923      1924      1925
1926      1927      1928      1929      1930
1931      1932      1933      1934      1935
1936      1937      1938      1939      1940
1941      1942      1943      1944      1945
1946      1947      1948      1949      1950
1951      1952      1953      1954      1955
1956      1957      1958      1959      1960
```

Figure 28. SYSLST Output for TP-PR Sample Problems (Part 1 of 2)

```
1961    1962    1963    1964    1965
1966    1967    1968    1969    1970
1971    1972    1973    1974    1975
1976    1977    1978    1979    1980
1981    1982    1983    1984    1985
1986    1987    1988    1989    1990
1991    1992    1993    1994    1995
1996    1997    1998    1999    2000

8001D IS IT EOF
REPLY Y
 END OF DATA
```

```
NUMBER OF INPUT BLOCKS PROCESSED  C00400
NUMBER OF OUTPUT BLOCKS PROCESSED C00400
END OF JOB
```

```
EOJ TPPR
```

Figure 28. SYSLST Output for TP-PR Sample Problems (Part 2 of 2)

Assembler 2

Program Name is Z.AS2. The Assembler 2 sample problem generates numbers from 2000 to 0002 in descending order and writes them on disk. To execute the Assembler 2 sample problem, the following job control cards are needed:

```
                                                         Col. 54    Col. 72
                                                         |          |
                                                         v          v
// JOB ASSEMBLE SAMPLE 2
// OPTION LINK,LIST,LOG,NOXREF
// EXEC ASSEMBLY
   ASSEMBLER SOURCE DECK (23 cards)
/*
// EXEC LNKEDT
// PAUSE ASSGN SYS002 to SCRATCH PACK
// ASSGN SYS002,X'cuu'
// VOL SYS002,OUTFILE
// DLAB 'SAMPLE PROBLEM FILE OF 2000 RECORDS             1hhhhhh',    C
               Col. 16
               0001,66060,66080,' SAMPLE TEST
// XTENT 1,000,000180000,000184002,'hhhhhh',SYS002
```

Note: Replace hhhhhh in the DLAB and XTENT cards with the volume serial number of the disk pack.

```
// EXEC
/&
```

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 29:

- Job control cards
- External symbol dictionary
- Source program listing
- Relocation dictionary
- Linkage editor storage map

```
// JOB ASSEMBLE  SAMPLE 2
// OPTION LINK,LIST,LOG,NOXREF
// EXEC ASSEMBLY
```

```
                                                                     EXTERNAL SYMBOL DICTIONARY
 SYMBOL  TYPE ID  ADDR  LENGTH LD ID


          PC  01 000000 00017C
IJGFOZZZ  ER  02
IJGFOZZZ  SD  03 000180 000200
```

```
       DOS ASSEMBLER TEST

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                         DOS CL3-0

                                       2           PRINT NOGEN                                                      $4650102
000000                                 3           START 0                                                          $4650103
000000 0580                            4 TESTGEN   BALR  8,0                     LOAD THE BASE REGISTER.            $4650104
000002                                 5           USING *,8                                                        $4650105
000002 41D0 811E             00120     6           LA    13,SAVEAREA                                                $4650106
                                       7           OPEN  OUTFILE                 OPEN THE OUTPUT FILE.              $4650107
000016 4130 07D0             007D0    15           LA    3,2000                  GET STARTING RECORD VALUE.         $4650108
00001A 4E30 8116             00118    16 NEXT      CVD   3,DWD                   CONVERT THE VALUE TO DECIMAL.      $4650109
00001E F337 200B 8116 0000B 00118     17           UNPK  11(4,2),DWD             UNPACK THE VALUE.                  $4650110
000024 96F0 200E       0000E          18           OI    14(2),X'F0'             MASK THE SIGN BIT.                 $4650111
                                      19           PUT   OUTFILE                 WRITE A RECORD ON DISK.            $4650112
000034 4630 8018             0001A    24           BCT   3,NEXT                  TEST FOR TASK COMPLETE.            $4650113
                                      25           CLOSE OUTFILE                 CLOSE THE DISK FILE.               $4650114
                                      33           EOJ                           RETURN TO THE CONTROL PROGRAM.     $4650115
                                      36 OUTFILE   DTFSD BLKSIZE=23,IOAREA1=A1,IOAREA2=A2,TYPEFLE=OUTPUT,          C$4650116
                                                         IOREG=(2),RECFORM=FIXUNB,DEVICE=2311                       $4650117
                                      85           SDMODFO                                                          $4650118
0000E8 4040404040404040              264 A1        DC    23C' '                                                     $4650119
0000FF 4040404040404040              265 A2        DC    23C' '                                                     $4650120
000118                               266 DWD       DS    D                                                          $4650121
000120                               267 SAVEAREA  DS    9D                                                         $4650122
000000                               268           END   TESTGEN                                                    $4650123
000168 5B5BC2D6D7C5D540              269                 =C'$$BOPEN '
000170 5B5BC2C3D3D6E2C5              270                 =C'$$BCLOSE'
000178 00000048                      271                 =A(OUTFILE)
```

Figure 29. SYSLST Output for Assembler (AS2) Sample Problems (Part 1 of 2)

```
                                          RELOCATION DICTIONARY

 POS.ID   REL.ID   FLAGS   ADDRESS

   01       01      0C     000010
   01       01      0C     000040
   01       01      0C     000050
   01       02      18     000059
   01       01      08     000071
   01       01      0C     000074
   01       01      0C     0000A0
   01       01      0C     0000A8
   01       01      08     0000B1
   01       01      08     0000B9
   01       01      08     0000C1
   01       01      08     0000C9
   01       01      08     0000D1
   01       01      08     0000D9
   01       01      08     0000E1
   01       01      0C     000178

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
```

```
// EXEC LNKEDT
```

```
JOB  ASSEMBLE            DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN     MAP
LIST   ENTRY
```

```
        PHASE  XFR-AD  LOCORE  HICORE  DSK-AD   ESD TYPE  LABEL     LOADED  REL-FR

     PHASE***  002000  002000  002388  22 1 2   CSECT               002000  002000

                                                CSECT     IJGFOZZZ  002180  002000
```

```
// PAUSE // ASSGN SYS002 TO SCRATCH PACK
// ASSGN SYS002,X'291'
// VOL SYS002,OUTFILE
// DLAB 'SAMPLE PROBLEM FILE OF 2000 RECORDS         1111111',        C
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000180000,000184002,'111111',SYS002
// EXEC
```

```
EOJ ASSEMBLE
```

Figure 29. SYSLST Output for Assembler (AS2) Sample Problems (Part 2 of 2)

Disk Sort/Merge

Program Name is Z.SM2. The disk sort/merge sample problem rearranges in ascending sequence the numbers generated by the Assembler 2 problem. To execute the disk sort/merge sample problem the following job control cards are needed:

```
                                                  Col. 54      Col. 72
                                                  |            |
                                                  v            v
// JOB DSORT
// PAUSE ASSGN SYS002 AND SYS004 to SCRATCH PACK
// ASSGN SYS002,X'cuu'
// ASSGN SYS004,X'cuu'
// VOL SYS002,FILEA
// DLAB 'SAMPLE PROBLEM FILE OF 2000 RECORDS        1hhhhhh',    C
            Col. 16
            0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000180000,000184002,'hhhhhh',SYS002
// VOL SYS002,FILEW
// DLAB 'SAMPLE PROBLEM WORK AREA FOR SORT RUN      1hhhhhh',    C
            Col. 16
            0001,66080,66080,' SAMPLE TEST ',DA
// XTENT 1,000,000184003,000186005,'hhhhhh',SYS002
// VOI SYS004,FIIEO
// DLAB 'SORTED FILE OF 2000 RECORDS FOR UTILITIES  1hhhhhh',    C
            Col. 16
            0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000180000,000184002,'hhhhhh',SYS004
```

Note: Replace hhhhhh in the DLAB and XTENT cards with the volume serial number of the disk pack.

```
// EXEC DSORT
   S/M CONTROL CARDS (6 cards)
/*
/&
```

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 30:

- Job control cards
- Sort control cards
- Sort parameters
- Computed constants

```
// JOB DSORT
// PAUSE  // ASSGN SYS002 AND SYS004 TO SCRATCH PACK
// ASSGN SYS002,X'291'
// ASSGN SYS004,X'291'
// VOL SYS002,FILEA
// DLAB 'SAMPLE PROBLEM FILE OF 2000 RECORDS          1111111',         C
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000180000,000184002,'111111',SYS002
// VOL SYS002,FILEW
// DLAB 'SAMPLE PROBLEM WORK AREA FOR SORT RUN        1111111',         C
               0001,66080,66080,' SAMPLE TEST ',DA
// XTENT 1,000,000184003,000186005,'111111',SYS002
// VOL SYS004,FILEO
// DLAB 'SORTED FILE OF 2000 RECORDS FOR UTILITIES    1111111',         C
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000180000,000184002,'111111',SYS004
// EXEC DSORT
```

```
7D631        ** JOB  DSORT    **        *** CONTROL CARD INFORMATION ***    ** 01/01/68 **

                     ***  CONTROL CARDS   ***

 SORT FIELDS=(12,4,A),FORMAT=BI,SIZE=2000                                    $4500001
 RECORD TYPE=F,LENGTH=(15,,15)                                               $4500002
 INPFIL BLKSIZE=(15,X),INPUT=D                                               $4500003
 OUTFIL BLKSIZE=75,OUTPUT=D                                                  $4500004
 OPTION PRINT,LABEL=(S,S)                                                    $4500005
 END                                                                         $4500006

SORT FIELD     01    LOCATION  0012  LENGTH  004  SEQUENCE A  FORMAT BI  FILES 1 SIZE  00002000
RECORD TYPE FIXED    LENGTH 1 0015 LENGTH 2 NONE LENGTH 3 0015
INPUT  BLOCKSIZE 00015 TYPE FIXED
INPUT  FILEA
TYPE   DISK
LABELS STD
OUTPUT BLOCKSIZE 00075 LABEL STD   TYPE DISK

OPTIONS SPECIFIED

OPTION  01 PRINT
OPTION  02 STORAGE  00065536

                    *** COMPUTED CONSTANTS ***

7D57I MAXIMUM FILE SIZE  00002410
7D58I NUMBER OF FILEW TRACKS SPECIFIED 00023
7D60I OUTPUT BLOCKS PER TRACK 26
7D61I SORT/MERGE PROGRAM ORIGIN 08192
7D65I 3615  MAXIMUM B3.
7D65I 3615  MAXIMUM B1.
```

```
EOJ DSORT
```

Figure 30. SYSLST Output for Disk Sort/Merge Sample Problems

Disk-to-Printer Utility

Program Name is Z.UTDKPR1. The disk-to-printer utility sample problem lists the numbers handled by the Assembler 2 and disk sort/merge sample problems. To execute the disk-to-printer utility sample problem, the following job control cards are needed:

```
                                                           Col. 54      Col. 72
                                                           |            |
                                                           v            v
// JOB DKPR
// PAUSE ASSGN SYS004 TO SCRATCH PACK
// ASSGN SYS004,X'cuu'
// PAUSE ASSGN SYS005 TO THE PRINTER
// ASSGN SYS005,X'cuu'
// UPSI 0000
// VOL SYS004, UIN
// DLAB 'SORTED FILE OF 2000 RECORDS FOR UTILITIES  1hhhhhh',     C
               Col. 16
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000180000,000184002,'hhhhhh',SYS004
```

Note: Replace hhhhhh in the DLAB and XTENT cards with the volume serial number of the disk pack.

```
// EXEC DKPR
   D-P CONTROL CARDS (2 cards)
/*
/&
```

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 31 :

- Job control cards
- Disk-to-printer control cards
- Utility parameter statements (10 lines)
- Numbers 0001-2000 (400 lines)
- NUMBER OF INPUT BLOCKS PROCESSED 000400
- NUMBER OF OUTPUT BLOCKS PROCESSED 000400
- END OF JOB

```
// JOB DKPR
// PAUSE // ASSGN SYS004 TO SCRATCH PACK
// ASSGN SYS004,X'291'
// PAUSE  // ASSGN SYS005 TO THE PRINTER
// ASSGN SYS005,X'00E'
// UPSI 0000
// VOL SYS004,UIN
// DLAB 'SORTED FILE OF 2000 RECORDS FOR UTILITIES   1111111',          C
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000180000,000184002,'111111',SYS004
// EXEC DKPR
```

```
// UDP TL,FF,A=(75,75),B=(120),OC,S1,PY                                 $4610001
// END                                                                  $4610002
DISK TO PRINT UTILITY
INPUT RECORD LENGTH 0075
OUTPUT RECORD LENGTH  0120
INPUT BLOCK LENGTH 00075
OUTPUT BLOCK LENGTH 00120
OUTPUT OPTION PRINT CHARACTER
2 INPUT,2 OUTPUT AREAS ASSIGNED
RECORD FORMAT FIXED
TYPE LIST
STARTING RECORD NUMBER  00000001
INPUT DEVICE TYPE  2311
```

0001	0002	0003	0004	0005
0006	0007	0008	0009	0010
0011	0012	0013	0014	0015
0016	0017	0018	0019	0020
0021	0022	0023	0024	0025
0026	0027	0028	0029	0030
0031	0032	0033	0034	0035
0036	0037	0038	0039	0040
0041	0042	0043	0044	0045
0046	0047	0048	0049	0050
0051	0052	0053	0054	0055
0056	0057	0058	0059	0060
0061	0062	0063	0064	0065
0066	0067	0068	0069	0070
0071	0072	0073	0074	0075
0076	0077	0078	0079	0080
0081	0082	0083	0084	0085
0086	0087	0088	0089	0090
0091	0092	0093	0094	0095
0096	0097	0098	0099	0100
0101	0102	0103	0104	0105
0106	0107	0108	0109	0110
0111	0112	0113	0114	0115
0116	0117	0118	0119	0120
0121	0122	0123	0124	0125
0126	0127	0128	0129	0130
0131	0132	0133	0134	0135
0136	0137	0138	0139	0140
0141	0142	0143	0144	0145
0146	0147	0148	0149	0150
0151	0152	0153	0154	0155
0156	0157	0158	0159	0160
0161	0162	0163	0164	0165
0166	0167	0168	0169	0170
0171	0172	0173	0174	0175
0176	0177	0178	0179	0180
0181	0182	0183	0184	0185
0186	0187	0188	0189	0190
0191	0192	0193	0194	0195
0196	0197	0198	0199	0200
0201	0202	0203	0204	0205
0206	0207	0208	0209	0210
0211	0212	0213	0214	0215
0216	0217	0218	0219	0220
0221	0222	0223	0224	0225
0226	0227	0228	0229	0230
0231	0232	0233	0234	0235
0236	0237	0238	0239	0240
0241	0242	0243	0244	0245
0246	0247	0248	0249	0250
0251	0252	0253	0254	0255
0256	0257	0258	0259	0260
0261	0262	0263	0264	0265
0266	0267	0268	0269	0270

Figure 31. SYSLST Output for DK-PR Sample Problems (Part 1 of 2)

```
     1866      1867      1868      1869      1870
     1871      1872      1873      1874      1875
     1876      1877      1878      1879      1880
     1881      1882      1883      1884      1885
     1886      1887      1888      1889      1890
     1891      1892      1893      1894      1895
     1896      1897      1898      1899      1900
     1901      1902      1903      1904      1905
     1906      1907      1908      1909      1910
     1911      1912      1913      1914      1915
     1916      1917      1918      1919      1920
     1921      1922      1923      1924      1925
     1926      1927      1928      1929      1930
     1931      1932      1933      1934      1935
     1936      1937      1938      1939      1940
     1941      1942      1943      1944      1945
     1946      1947      1948      1949      1950
```

```
     1951      1952      1953      1954      1955
     1956      1957      1958      1959      1960
     1961      1962      1963      1964      1965
     1966      1967      1968      1969      1970
     1971      1972      1973      1974      1975
     1976      1977      1978      1979      1980
     1981      1982      1983      1984      1985
     1986      1987      1988      1989      1990
     1991      1992      1993      1994      1995
     1996      1997      1998      1999      2000
END OF DATA
```

```
NUMBER OF INPUT BLOCKS PROCESSED  000400
NUMBER OF OUTPUT BLOCKS PROCESSED 000400
END OF JOB
```

```
EOJ DKPR
```

Figure 31. SYSLST Output for DK-PR Sample Problems (Part 2 of 2)

Assembler 3

Program Name is Z.AS3. The Assembler 3 sample problem generates numbers from 100 to 001, in descending order, and writes them on a data cell. To execute the Assembler 3 sample problem the following job control cards are needed:

```
                                                    Col. 54    Col. 72
                                                    |          |
                                                    v          v
// JOB ASSEMBLE SAMPLE 3
// OPTION LINK,LIST,LOG,NOXREF
// EXEC ASSEMBLY
   ASSEMBLER SOURCE DECK (23 cards)
/*
// EXEC LNKEDT
// PAUSE ASSGN SYS004 TO DATA CELL
// ASSGN SYS004,X'cuu'
// VOL SYS004,OUTFILE
// DLAB 'SAMPLE PROBLEM FILE OF 100 RECORDS             1hhhhhh',    C
               Col. 16
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,0,310002000,310001019,'hhhhhh',SYS004
```

Note: Replace hhhhhh in the DLAB and XTENT cards with the volume serial number of the data cell.

```
// PAUSE MOUNT CELL hhhhhh ON STATION 3
// EXEC
/&
```

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 32:

- Job control cards
- External symbol dictionary
- Source program listing
- Relocation dictionary
- Linkage editor storage map

```
// JOB ASSEMBLE SAMPLE 3
// OPTION LINK,LIST,LOG,NOXREF
// EXEC ASSEMBLY
```

```
                                     EXTERNAL SYMBOL DICTIONARY
 SYMBOL  TYPE ID  ADDR  LENGTH LD ID

          PC  01 000000 00017C
IJGF0Z2Z  ER  02
IJGF0Z2Z  SD  03 000180 00020C
```

```
        DOS ASSEMBLER TEST

   LOC  OBJECT CODE      ADDR1 ADDR2  STMT    SOURCE STATEMENT                                                   DOS CL3-0

                                         2           PRINT NOGEN                                                        $4650202
 000000                                  3           START 0                                                            $4650203
 000000 0580                             4 TESTGEN   BALR  8,0                  LOAD THE BASE REGISTER.                 $4650204
 000002                                  5           USING *,8                                                          $4650205
 000002 41D0 811E             00120      6           LA    13,SAVEAREA                                                  $4650206
                                         7           OPEN  OUTFILE              OPEN THE OUTPUT FILE.                   $4650207
 000016 4130 0064             00064     15           LA    3,100                GET STARTING RECORD VALUE.              $4650208
 00001A 4E30 8116             00118     16 NEXT      CVD   3,DWD                CONVERT THE VALUE TO DECIMAL.           $4650209
 00001E F337 2008 8116 00008 00118      17           UNPK  11(4,2),DWD          UNPACK THE VALUE.                       $4650210
 000024 96F0 200E       0000E           18           OI    14(2),X'F0'          MASK THE SIGN BIT.                      $4650211
                                        19           PUT   OUTFILE              WRITE A RECORD ON THE DATA CELL.        $4650212
 000034 4630 8018             0001A     24           BCT   3,NEXT               TEST FOR TASK COMPLETE.                 $4650213
                                        25           CLOSE OUTFILE              CLOSE THE DATA CELL FILE.               $4650214
                                        33           EOJ                        RETURN TO THE CONTROL PROGRAM.          $4650215
                                        36 OUTFILE   DTFSD BLKSIZE=23,IOAREA1=A1,IOAREA2=A2,TYPEFLE=OUTPUT,            C$4650216
                                                           IOREG=(2),RECFORM=FIXUNB,DEVICE=2321,VERIFY=YES              $4650217
                                        85           SDMODFO                                                            $4650218
 0000E8 4040404040404040               264 A1        DC    23C' '                                                       $4650219
 0000FF 4040404040404040               265 A2        DC    23C' '                                                       $4650220
 000118                                266 DWD       DS    D                                                            $4650221
 000120                                267 SAVEAREA  DS    9D                                                           $4650222
 000000                                268           END   TESTGEN                                                      $4650223
 000168 5B5BC2D6D7C5D540               269                 =C'$$BOPEN '
 000170 5B5BC2C3D3D6E2C5               270                 =C'$$BCLOSE'
 000178 00000048                       271                 =A(OUTFILE)
```

```
                                     RELOCATION DICTIONARY

 POS.ID   REL.ID   FLAGS   ADDRESS

   01       01      0C     000010
   01       01      0C     000040
   01       01      0C     000050
   01       02      18     000059
   01       01      08     000071
   01       01      0C     000074
   01       01      0C     0000A0
   01       01      0C     0000A8
   01       01      08     0000B1
   01       01      08     0000B9
   01       01      08     0000C1
   01       01      08     0000C9
   01       01      08     0000D1
   01       01      08     0000D9
   01       01      08     0000E1
   01       01      0C     000178


NO STATEMENTS FLAGGED IN THIS ASSEMBLY
```

Figure 32. SYSLST Output for Assembler (AS3) Sample Problems (Part 1 of 2)

```
// EXEC LNKEDT
```

```
JOB  ASSEMBLE             DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN    MAP
LIST   ENTRY
```

```
       PHASE  XFR-AD  LOCORE  HICORE  DSK-AD  ESD TYPE  LABEL     LOADED  REL-FR

    PHASE***  002000  002000  00238B  1E 5 2  CSECT               002000  002000

                                              CSECT     IJGFOZZZ  002180  002000
```

```
// PAUSE  // ASSGN SYS004 TO DATA CELL
// ASSGN SYS004,X'292'
// VOL SYS004,OUTFILE
// DLAB 'SAMPLE PROBLEM FILE OF 100 RECORDS         1111111',        C
             0001,66080,66080,' SAMPLE TEST '
// XTENT 1,0,310001000,310001019,'111111',SYS004
// PAUSE MOUNT CELL 111111 ON STATION 3
// EXEC
```

```
EOJ ASSEMBLE
```

Figure 32. SYSLST Output for Assembler (AS3) Sample Problems (Part 2 of 2)

Data Cell-to-Printer

Program Name is Z.UTDCPR1. The data cell-to-printer sample problem lists the numbers that the Assembler 3 sample problem wrote on the data cell. To execute the data cell-to-printer sample problem the following job control cards are needed:

```
                                                    Col. 54     Col. 72
                                                    |           |
                                                    v           v
// JOB DCPR
// PAUSE ASSGN SYS004 TO DATA CELL
// ASSGN SYS004,X'cuu'
// PAUSE ASSGN SYS005 TO THE PRINTER
// ASSGN SYS005,X'cuu'
// UPSI 0000
// VOL SYS004,UIN
// DLAB 'SAMPLE PROBLEM FILE OF 100 RECORDS          1hhhhhh',   C
               Col. 16
               0001,66080,66090,' SAMPLE TEST '
// XTENT 1,0,310001000,310001019,'hhhhhh',SYS004
```

Note: Replace hhhhhh in the DLAB and XTENT cards with the volume serial number of the data cell.

```
// EXEC DCPR
   DC-P CONTROL CARDS (2 cards)
/*
/&
```

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 33:

- Job control cards
- Data cell-to-printer control cards
- Utility parameter statements (10 lines)
- Number 0100-0001 (100 lines)
- NUMBER OF INPUT BLOCKS PROCESSED 000100
- NUMBER OF OUTPUT BLOCKS PROCESSED 000100
- END OF JOB

```
// JOB DCPR
// PAUSE  // ASSGN SYS004 TO DATA CELL
// ASSGN SYS004,X'292'
// PAUSE  // ASSGN SYS005 TO THE PRINTER
// ASSGN SYS005,X'00E'
// UPSI C000
// VOL SYS004,UIN
// DLAB 'SAMPLE PROBLEM FILE OF 100 RECORDS          1111111',          C
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,0,310001000,310001019,'111111',SYS004
// EXEC DCPR
```

```
// UMP TL,FF,A=(15,15),B=(120),S1,PY                                   $4630001
// END                                                                 $4630002
DATA CELL TO PRINT UTILITY
INPUT RECORD LENGTH 0015
OUTPUT RECORD LENGTH  0120
INPUT BLOCK LENGTH 00015
OUTPUT BLOCK LENGTH 00120
OUTPUT OPTION PRINT CHARACTER
2 INPUT,2 OUTPUT AREAS ASSIGNED
RECORD FORMAT FIXED
TYPE LIST
STARTING RECORD NUMBER  00000001
INPUT DEVICE TYPE  2321
```

```
            0100
            0099
            0098
            0097
            0096
            0095
            0094
            0093
            0092
            0091
            0090
            0089
            0088
            0087
            0086
            0085
            0084
            0083
            0082
            0081
            0080
            0079
            0078
            0077
            0076
            0075
            0074
            0073
            0072
            0071
            0070
            0069
            0068
            0067
            0066
            0065
            0064
            0063
            0062
            0061
```

```
```

Figure 33. SYSLST Output for DC-PR Sample Problems (Part 1 of 2)

```
            0015
            0014
            0013
            0012
            0011
            0010
            0009
            0008
            0007
            0006
            0005
            0004
            0003
            0002
            0001
    END OF DATA
```

```
    NUMBER OF INPUT BLOCKS PROCESSED  000100
    NUMBER OF OUTPUT BLOCKS PROCESSED 000100
    END OF JOB
```

```
    EOJ DCPR
```

Figure 33. SYSLST Output for DC-PR Sample Problems (Part 2 of 2)

Assembler 4

Program Name is Z.AS4. The Assembler 4 sample problem generates numbers from 2000 to 0001 in descending order and writes them on tape as 15 character unblocked records (11 blanks followed by 4 zoned digits, unsigned). To execute the Assembler 4 sample problem the following job control cards are needed:

```
// JOB ASSEMBLY SAMPLE 4
// OPTION LINK,LIST,LOG,NOXREF
// EXEC ASSEMBLY
   Assembler source deck (23 cards)
/*
// EXEC LNKEDT
// PAUSE ASSGN SYS002 TO A 9 TRACK TAPE DRIVE
// ASSGN SYS002,X'cuu'
// MTC REW,SYS002
// EXEC
/&
```

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 34:

- Job control cards
- External symbol dictionary
- Source program listing
- Relocation dictionary
- Linkage editor storage map

```
// JOB ASSEMBLY SAMPLE 4
// OPTION LINK,LIST,LOG,NOXREF
// EXEC ASSEMBLY
```

```
                                        EXTERNAL SYMBOL DICTIONARY                                   PAGE    1
 SYMBOL   TYPE ID  ADDR   LENGTH LD ID

          PC   01 000000 00012C
IJFFZZZZ  ER   02
IJFFZZZZ  SD   03 00013C 0002D2
```

```
SMPL    DOS ASSEMBLER TEST                                                                              PAGE    1

  LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                    DOS CL3-1 01/01/01

                                        2          PRINT NOGEN                                            $4650402
000000                                  3          START 0                                                $4650403
000000 0580                             4 TESTGEN  BALR  8,0                   LOAD BASE REG              $4650404
000002                                  5          USING *,8                                              $4650405
000002 4100 80CE              000D0     6          LA    13,SAVEAREA                                      $4650406
                                        7          OPEN  OUTFILE               OPEN OUTPUT FILE           $4650407
000016 4130 07D0              007D0    15          LA    3,2000                GET STARTING RECORD VALUE  $4650408
00001A 4E30 80C6              000C8    16 NEXT     CVD   3,DWD                 CONVERT VALUE TO DEC.      $4650409
00001E F337 200B 80C6 0000B 000C8      17          UNPK  11(4,2),DWD           UNPACK IT                  $4650410
000024 96F0 200E        0000E          18          OI    14(2),X'F0'           MASK SIGN BIT              $4650411
                                       19          PUT   OUTFILE               WRITE RECORD ON TAPE       $4650412
000034 4630 8018              0001A    24          BCT   3,NEXT                TEST FOR TASK COMPLETE     $4650413
                                       25          CLOSE OUTFILE               CLOSE TAPE FILE            $4650414
                                       33          EOJ                         RETURN TO CONTROL PROGRAM  $4650415
                                       36 OUTFILE  DTFMT BLKSIZE=18,DEVADDR=SYS002,FILABL=NO,IOAREA1=A1, X$4650416
                                                         IOAREA2=A2,IOREG=(2),TYPEFLE=OUTPUT              $4650417
                                       67          MTMOD RECFORM=FIXUNB                                   $4650418
00009E 404040404040404040             295 A1       DC    18C' '                                           $4650419
0000B0 404040404040404040             296 A2       DC    18C' '                                           $4650420
0000C8                                297 DWD      DS    D                                                $4650421
0000D0                                298 SAVEAREA DS    9D                                               $4650422
000000                                299          END   TESTGEN                                          $4650423
000118 5B5BC2D6D7C5D540               300                =C'$$BOPEN '
000120 5B5BC2C3D3D6E2C5               301                =C'$$BCLOSE'
000128 0C000048                       302                =A(OUTFILE)
```

```
                                        RELOCATION DICTIONARY                                        PAGE    1

 POS.ID   REL.ID   FLAGS   ADDRESS

   01       01      0C     000010
   01       01      0C     000040
   01       01      0C     000050
   01       02      18     000059
   01       01      08     00006D
   01       01      08     000081
   01       01      0C     000088
   01       01      0C     00008C
   01       01      0C     000094
   01       01      0C     000128


NO STATEMENTS FLAGGED IN THIS ASSEMBLY
```

```
// EXEC LNKEDT
```

Figure 34. SYSLST Output for Assembler (AS4) Sample Problems (Part 1 of 2)

```
JOB  ASSEMBLY            DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN     MAP
LIST   ENTRY
```

```
01/01/01    PHASE  XFR-AD  LOCORE  HICORE  DSK-AD   ESD TYPE  LABEL     LOADED  REL-FR

        PHASE***  002800  002800  002C01  28 06 2  CSECT               002800  002800

                                                    CSECT     IJFFZZZZ  002930  002800
```

```
// PAUSE ASSGN SYS002 TO A 9 TRACK TAPE DRIVE
// ASSGN SYS002,X'282'
// MTC REW,SYS002
// EXEC
```

```
EOJ ASSEMBLY
```

Figure 34. SYSLST Output for Assembler (AS4) Sample Problems (Part 2 of 2)

Tape and Disk Sort/Merge -- 2400 Application

Program Name is Z.SM4. The Tape and Disk Sort/Merge sample problem rearranges in ascending sequence the records produced by the Assembler 4 program and writes them on another tape, five records per block. To execute the Tape and Disk Sort/Merge sample problem the following job control cards are needed:

```
// JOB SORT 2400
// PAUSE ASSGN SYS002 TO SAME DRIVE AS PREVIOUS JOB
// PAUSE ASSGN SYS001,SYS003,SYS004,SYS005 TO DIFF TAPES
// ASSGN SYS001,X'cuu'
// ASSGN SYS002,X'cuu'
// ASSGN SYS003,X'cuu'
// ASSGN SYS004,X'cuu'
// ASSGN SYS005,X'cuu'
// TLBL SORTOUT,,64/001
// TLBL SORTIN1,,64/001
// TLBL SORTWK1,,64/001
// TLBL SORTWK2,,64/001
// TLBL SORTWK3,,64/001
// LBLTYP TAPE
// EXEC SORT
   Sort/merge control cards (6 cards)
/&
```

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 35.

- Job control cards
- Sort control cards
- Sort parameters
- Computed constants

```
// JOB SORT 2400
// PAUSE ASSGN SYS002 TO SAME DRIVE AS PREVIOUS JOB
// PAUSE ASSGN SYS001,SYS003,SYS004,SYS005 TO DIFF TAPES
// ASSGN SYS001,X'286'
// ASSGN SYS002,X'282'
// ASSGN SYS003,X'287'
// ASSGN SYS004,X'284'
// ASSGN SYS005,X'285'
// TLBL SORTOUT,,64/001
// TLBL SORTIN1,,64/001
// TLBL SORTWK1,,64/001
// TLBL SORTWK2,,64/001
// TLBL SORTWK3,,64/001
// LBLTYP TAPE
// EXEC SORT

7000I  SORT FORMAT=BI,FIELDS=(12,4,A),WORK=3                        $4830001
7000I  RECORD TYPE=F,LENGTH=18                                      $4830002
7000I  OPTION PRINT=ALL,LABEL=(U,U,U)                               $4830003
7000I  INPFIL VOLUME=1,BLKSIZE=18                                   $4830004
7000I  OUTFIL BLKSIZE=90                                            $4830005
7000I  END                                                          $4830006
7050I NMAX=001144876
7051I B=    000009991
7052I G=    000006041
7001I PHASE 0 END,NO DETECTED ERRORS
```

```
EOJ SORT
```

Figure 35. SYSLST Output for Tape and Disk Sort/Merge Sample Problem

Tape-to-Printer Utility

Program Name is Z.UTTPPR2. The Tape-to-Printer Utility sample problem lists the numbers handled by the Assembler 4 and tape and disk sort/merge (2400 application) sample programs. To execute the tape to printer utility sample problem, the following job control cards are needed:

```
// JOB TPPR 2400 TO PRINTER
// PAUSE ASSGN SYS004 TO SYS001 OF PREVIOUS JOB
// ASSGN SYS004,X'cuu'
// PAUSE ASSGN SYS005 TO THE PRINTER
// ASSGN SYS005,X'cuu'
// UPSI 1000
// EXEC TPPR
   Tape-to-printer utility control cards (2 cards)
/*
/&
```

The user must reply IT IS EOF to the logged message by entering Ⓑ (alternate code 5) from the IBM 1052 printer-keyboard.

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 36:

- Job control cards
- Tape-to-printer control cards
- Utility parameter statements (11 lines)
- Numbers 0001-2000 (400 lines)
- IS IT EOF
- REPLY Y (Ⓑ is not printed)
- END OF DATA
- NUMBER OF INPUT BLOCKS PROCESSED 000400
- NUMBER OF OUTPUT BLOCKS PROCESSED 000400
- END OF JOB

```
// JOB TPPR 2400 TO PRINTER
// PAUSE ASSGN SYS004 TO SYS001 OF PREVEOUS JOB
// ASSGN SYS004,X'286'
// PAUSE ASSGN SYS005 TO THE PRINTER
// ASSGN SYS005,X'00E'
// UPSI 1000
// EXEC TPPR
```

```
// UTP TL,FF,A=(90,90),B=(144),IR,OC,S1,PY                              $4630101
// END                                                                  $4630102
TAPE TO PRINT UTILITY
INPUT RECORD LENGTH 0090
OUTPUT RECORD LENGTH  0144
INPUT BLOCK LENGTH 00090
OUTPUT BLOCK LENGTH 00144
INPUT OPTION REWIND
OUTPUT OPTION PRINT CHARACTER
2 INPUT,2 OUTPUT AREAS ASSIGNED
RECORD FORMAT FIXED
TYPE LIST
STARTING RECORD NUMBER  00000001
```

```
        0001        0002        0003        0004        0005
        0006        0007        0008        0009        0010
        0011        0012        0013        0014        0015
        0016        0017        0018        0019        0020
        0021        0022        0023        0024        0025
        0026        0027        0028        0029        0030
        0031        0032        0033        0034        0035
        0036        0037        0038        0039        0040
        0041        0042        0043        0044        0045
        0046        0047        0048        0049        0050
        0051        0052        0053        0054        0055
        0056        0057        0058        0059        0060
        0061        0062        0063        0064        0065
        0066        0067        0068        0069        0070
        0071        0072        0073        0074        0075
        0076        0077        0078        0079        0080
        0081        0082        0083        0084        0085
        0086        0087        0088        0089        0090
        0091        0092        0093        0094        0095
        0096        0097        0098        0099        0100
        0101        0102        0103        0104        0105
        0106        0107        0108        0109        0110
        0111        0112        0113        0114        0115
        0116        0117        0118        0119        0120
        0121        0122        0123        0124        0125
        0126        0127        0128        0129        0130
        0131        0132        0133        0134        0135
        0136        0137        0138        0139        0140
        0141        0142        0143        0144        0145
        0146        0147        0148        0149        0150
        0151        0152        0153        0154        0155
        0156        0157        0158        0159        0160
        0161        0162        0163        0164        0165
        0166        0167        0168        0169        0170
        0171        0172        0173        0174        0175
        0176        0177        0178        0179        0180
        0181        0182        0183        0184        0185
        0186        0187        0188        0189        0190
        0191        0192        0193        0194        0195
        0196        0197        0198        0199        0200
        0201        0202        0203        0204        0205
        0206        0207        0208        0209        0210
        0211        0212        0213        0214        0215
        0216        0217        0218        0219        0220
        0221        0222        0223        0224        0225
        0226        0227        0228        0229        0230
        0231        0232        0233        0234        0235
        0236        0237        0238        0239        0240
        0241        0242        0243        0244        0245
        0246        0247        0248        0249        0250
        0251        0252        0253        0254        0255
        0256        0257        0258        0259        0260
        0261        0262        0263        0264        0265
        0266        0267        0268        0269        0270
        0271        0272        0273        0274        0275
        0276        0277        0278        0279        0280
PAGE  1
```

Figure 36. SYSLST Output for Tape-to-Printer Sample Problems (Part 1 of 4)

```
      0281       0282       0283       0284       0285
      0286       0287       0288       0289       0290
      0291       0292       0293       0294       0295
      0296       0297       0298       0299       0300
```

```
      0536       0537       0538       0539       0540
      0541       0542       0543       0544       0545
      0546       0547       0548       0549       0550
      0551       0552       0553       0554       0555
      0556       0557       0558       0559       0560
PAGE  2
```

```
      0561       0562       0563       0564       0565
      0566       0567       0568       0569       0570
      0571       0572       0573       0574       0575
```

```
      0816       0817       0818       0819       0820
      0821       0822       0823       0824       0825
      0826       0827       0828       0829       0830
      0831       0832       0833       0834       0835
      0836       0837       0838       0839       0840
PAGE  3
```

```
      0841       0842       0843       0844       0845
      0846       0847       0848       0849       0850
      0851       0852       0853       0854       0855
```

```
      1101       1102       1103       1104       1105
      1106       1107       1108       1109       1110
      1111       1112       1113       1114       1115
      1116       1117       1118       1119       1120
PAGE  4
```

Figure 36. SYSLST Output for Tape-to-Printer Sample Problems (Part 2 of 4)

```
   1121        1122        1123        1124        1125
   1126        1127        1128        1129        1130
   1131        1132        1133        1134        1135
   1136        1137        1138        1139        1140
```

```
   1371        1372        1373        1374        1375
   1376        1377        1378        1379        1380
   1381        1382        1383        1384        1385
   1386        1387        1388        1389        1390
   1391        1392        1393        1394        1395
   1396        1397        1398        1399        1400
PAGE  5
```

```
   1401        1402        1403        1404        1405
   1406        1407        1408        1409        1410
   1411        1412        1413        1414        1415
   1416        1417        1418        1419        1420
```

```
   1656        1657        1658        1659        1660
   1661        1662        1663        1664        1665
   1666        1667        1668        1669        1670
   1671        1672        1673        1674        1675
   1676        1677        1678        1679        1680
PAGE  6
```

```
   1681        1682        1683        1684        1685
   1686        1687        1688        1689        1690
   1691        1692        1693        1694        1695
   1696        1697        1698        1699        1700
   1701        1702        1703        1704        1705
```

```
   1946        1947        1948        1949        1950
   1951        1952        1953        1954        1955
   1956        1957        1958        1959        1960
PAGE  7
```

Figure 36. SYSLST Output for Tape-to-Printer Sample Problems (Part 3 of 4)

```
     1961        1962        1963        1964        1965
     1966        1967        1968        1969        1970
     1971        1972        1973        1974        1975
     1976        1977        1978        1979        1980
     1981        1982        1983        1984        1985
     1986        1987        1988        1989        1990
     1991        1992        1993        1994        1995
     1996        1997        1998        1999        2000

8C01D IS IT EOF
REPLY Y
 END OF DATA

PAGE  8
```

```
NUMBER OF INPUT BLOCKS PROCESSED  000400
NUMBER OF OUTPUT BLOCKS PROCESSED 000400
END OF JOB
```

```
EOJ TPPR
```

Figure 36. SYSLST Output for Tape-to-Printer Sample Problem (Part 4 of 4)

Assembler 5

Program Name is Z.AS5. The Assembler 5 sample problem generates numbers from 2000 to 0001 in descending order and writes them to a 2311 disk storage device. To execute the Assembler 5 sample problem, the following job control cards are needed:

```
                                                  Col. 54    Col. 72
                                                  |          |
                                                  v          v
// JOB ASSEMBLY SAMPLE 5
// OPTION LINK,LIST,LOG,NOXREF
// EXEC ASSEMBLY
         Assembler source deck (23 cards)
/*
// EXEC LNKEDT
// PAUSE ASSGN SYS002 TO SCRATCH PACK
// ASSGN SYS002,X'cuu'
// VOL SYS002,OUTFILE
// DLAB 'SAMPLE PROBLEM FILE OF 2000 RECORDS          1hhhhhh',    X
           Col. 16
           0001,66060,66080,' SAMPLE TEST '
// XTENT 1,000,000180000,000184002,'hhhhhh',SYS002
// EXEC
/&
```

Note: Replace hhhhhh in the DLAB and XTENT cards with the volume serial number of the disk pack.

SYSLOG output is shown in Figure 17.

Following is a summary of the SYSLST output shown in Figure 37:

- Job control cards
- External symbol dictionary
- Source program listing
- Relocation dictionary
- Linkage editor storage map

```
// JOB ASSEMBLY SAMPLE 5
// OPTION LINK,LIST,LOG,NOXREF
// EXEC ASSEMBLY
```

```
                                        EXTERNAL SYMBOL DICTIONARY                                     PAGE    1
 SYMBOL   TYPE ID  ADDR   LENGTH LD ID

          PC   01 000000 00017C
IJGFOZZZ  ER   02
IJGFOZZZ  SD   03 000180 000202
```

```
      DOS ASSEMBLER TEST                                                                                  PAGE    1

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          DOS CL3-1 08/10/68

                                       2           PRINT NOGEN                                                $4650502
000000                                 3           START 0                                                    $4650503
000000 0580                            4 TESTGEN   BALR  8,0                   LOAD THE BASE REGISTER.        $4650504
000002                                 5           USING *,8                                                  $4650505
000002 41D0 811E             00120     6           LA    13,SAVEAREA                                          $4650506
                                       7           OPEN  OUTFILE               OPEN THE OUTPUT FILE.          $4650507
000016 4130 07D0             007D0    15           LA    3,2000                GET STARTING RECORD VALUE.     $4650508
00001A 4E30 8116             00118    16 NEXT      CVD   3,DWD                 CONVERT THE VALUE TO DECIMAL.  $4650509
00001E F337 2008 8116 0000B  00118    17           UNPK  11(4,2),DWD           UNPACK THE VALUE.              $4650510
000024 96F0 200E       0000E          18           OI    14(2),X'F0'           MASK THE SIGN BIT.             $4650511
                                      19           PUT   OUTFILE               WRITE A RECORD ON DISK.        $4650512
000034 4630 8018             0001A    24           BCT   3,NEXT                TEST FOR TASK COMPLETE.        $4650513
                                      25           CLOSE OUTFILE               CLOSE THE DISK FILE.           $4650514
                                      33           EOJ                         RETURN TO THE CONTROL PROGRAM. $4650515
                                      36 OUTFILE   DTFSD BLKSIZE=23,IOAREA1=A1,IOAREA2=A2,TYPEFLE=OUTPUT,    C$4650516
                                                         IOREG=(2),RECFORM=FIXUNB,DEVICE=2311                 $4650517
                                      86           SDMODFO                                                    $4650518
0000E8 4040404040404040              263 A1        DC    23C' '                                               $4650519
0000FF 4040404040404040              264 A2        DC    23C' '                                               $4650520
000118                               265 DWD       DS    D                                                    $4650521
000120                               266 SAVEAREA  DS    9D                                                   $4650522
000000                               267           END   TESTGEN                                              $4650523
000168 5B5BC2D6D7C5D540              268                 =C'$$BOPEN '
000170 5B5BC2C3D3D6E2C5              269                 =C'$$BCLOSE'
000178 00000048                      270                 =A(OUTFILE)
```

```
                                           RELOCATION DICTIONARY                                       PAGE    1

 POS.ID    REL.ID    FLAGS    ADDRESS

   01        01       0C      000010
   01        01       0C      000040
   01        01       0C      000050
   01        02       18      000059
   01        01       08      000071
   01        01       0C      000074
   01        01       0C      0000A0
   01        01       0C      0000A8
   01        01       08      0000B1
   01        01       08      0000B9
   01        01       08      0000C1
   01        01       08      0000C9
   01        01       08      0000D1
   01        01       08      0000D9
   01        01       08      0000E1
   01        01       0C      000178


NO STATEMENTS FLAGGED IN THIS ASSEMBLY
```

Figure 37. SYSLST Output for Assembler (AS5) Sample Problems (Part 1 of 2)

```
// EXEC LNKEDT
```

```
JOB  ASSEMBLY  08/10/68   DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN     MAP
LIST   ENTRY
```

```
C8/1C/68   PHASE   XFR-AD  LOCORE  HICORE  DSK-AD   ESD TYPE  LABEL     LOADED  REL-FR

        PHASE***  0C2800  0C28C0  002881  2B 06 2  CSECT               002800  0028C0

                                                   CSECT     IJGFOZZZ  002980  002800
```

```
// PAUSE ASSGN SYS0C2 TO SCRATCH PACK
// ASSGN SYS0C2,X'191'
// VOL SYS002,OUTFILE
// DLAB 'SAMPLE PROBLEM FILE OF 2000 RECORDS        1191191',        X
               0001,66060,66080,' SAMPLE TEST '
// XTENT 1,000,000180000,000184002,'191191',SYS002
// EXEC
```

```
EOJ ASSEMBLY
```

Figure 37. SYSLST Output for Assembler (AS5) Sample Problems (Part 2 of 2)

Tape and Disk Sort/Merge -- 2311 Application

Program Name is Z.SM5. The tape and disk sort/merge sample problem rearranges in ascending sequence the records produced by the Assembler 5 program and writes them back to a 2311 disk storage device. To execute the tape and disk sort/merge sample problem the following job control cards are needed:

```
                                                       Col. 54      Col. 72
// JOB SORT 2311
// PAUSE ASSGN SYS001,SYS002,SYS003 TO SCRATCH PACK
// ASSGN SYS001,X'cuu'
// ASSGN SYS002,X'cuu'
// ASSGN SYS003,X'cuu'
// VOL SYS001,SORTOUT
// DLAB 'SORTED FILE OF 2000 RECORDS FOR UTILITIES  1hhhhhh',    X
               Col. 16
               0001,66080,66080' SAMPLE TEST '
// XTENT 1,000,000171000,000174002,'hhhhhh',SYS001
// VOL SYS002,SORTIN1
// DLAB 'SAMPLE PROBLEM FILE OF 2000 RECORDS        1hhhhhh',    X
               Col. 16
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000180000,000184002,'hhhhhh',SYS002
// VOL SYS003,SORTWK1
// DLAB 'SAMPLE PROGRAM WORK AREA FOR SORT RUN      1hhhhhh',    X
               Col. 16
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000184003,000187005,'hhhhhh',SYS003
// EXEC SORT
   Sort/merge control cards (6 cards)
/&
```

Note: Replace hhhhhh in the DLAB and XTENT cards with the volume serial number of the disk pack.

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 38:

- Job control cards
- Sort control cards
- Sort parameters
- Computed constants

```
// JOB SORT 2311
// PAUSE ASSGN SYS001,SYS002,SYS003 TO SCRATCH PACK
// ASSGN SYS001,X'191'
// ASSGN SYS002,X'191'
// ASSGN SYS003,X'191'
// VOL SYS001,SORTOUT
// DLAB 'SORTED FILE OF 2000 RECORDS FOR UTILITIES   1191191',          X
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000171000,000174002,'191191',SYS001
// VOL SYS002,SORTIN1
// DLAB 'SAMPLE PROBLEM FILE OF 2000 RECORDS         1191191',          X
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000180000,000184002,'191191',SYS002
// VOL SYS003,SORTWK1
// DLAB 'SAMPLE PROGRAM WORK AREA FOR SORT RUN       1191191',          X
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000184003,000187005,'191191',SYS003
// EXEC SORT

7000I  SORT FIELDS=(11,4,A),FORMAT=BI,SIZE=2000,WORK=1                 $4830101
7000I  RECORD TYPE=F,LENGTH=(15,,15)                                   $4830102
7000I  INPFIL BLKSIZE=75                                               $4830103
7000I  OUTFIL BLKSIZE=75                                               $4830104
7000I  OPTION PRINT=ALL,LABEL=(S,S),STORAGE=10240                      $4830105
7000I  END                                                             $4830106
7050I NMAX = 00003432
7051I B = 00000828
7052I G = 00000112
7001I PHASE 0 END,NO DETECTED ERRORS
```

```
EOJ SORT
```

Figure 38. SYSLST Output for Tape and Disk Sort/Merge Sample Problem

Disk-to-Printer Utility

Program Name is Z.UTDKPR2. The disk-to-printer utility sample problem lists the numbers handled by the Assembler 5 and tape and disk sort/merge (2311 application) sample problems. To execute the disk-to-printer utility sample problem, the following job control cards are needed:

```
                                                          Col. 54      Col. 72
                                                          |            |
                                                          v            v
// JOB DKPR 2311 TO PRINTER
// PAUSE ASSGN SYS004 TO SCRATCH PACK
// ASSGN SYS004,X'cuu'
// PAUSE ASSGN SYS005 TO THE PRINTER
// ASSGN SYS005,X'cuu'
// UPSI 000
// VOL SYS004, UIN
// DLAB 'SORTED FILE OF 2000 RECORDS FOR UTILITIES  1hhhhhh',      X
            Col. 16
            0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000171000,000174002,'hhhhhh',SYS004
// EXEC DKPR
   Disk-to-printer utility control cards (2 cards)
/*
```

Note: Replace hhhhhh in the DLAB and XTENT cards with the volume serial number of the disk pack.

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 39:

- Job control cards
- Utility parameter statements (10 lines)
- Numbers 0001-2000 (400 lines)
- NUMBER OF INPUT BLOCKS PROCESSED 000400
- NUMBER OF OUTPUT BLOCKS PROCESSED 000400
- END OF JOB

```
// JOB DKPR 2311 TO PRINTER
// PAUSE ASSGN SYS004 TO SCRATCH PACK
// ASSGN SYS004,X'191'
// PAUSE ASSGN SYS005 TO THE PRINTER
// ASSGN SYS005,X'00E'
// UPSI 000
// VOL SYS004, UIN
// DLAB 'SORTED FILE OF 2000 RECORDS FOR UTILITIES   1191191',          X
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000171000,000174002,'191191',SYS004
// EXEC DKPR
```

```
// UDP TL,FF,A=(75,75),B=(120),OC,S1,PY                           $4610101
// END                                                            $4610102
DISK TO PRINT UTILITY
INPUT RECORD LENGTH 0075
OUTPUT RECORD LENGTH  0120
INPUT BLOCK LENGTH 00075
OUTPUT BLOCK LENGTH 00120
OUTPUT OPTION PRINT CHARACTER
2 INPUT,2 OUTPUT AREAS ASSIGNED
RECORD FORMAT FIXED
TYPE LIST
STARTING RECORD NUMBER  00000001
INPUT DEVICE TYPE  2311
```

```
            0001        0002        0003        0004        0005
            0006        0007        0008        0009        0010
            0011        0012        0013        0014        0015
            0016        0017        0018        0019        0020
            0261        0262        0263        0264        0265
            0266        0267        0268        0269        0270
 PAGE  1
```

```
            0271        0272        0273        0274        0275
            0276        0277        [illegible] [illegible] 0280
            0281        0282        [illegible] [illegible] 0285
            0286        0287        [illegible] [illegible] 0290
            0536        [illegible] [illegible] [illegible] 0540
            [illegible] [illegible] [illegible] [illegible] 0545
            0546        [illegible] [illegible] [illegible] 0550
 PAGE  2
```

Figure 39. SYSIST Output for Disk-to-Printer Sample Problems (UTDKPR2) (Part 1 of 3)

```
          0551          0552          0553          0554          0555
          0556          0557          0558          0559          0568
          0567          0566          0565          0564          0563
          0562          0561          0560          0569          0570
          0571          0572          0573          0574          0575
          0576          0577          0578          0579          0580
```

```
          0811          0812          0813          0814          0815
          0816          0817          0818          0819          0820
          0821          0822          0823          0824          0825
          0826          0827          0828          0829          0830
PAGE   3
```

```
          0831          0832          0833          0834          0835
          0836          0837          0838          0839          0840
          0841          0842          0843          0844          0845
          0846          0847          0848          0849          0850
          0851          0852          0853          0854          0855
          0856          0857          0858          0859          0860
```

```
          1096          1097          1098          1099          1100
          1101          1102          1103          1104          1105
          1106          1107          1108          1109          1110
PAGE   4
```

```
          1111          1112          1113          1114          1115
          1116          1117          1118          1119          1120
          1121          1122          1123          1124          1125
          1126          1127          1128          1129          1130
```

```
          1366          1367          1368          1369          1370
          1371          1372          1373          1374          1375
          1376          1377          1378          1379          1380
          1381          1382          1383          1384          1385
          1386          1387          1388          1389          1390
PAGE   5
```

Figure 39. SYSLST Output for Disk-to-Printer Sample Problems (UTDKPR2) (Part 2 of 3)

```
         1391          1392          1393          1394          1395
         1396          1397          1398          1399          1408
         1407          1406          1405          1404          1403
         1402          1401          1400          1409          1410
         1411          1412          1413          1414          1415
```

```
         1651          1652          1653          1654          1655
         1656          1657          1658          1659          1660
         1661          1662          1663          1664          1665
         1666          1667          1668          1669          1670
 PAGE  6
```

```
         1671          1672          1673          1674          1675
         1676          1677          1678          1679          1680
         1681          1682          1683          1684          1685
         1686          1687          1688          1689          1690
```

```
         1931          1932
         1936          1937          1938          1939          1940
         1941          1942          1943          1944          1945
         1946          1947          1948          1949          1950
 PAGE  7
```

```
         1951          1952          1953          1954          1955
         1956          1957          1958          1959          1960
         1961          1962          1963          1964          1965
         1966          1967          1968          1969          1970
         1971          1972          1973          1974          1975
         1976          1977          1978          1979          1980
         1981          1982          1983          1984          1985
         1986          1987          1988          1989          1990
         1991          1992          1993          1994          1995
         1996          1997          1998          1999          2000
  END OF DATA
 PAGE  8
```

```
 NUMBER OF INPUT BLOCKS PROCESSED  000400
 NUMBER OF OUTPUT BLOCKS PROCESSED 000400
 END OF JOB
```

```
 EOJ DKPR
```

Figure 39. SYSLST Output for Disk-to-Printer Sample Problems (UTDKPR2) (Part 3 of 3)

Assembler 6

Program Name is Z.AS6. The Assembler 6 sample problem generates numbers from 3000 to 0001 in descending order and writes them to a 2314 direct access storage facility. To execute the Assembler 6 sample problem, the following job control cards are needed:

```
                                                      Col. 54       Col. 72
// JOB ASSEMBLY SAMPLE 6
// OPTION LINK,LIST,LOG,NOXREF
// EXEC ASSEMBLY
   Assembler source deck (23 cards)
/*
// EXEC LNKEDT
// PAUSE ASSGN SYS002 TO SCRATCH PACK
// ASSGN SYS002,X'cuu'
// VOL SYS002,OUTFILE
// DLAB 'SAMPLE PROBLEM FILE OF 2000 RECORDS          1hhhhhh',     X
               Col. 16
               0001,66060,66080,' SAMPLE TEST '
// XTENT 1,000,000180000,000184002,'hhhhhh',SYS002
// EXEC
/&
```

Note: Replace hhhhhh in the DLAB and XTENT cards with the volume serial number of the disk pack.

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 40:

- Job control cards
- External symbol dictionary
- Source program listing
- Relocation dictionary
- Linkage editor storage map

```
// JOB ASSEMBLY SAMPLE 6
// OPTION LINK,LIST,LOG,NOXREF
// EXEC ASSEMBLY
```

```
                                   EXTERNAL SYMBOL DICTIONARY                                    PAGE    1
 SYMBOL   TYPE ID  ADDR   LENGTH LD ID

           PC  01 000000 00017C
IJGFOZZZ   ER  02
IJGFOZZZ   SD  03 000180 000202
```

```
      DOS ASSEMBLER TEST                                                                           PAGE    1

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  DOS CL3-1 08/10/68

                                       2          PRINT NOGEN                                        $4650602
000000                                 3          START 0                                            $4650603
000000 0580                            4 TESTGEN  BALR  8,0              LOAD THE BASE REGISTER.     $4650604
000002                                 5          USING *,8                                          $4650605
000002 41D0 811E              00120    6          LA    13,SAVEAREA                                  $4650606
                                       7          OPEN  OUTFILE          OPEN THE OUTPUT FILE.       $4650607
000016 4130 0BB8              00BB8   15          LA    3,3000           GET STARTING RECORD VALUE.  $4650608
00001A 4E30 8116              00118   16 NEXT     CVD   3,DWD            CONVERT THE VALUE TO DECIMAL. $4650609
00001E F337 200B 8116 0000B 00118     17          UNPK  11(4,2),DWD      UNPACK THE VALUE.           $4650610
000024 96F0 200E        0000E         18          OI    14(2),X'F0'      MASK THE SIGN BIT.          $4650611
                                      19          PUT   OUTFILE          WRITE A RECORD ON DISK.     $4650612
000034 4630 8018              0001A   24          BCT   3,NEXT           TEST FOR TASK COMPLETE.     $4650613
                                      25          CLOSE OUTFILE          CLOSE THE DISK FILE.        $4650614
                                      33          EOJ                    RETURN TO THE CONTROL PROGRAM. $4650615
                                      36 OUTFILE  DTFSD BLKSIZE=23,IOAREA1=A1,IOAREA2=A2,TYPEFLE=OUTPUT, C$4650616
                                                        IOREG=(2),RECFORM=FIXUNB,DEVICE=2314          $4650617
                                      86          SDMODFO                                            $4650618
0000E8 4040404040404040              263 A1       DC    23C' '                                       $4650619
0000FF 4040404040404040              264 A2       DC    23C' '                                       $4650620
000118                               265 DWD      DS    D                                            $4650621
000120                               266 SAVEAREA DS    9D                                           $4650622
000000                               267          END   TESTGEN                                      $4650623
000168 5B5BC2D6D7C5D540              268                =C'$$BOPEN '
000170 5B5BC2C3D3D6E2C5              269                =C'$$BCLOSE'
000178 00000048                      270                =A(OUTFILE)
```

```
                                   RELOCATION DICTIONARY                                         PAGE    1

 POS.ID   REL.ID   FLAGS    ADDRESS

   01       01      0C      000010
   01       01      0C      000040
   01       01      0C      000050
   01       02      18      000059
   01       01      08      000071
   01       01      0C      000074
   01       01      0C      0000A0
   01       01      0C      0000A8
   01       01      08      0000B1
   01       01      08      0000B9
   01       01      08      0000C1
   01       01      08      0000C9
   01       01      08      0000D1
   01       01      08      0000D9
   01       01      08      0000E1
   01       01      0C      000178


NO STATEMENTS FLAGGED IN THIS ASSEMBLY
```

Figure 40. SYSIST Output for Assembler (AS6) Sample Problems (Part 1 of 2)

```
// EXEC LNKEDT
```

```
JOB  ASSEMBLY  08/10/68   DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN     MAP
LIST   ENTRY
```

```
08/10/68    PHASE   XFR-AD  LOCORE  HICORE  DSK-AD   ESD TYPE  LABEL     LOADED  REL-FR

        PHASE***   002800  002800  002B81  28 06 2  CSECT               002800  002800

                                                    CSECT     IJGFOZZZ  002980  002800
```

```
// PAUSE ASSGN SYS002 TO SCRATCH PACK
// ASSGN SYS002,X'230'
// VOL SYS002,OUTFILE
// DLAB 'SAMPLE PROBLEM FILE OF 3000 RECORDS         1X30X30',         X
               0001,66060,66080,' SAMPLE TEST '
// XTENT 1,000,000180000,000184002,'X30X30',SYS002
// EXEC
```

```
EOJ ASSEMBLY
```

Figure 40. SYSLST Output for Assembler (AS6) Sample Problems (Part 2 of 2)

Tape and Disk Sort/Merge -- 2314 Application

Program Name is Z.SM6. The tape and disk sort/merge sample problem rearranges the records produced by the Assembler 6 program in ascending sequence and writes them back to a 2314 direct access storage facility. To execute the tape and disk sort/merge sample problem the following job control cards are needed:

```
                                                     Col. 54      Col. 72
// JOB SORT 2314
// PAUSE ASSGN SYS001,SYS002,SYS003 TO SCRATCH PACK
// ASSGN SYS001,S'cuu'
// ASSGN SYS002,X'cuu'
// ASSGN SYS003,X'cuu'
// VOI SYS001,SORTOUT
// DLAB 'SORTED FILE OF 2000 RECORDS FOR UTILITIES  1hhhhhh',    X
               Col. 16
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000171000,000174002,'hhhhhh',SYS001
// VOL SYS002,SORTIN1
// DLAB 'SAMPLE PROBLEM FILE OF 2000 RECORDS        1hhhhhh',    X
               Col. 16
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000180000,000184002,'hhhhhh',SYS002
// VOL SYS003,SORTWK1
// DLAB 'SAMPLE PROGRAM WORK AREA FOR SORT RUN      1hhhhhh',    X
               Col. 16
               0001,66080,66080,' SAMPLE TEST ,
// XTENT 1,000,000184003,000187005,'hhhhhh',SYS003
// EXEC SORT
   Sort/merge control cards (6 cards)
/&
```

Note: Replace hhhhhh in the DLAB and XTENT cards with the volume serial number of the disk pack.

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 41:

- Job control cards
- Sort control cards
- Sort parameters
- Computed constants

```
// JOB SORT 2314
// PAUSE ASSGN SYS001,SYS002,SYS003 TO SCRATCH PACK
// ASSGN SYS001,X'230'
// ASSGN SYS002,X'230'
// ASSGN SYS003,X'231'
// VOL SYS001,SORTOUT
// DLAB 'SORTED FILE OF 3000 RECORDS FOR UTILITIES   1X30X30',         X
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000171000,000174002,'X30X30',SYS001
// VOL SYS002,SORTIN1
// DLAB 'SAMPLE PROBLEM FILE OF 3000 RECORDS         1X30X30',         X
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000180000,000184002,'X30X30',SYS002
// VOL SYS003,SORTWK1
// DLAB 'SAMPLE PROGRAM WORK AREA FOR SORT RUN       1X31X31',         X
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000184003,000187005,'X31X31',SYS003
// EXEC SORT

7000I  SORT FIELDS=(11,4,A),FORMAT=BI,SIZE=3000,WORK=1                    $4830201
7000I  RECORD TYPE=F,LENGTH=(15,,15)                                       $4830202
7000I  INPFIL BLKSIZE=75                                                   $4830203
7000I  OUTFIL BLKSIZE=75                                                   $4830204
7000I  OPTION PRINT=ALL,LABEL=(S,S),STORAGE=22528                          $4830205
7000I  END                                                                 $4830206
7050I NMAX = 00013540
7051I B = 00001683
7052I G = 00000443
7001I PHASE 0 END,NO DETECTED ERRORS
```

```
EOJ SORT
```

Figure 41. SYSLST Output for Tape and Disk Sort/Merge Sample Problem (SM6)

Disk-to-Printer Utility

Program Name is Z.UTDKPR3. The disk-to-printer utility sample problem lists the numbers handled by the Assembler 6 and tape and disk sort/merge (2314 application) sample problems. To execute the disk-to-printer utility sample problem, the following job control cards are needed:

```
                                                   Col. 54     Col. 72
// JOB DKPR 2314 TO PRINTER
// PAUSE ASSGN SYS004 TO SCRATCH PACK
// ASSGN SYS004,X'cuu'
// PAUSE ASSGN SYS005 TO THE PRINTER
// ASSGN SYS005,X'cuu'
// UPSI 000
// VOL SYS004, UIN
// DLAB 'SORTED FILE OF 2000 RECORDS FOR UTILITIES  1hhhhhh',    X
               Col. 16
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000171000,000174002,'hhhhhh',SYS004
// EXEC DKPR
               Disk-to-printer utility control cards (2 cards)
/*
```

Note: Replace hhhhhh in the DLAB and XTENT cards with the volume serial number of the disk pack.

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 42:

- Job control cards
- Disk-to-printer control cards
- Utility parameter statements (10 lines)
- Numbers 0001-3000 (600 lines)
- NUMBER OF INPUT BLOCKS PROCESSED 000600
- NUMBER OF OUTPUT BLOCKS PROCESSED 000600
- END OF JOB

```
// JOB DKPR 2314 TO PRINTER
// PAUSE ASSGN SYS004 TO SCRATCH PACK
// ASSGN SYS004,X'230'
// PAUSE ASSGN SYS005 TO THE PRINTER
// ASSGN SYS005,X'00E'
// UPSI 000
// VOL SYS004, UIN
// DLAB 'SORTED FILE OF 3000 RECORDS FOR UTILITIES   1X30X30',          X
               0001,66080,66080,' SAMPLE TEST '
// XTENT 1,000,000171000,000174002,'X30X30',SYS004
// EXEC DKPR
```

```
// UDP TL,FF,A=(75,75),B=(120),OC,S1,PY,E=(2314)                $4610201
// END                                                          $4610202
DISK TO PRINT UTILITY
INPUT RECORD LENGTH 0075
OUTPUT RECORD LENGTH  0120
INPUT BLOCK LENGTH 00075
OUTPUT BLOCK LENGTH 00120
OUTPUT OPTION PRINT CHARACTER
2 INPUT,2 OUTPUT AREAS ASSIGNED
RECORD FORMAT FIXED
TYPE LIST
STARTING RECORD NUMBER  00000001
INPUT DEVICE TYPE  2314
```

```
          0001          0002          0003          0004          0005
          0006          0007          0008          0009          0010
          0011          0012          0013          0014          0015
          0016          0017          0018          0019          0020
          0251          0252          0253          0254          0255
          0256          0257          0258          0259          0260
          0261          0262          0263          0264          0265
          0266          0267          0268          0269          0270
PAGE  1
```

Figure 42. SYSLST Output for Disk-to-Printer Sample Problems (UTDKPR3) (Part 1 of 5)

0271	0272	0273	0274	0275
0276	0277	0278	0279	0280
0281	0282	0283	0284	0285
0286	0287	0288	0289	0290
0531	0532	0533	0534	0535
0536	0537	0538	0539	0540
0541	0542	0543	0544	0545
0546	0547	0548	0549	0550

PAGE 2

0551	0552	0553	0554	0555
0556	0557	0558	0559	0560
0561	0562	0563	0564	0565
0566	0567	0568	0569	0570
0801	0802	0803	0804	0805
0806	0807	0808	0809	0810
0811	0812	0813	0814	0815
0816	0817	0818	0819	0820
0821	0822	0823	0824	0825
0826	0827	0828	0829	0830

PAGE 3

0831	0832	0833	0834	0835
0836	0837	0838	0839	0840
0841	0842	0843	0844	0845
0846	0847	0848	0849	0850
1081	1082	1083	1084	1085
1086	1087	1088	1089	1090
1091	1092	1093	1094	1095
1096	1097	1098	1099	1100
1101	1102	1103	1104	1105
1106	1107	1108	1109	1110

PAGE 4

Figure 42. SYSLST Output for Disk-to-Printer Sample Problems (UTDKPR3) (Part 2 of 5)

```
1111          1112          1113          1114          1115
1116          1117          1118          1119          1120
1121          1122          1123          1124          1125
```

```
1371          1372          1373          1374          1375
1376          1377          1378          1379          1380
1381          1382          1383          1384          1385
1386          1387          1388          1389          1390
PAGE  5
```

```
1391          1392          1393          1394          1395
1396          1397          1398          1399          1400
1401          1402          1403          1404          1405
1406          1407          1408          1409          1410
```

```
1656          1655          1654          1653          1652
1651          1650          1658          1659          1660
1661          1662          1663          1664          1665
1666          1667          1668          1669          1670
PAGE  6
```

```
1671          1672          1673          1674          1675
1676          1677          1678          1679          1680
1681          1682          1683          1684          1685
```

```
1931          1932          1933          1934          1935
1936          1937          1938          1939          1940
1941          1942          1943          1944          1945
1946          1947          1948          1949          1950
PAGE  7
```

Figure 42. SYSLST Output for Disk-to-Printer Sample Problems (UTDKPR3) (Part 3 of 5)

1951	1952	1953	1954	1955
1956	1957	1958	1959	1960
1961	1962	1963	1964	1965
2211	2212	2213	2214	2215
2216	2217	2218	2219	2220
2221	2222	2223	2224	2225
2226	2227	2228	2229	2230

PAGE 8

2231	2232	2233	2234	2235
2236	2237	2238	2239	2240
2241	2242	2243	2244	2245
2246	2247	2248	2249	2250
2486	2487	2488	2489	2490
2491	2492	2493	2494	2495
2496	2497	2498	2499	2500
2501	2502	2503	2504	2505
2506	2507	2508	2509	2510

PAGE 9

2511	2512	2513	2514	2515
2516	2517	2518	2519	2520
2521	2522	2523	2524	2525
2771	2772	2773	2774	2775
2776	2777	2778	2779	2780
2781	2782	2783	2784	2785
2786	2787	2788	2789	2790

PAGE 10

Figure 42. SYSLST Output for Disk-to-Printer Sample Problem (UTDKPR3) (Part 4 of 5)

```
     2791          2792          2793          2794          2795
     2796          2797          2798          2799          2800
     ...
     2986          2987          2988          2989          2990
     2991          2992          2993          2994          2995
     2996          2997          2998          2999          3000
 END OF DATA
PAGE 11
```

```
NUMBER OF INPUT BLOCKS PROCESSED  000600
NUMBER OF OUTPUT BLOCKS PROCESSED 000600
END OF JOB
```

```
EOJ DKPR
```

Figure 42. SYSLST Output for Disk-to-Printer Sample Problem (UTDKPR3) (Part 5 of 5)

Autotest

Program Name is Z.AT1. The Autotest sample problem demonstrates to the problem programmer some of the more useful features of Autotest. Basically, the program is a two-step loop that:

1. Reads a data card, and

2. Moves certain fields of the card to designated storage areas.

This simple loop is repeated five times before the end-of-job routine is entered. The Autotest sample problem source program listing is shown in Figure 43.

If the Autotest sample problem is run in a multiprogramming system, no core may be allocated to the foreground area (F1 and F2=0K).

All Autotest control cards have a ./ punched in columns 1 and 2, respectively. The first and second Autotest control cards perform any phase and control section qualification included in the test deck. Because the example has only a single phase and one control section, no modification is required and these two cards could be eliminated.

Autopatch capabilities are demonstrated by the next six control cards, resulting in the following action:

- A new instruction is added to the program between two other existing instructions.
- One instruction is exchanged for another.
- The assembled value of a constant is changed.

Another important function of Autotest is demonstrated by the test request cards (./ TR). The first of these cards initiates test request Number 001, and the following continuation card initiates test request Number 002. These two cards result in the following:

- The 16 general registers and the two areas of storage at SYMA and SYMC+3 are displayed prior to executing the instruction at THERE. (Because there are five data cards, this request will be executed five times.)
- The storage locations IN+75 and IN+76 are displayed for only the second and fourth data cards as specified by the ON parameters.

The ./ PCC card (program control card) gives a normal end-of-job storage printout. This printout can have one or more formats.

The user installation must define the work area for Autotest if nonstandard labels are used. In the case of nonstandard labels, the job control cards required are: DLBL and EXTENT. The file name in the VOL card must be IJSYSAT. Twenty tracks are required in the EXTENT card. Only one work area may be assigned and it must be continuous unless the split-cylinder concept is used. If this concept is used, refer to IBM System/360 Disk Operating System, System Control and System Service Programs, Form C24-5036. For additional information concerning Autotest work files, refer to IBM System/360 Disk Operating System, Autotest, Form C24-5062.

The job control cards used in executing the Autotest sample problem are:

```
// JOB AUTOTEST
// OPTION SYM,LINK
// EXEC ASSEMBLY
   AUTOTEST SOURCE DECK (27 cards)
/*
// EXEC ATLEDT
./ PHQ AAAAAAAA
./ CSQ CSECT1
./ ADD THERE -6,40
         MVC      SYMB,SYMA   *D203A08EA086
./ EXC BC2,2
 * 4710A034
./ CON SYMA,2
 * 0000000000000000
./ TR THERE,,CPL,SYMA,SYMC+3,,C,G
            Col.  16
            DSP,IN+75,IN+76,(2,4,2),B
./ PCC AMS,AMS,2000
// EXEC
   AUTOTEST DATA DECK (5 cards)
/*
./ ATEOF
```

SYSLOG output is shown in Figure 17.

The following list summarizes the SYSLST output shown in Figure 43:

- Job control cards
- External symbol dictionary
- Source program listing
- Relocation dictionary
- Cross-reference
- Linkage editor input diagnostic
- ./ Autotest control cards (11 cards)
- Autotest data (5 cards)
- Test requests
- Normal EOF storage dump
- EOJ AUTOTEST

Autotest Problem Test Results

For the following discussion, refer to the circled letters on the SYSLST output for the Autotest sample problem, Figure 43. The Autopatch control card (./ ADD THERE-6,40) inserts an additional instruction between statement numbers 10 and 11 (see A). The core dump at the end of the program shows that the instruction at MVC2 (see E) has been replaced by a supervisor call (SVC 254). This instruction forms a link to the patch area in storage. The actual patch is displayed at (F) and consists of two MVC instructions and SVC 255 which forms the return link to the instruction located at THERE. The instruction (./ EXC BC2,2) causes the instruction at BC2 (see B) to be replaced by SVC 253. The core dump shows the resulting patch at points G and H. Finally, the constant at C is changed from one to zero before program execution. The test requests at location THERE (see D) are shown on a separate page of the listing (Figure 43). Test request Number 1 is displayed a total of five times, once for each data card. Test request Number 2 appears only for the second and fourth data cards. The rest of the printout contains the normal EOJ core dump.

```
// JOB AUTOTEST
// OPTION SYM,LINK
// EXEC ASSEMBLY
```

```
                                        EXTERNAL SYMBOL DICTIONARY
SYMBOL   TYPE ID  ADDR  LENGTH LD ID

CSECT1     SD  01 000000 000084
```

```
LOC   OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                DOS CL3-0

                                      1             REPRO                                  $4590001
                                      2             PHASE AAAAAAAA,S                       $4590002
000000                                3 CSECT1      START 0                                $4590003
000002                                4 R2          EQU   2                                $4590004
000003                                5 R3          EQU   3                                $4590005
00000A                                6 R10         EQU   10                               $4590006
000000 05A0                           7 HERE        BALR  R10,0                            $4590007
000002                                8             USING *,R10                            $4590008
000002 4520 A018            0001A     9 BAL1        BAL   R2,READ                          $4590009
000006 D203 A086 A040 00088 00042 A→10 MVC1        MVC   SYMA(4),IN+10                    $4590010
00000C D203 A092 A081 00094 00083    11 MVC2        MVC   SYMC(4),IN+75                    $4590011
000012 4830 A092            00094 D→12 THERE       LH    R3,SYMC                          $4590012
000016 47F0 A000            00002    13 BC1         BC    15,HERE+2                        $4590013
00001A                               14 READ        EQU   *                                $4590014
                                     15 CHAN        EXCP  CARDCCB                          $4590015
                                     16+* 360N-CL-453 EXCP      CHANGE LEVEL  3-0
00001A 5810 A0AE            000B0    17+CHAN        L     1,=A(CARDCCB)
00001E 0A00                          18+            SVC   0
                                     19 WAIT1       WAIT  CARDCCB                          $4590016
                                     20+* 360N-CL-453 WAIT      CHANGE LEVEL  3-0
000020 5810 A0AE            000B0    21+WAIT1       L     1,=A(CARDCCB)
000024 9180 1002      00002          22+            TM    2(1),X'80'
000028 4710 A02C            0002E    23+            BO    *+6
00002C 0A07                          24+            SVC   7
00002E 9101 A09A      0009C          25 TEST2       TM    CARDCCB+4,1                      $4590017
000032 0712                       B→26 BC2         BCR   1,R2                             $4590018
000034 07F2                          27 BCR1        BCR   15,R2                            $4590019
                                     28 EOJ         EOJ                                    $4590020
                                     29+* 360N-CL-453 EOJ       CHANGE LEVEL  3-0
000036 0A0E                          30+EOJ         SVC   14
000038 4040404040404040              31 IN          DC    CL80' '                          $4590021
000088 0000000100000001           C→32 SYMA        DC    2F'1'                            $4590022
000090 00000000                      33 SYMB        DC    F'0'                             $4590023
000094 00000000                      34 SYMC        DC    2H'0'                            $4590024
                                     35 CARDCCB     CCB   SYSIPT,CARDCCW                   $4590025
                                     36+* 360N-CL-453 CCB       CHANGE LEVEL  3-0
000098 0000                          37+CARDCCB     DC    XL2'0' RESIDUAL COUNT
00009A 0000                          38+            DC    XL2'0' COMMUNICATIONS BYTES
00009C 0000                          39+            DC    XL2'0' CSW STATUS BYTES
00009E 00                            40+            DC    AL1(0) LOGICAL UNIT CLASS
00009F 01                            41+            DC    AL1(1) LOGICAL UNIT
0000A0 00                            42+            DC    XL1'0'
0000A1 0000A8                        43+            DC    AL3(CARDCCW) CCW ADDRESS
0000A4 00                            44+            DC    B'00000000' STATUS BYTE
0000A5 000000                        45+            DC    AL3(0) CSW CCW ADDRESS
0000A8 0200003820000050              46 CARDCCW     CCW   2,IN,32,80                       $4590026
000000                               47             END   HERE                             $4590027
0000B0 00000098                      48                   =A(CARDCCB)
```

```
                                        RELOCATION DICTIONARY

POS.ID   REL.ID   FLAGS   ADDRESS

  01       01      08     0000A1
  01       01      08     0000A9
  01       01      0C     0000B0
```

Figure 43. SYSLST Output for Autotest Sample Problems (Part 1 of 4)

```
                                        CROSS-REFERENCE

SYMBOL     LEN   VALUE  DEFN

BAL1      00004 000002 0009
BCR1      00002 000034 0027
BC1       00004 000016 0013
BC2       00002 000032 0026
CARDCCB   00002 000098 0037    0017   0021   0025   0048
CARDCCW   00008 0000A8 0046    0043
CHAN      00004 00001A 0017
CSECT1    00001 000000 0003
EOJ       00002 000036 0030
HERE      00002 00000C 0007    0013   0047
IN        00080 000038 0031    0010   0011   0046
MVC1      00006 000006 0010
MVC2      00006 00000C 0011
READ      00001 00001A 0014    0009
R10       00001 00000A 0006    0007   0008
R2        00001 000002 0004    0009   0026   0027
R3        00001 000003 0005    0012
SYMA      00004 000088 0032    0010
SYMB      00004 000090 0033
SYMC      00002 000094 0034    0011   0012
TEST2     00004 00002E 0025
THERE     00004 000012 0012
WAIT1     00004 000020 0021

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
```

```
// EXEC ATLEDT
```

```
JOB  AUTOTEST           AUTOTEST DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN    MAP
LIST            PHASE AAAAAAAA,S                                  $4590002
LIST    ENTRY
```

```
         PHASE   XFR-AD   LOCORE   HICORE   DSK-AD    ESD TYPE   LABEL     LOADED   REL-FR

        AAAAAAAA  002000   002000   002083   21 5 2    CSECT      CSECT1    002000   002000
```

```
SVC=  TR=                                    CARD-IMAGE                                    PPA-TPA     ERROR/S
           ./ PHQ AAAAAAAA
           ./ CSQ CSECT1
           ./ ADD THERE-6,40                                                               00200C
254                 MVC    SYMB,SYMA               *D203A08EA086
           ./ EXC BC2,2                                                                    002032
253         *4710A034
           ./ CON SYMA,2                                                                   002088
            *0000000000000000
252   001  ./ TR THERE,,DPL,SYMA,SYMC+3,,C,G                              C                002012
      002                    DSP,IN+75,IN+76,(2,4,2),B
           ./ PCC AMS,AMS,2000
```

Figure 43. SYSLST Output for Autotest Sample Problems (Part 2 of 4)

```
// EXEC
```

```
TEST REQUEST ID  001  ON  00001

REG0     0000FB90     00002098     9C002006     00002000     0000FF8C     0000FC15     A000213C     C0000002
REG8     00000043     0C000000     50002002     00002000     00002C38     00003C38     000001B0     00C0008B

        002088  0 0 0 1               0 0 0 1    A A A A

TEST REQUEST ID  001  ON  00002

REG0     0000FB90     00002098     8C002006     FFFFC1C1     0000FF8C     0000FC15     A000213C     C0000002
REG8     00000043     00000000     5C002002     00002000     00002C38     00003C38     000001B0     C000008B

        002088  0 0 0 2               0 0 0 2    B B B B

TEST REQUEST ID  002  ON  00002

        002083    C2C2
                  B B

TEST REQUEST ID  001  ON  00003

REG0     0000F890     0C002098     80002006     FFFFC2C2     0000FF8C     0000FC15     A000213C     C0000002
REG8     00000043     00000000     50002002     00002000     00002C38     00003C38     000001B0     0000008B

        002088  0 0 0 3               0 0 0 3    C C C C

TEST REQUEST ID  001  ON  00004

REG0     0000FB90     0C002098     80002006     FFFFC3C3     0000FF8C     0000FC15     A000213C     C0000002
REG8     00000043     0C000000     5C002002     00002000     00002C38     00003C38     000001B0     0000008B

        002088  0 0 0 4               0 0 0 4    D D D D

TEST REQUEST ID  002  ON  00004

        002083    C4C4
                  D D

TEST REQUEST ID  001  ON  00005

REG0     0C00F890     00002098     8C002006     FFFFC4C4     0C00FF8C     0000FC15     A000213C     00000002
REG8     00000043     0C000000     50002002     00002000     00002C38     00003C38     000001B0     0000008B

        002088  0 0 0 5               0 0 0 5    E E E E
```

```
LIST OF PHASE-NAMES

AAAAAAAA        FETCH    002000       0020B3
```

```
NORMAL END OF JOB
USER PSW - FF15000E70002038

GR 0-7  0000FB90 00002098 80002006 FFFFC5C5 0000FF8C 0000FC15 A000213C 00000002
GR 8-F  00000043 00000000 50002002 00002000 00002C38 00003C38 000001B0 0000008B
FR 0-6  4131998D1F000000  4153235060400000  BE7C2000E9886200  0000000000000000

COMREG  F0F161F0 F161F6F8 20002000 00000000 00000000 00000000 C1E4E3D6 E3C5E2E3 0000FFFF 000020B3 000020B3 0000
```

Figure 43. SYSLST Output for Autotest Sample Problems (Part 3 of 4)

```
-LABEL LENGTH- 0000

 S2        BAL1
 S1     CSECT1       MVC1                  E MVC2                THERE      BC1        READ
002000  05A04520  A018D203  A086A040  0AFE0700  002010  07000AFC  070047F0  A0005810  A0AE0A00
 CH                  K                                                 0
 MN     BALRBAL      MVC                 SVC BCR           BCR SVC   BCR BC       L          SVC

 S2                                                                  EOJ
 S1     WAIT1                             TEST2             G BC2   BCR1      IN
002020  5810A0AE  91801002  4710A02C  0A079101  002030  A09A0AFD  07F20A0E  615C4040  40404040
 CH                                                                 2        / *
 MN     L         TM  LPR   BC        SVC TM                  SVC   BCR SVC      STH    STH STH

002040  404C4040  40404040  40404040  40404040  002050  40404040  40404040  40404040  40404040
 CH
 MN     STH STH   STH STH   STH STH   STH STH           STH STH   STH STH   STH STH   STH STH
002060  DITTO
```

```
00FFA0  07C1FF00  22FC1200  0002C1C1  C1C1C1C1  00FFB0  C1C11200  04001200  04101400  09001400
 CH        A                     A A  A A A A           A A
 MN     BCR       LTDRLTR                                   LTR   SPM LTR   SPM NR    ISK NR

00FFC0  09011400  02000000  20B30020  38090000  00FFD0  00000100  00000000  00FF0000  20B30000
 CH
 MN     ISK NR              LPDR      LER                                             LPDR

00FFE0  12000100  12000103  12000200  12000201  00FFF0  12000300  12000300  00000000  00000000
 CH
 MN     LTR       LTR       LTR       LTR               LTR       LTR
 END
```

```
EOJ AUTOTEST
```

Figure 43. SYSLST Output for Autotest Sample Problems (Part 4 of 4)

Multiprogramming

Multiprogramming capabilities are demonstrated by the multiprogramming sample problem. A control program supporting multiprogramming is required. A card-to-printer utility macro lists fifteen test cards containing the instructions and necessary operator commands to initiate the job in the foreground-one (F1) area. The card-to-printer utility macro is first cataloged and then assembled and executed as a background job. The fifteen test cards are listed on the printer. When the last card in the reader has been read and processed, the following message will be printed on SYSLOG:

    0P08A INTERV REQ SYSRDR ...

To clear the intervention required condition on the reader, the operator should enter the following two cards in the reader and ready the device:

    /&

    // PAUSE REFER TO PRINTER FOR ADDITIONAL INSTRUCTIONS

The operator can now initiate this same program in the foreground area by following the instructions previously listed on SYSLST when the job was run in the background problem area.

The same data previously used for the background execution (fifteen test cards) is used as input data for the foreground execution. The data cards must, therefore, be removed from the card reader output stacker and they must be replaced in the input hopper. Then the card reader is made ready. The operator should now follow the instructions contained in the test data to initiate the job in the foreground area.

Several assumptions have been made in the multiprogramming sample problem. The first assumption is that a foreground-one area has previously been allocated. If this has not been done, the operator should refer to the DOS Operating Guide referenced on the cover of this publication.

The second assumption is that SYSRDR or SYSIPT is assigned to 00C and SYSLST assigned to 00E and that there are no other logical units assigned to these devices. If this is not the case, message

    1A13D CONFLICTING I/O ASSIGNMENTS

will be issued on SYSLOG when the operator tries to initiate the program in the foreground area. To recover from this condition, the operator should reply CANCEL to the preceding message. The message

    F1 0S02I JOB FG INIT. CANCELED DUE TO PROGRAM REQUEST

will then be issued. The operator should then press the Request key on the 1052-Printer-Keyboard. The message

    1I60A READY FOR COMMUNICATIONS

will be issued. The operator should next enter the START command followed by

    ASSGN SYSxxx,UA

where xxx is the unit(s) retaining a background assignment for the indicated device. The operator can now initiate the job in the foreground-one area as previously described. To execute the multiprogramming sample problem the following cards must be entered:

```
// JOB CATALOG SAMPLE FOREGROUND/BACKGROUND PROGRAM
// OPTION CATAL
   PHASE CARDPRNT,+0
// EXEC ASSEMBLY
         Column 10
         PRINT NOGEN
         START 0
         INCARD
         OUTPRT BUFSIZ=80
         END
/*
// EXEC LNKEDT
/&
// JOB EXECUTE SAMPLE FOREGROUND/BACKGROUND PROGRAM
// ASSGN SYS001,X'00C'                    (Note 1)
// ASSGN SYS002,X'00E'                    (Note 2)
// PAUSE CARD READER END-OF-FILE SIGNALS END OF CARD INPUT
// EXEC CARDPRNT
* THESE COMMENTS AND THE FOLLOWING STATEMENTS ARE TEST INPUT
* NOW IN A PAUSE STATUS, UNASSIGN THE PRINTER (X'00E') AND THE CARD
* READER (X'00C') TO PERMIT SUBSEQUENT ASSIGNMENT TO FOREGROUND
* AREA.  TO DO THIS, TYPE THE FOLLOWING
ASSGN SYSRDR,UA
ASSGN SYSIPT,UA                           (Note 3)
ASSGN SYSLST,UA
* TO SUSPEND BACKGROUND (BG) AREA PROCESSING TYPE
STOP
* TO INITIATE PROGRAM NAMED CRDPRNT IN FOREGROUND1 (F1) AREA,
* THE FOLLOWING FOUR STATEMENTS ARE TO BE TYPED ON SYSLOG (1052)
START F1                                  (Note 4)
ASSGN SYS001,X'00C'                       (Note 1)
```

```
ASSGN SYS002,X'00E'                    (Note 2)
EXEC CARDPRNT
```

Note 1: SYS001 must be assigned to a card reader.

Note 2: SYS002 must be assigned to a printer.

Note 3: These cards are required only if SYS001 or SYS002 retain a background assignment. A device used by a foreground program cannot be assigned to a background area at the same time.

Note 4: This command is processed by the ATTN routine (Refer to DOS Operating Guide referenced on the cover of this publication.)

The test cards for the multiprogramming sample problem are output on SYS002, which was previously assigned to the system printer. LOG must be keyed on the 1052 Printer-Keyboard at the beginning of the job to obtain a complete listing of all job control cards and messages. The SYSLOG output is shown in Figure 18. The SYSLST output in Figure 44 contains:

- Job control cards
- External symbol dictionary
- Source program listing
- Relocation dictionary
- Diagnostics
- Linkage editor storage map
- List of test cards processed by card-to-printer utility program

```
// JOB CATALOG SAMPLE FOREGROUND/BACKGROUND PROGRAM
// OPTION CATAL
 PHASE CARDPRNT,+0
// EXEC ASSEMBLY
```

```
                                   EXTERNAL SYMBOL DICTIONARY                    PAGE    1
SYMBOL  TYPE ID  ADDR  LENGTH LD ID

        PC  01 000000 00034A
```

```
                                                                                 PAGE    1

  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     1           PRINT NOGEN
000000                               2           START 0
                                     3           INCARD
                                   109           OUTPRT  BUFSIZ=80
                                   237           END
```

```
                                       RELOCATION DICTIONARY                     PAGE   1

POS.ID   REL.ID   FLAGS   ADDRESS

  01       01      CC     000050
  01       01      C8     0000C1
  01       01      C8     0000C9
  01       01      C8     0000D9
  01       01      C8     0000E1
  01       01      0C     0001CC
  01       01      C8     000269
  01       01      C8     000271
  01       01      C8     000281
  01       01      C8     000289
  01       01      C8     000299
  01       01      C8     0002A1
```

Figure 44. SYSLST Output for Multiprogramming Sample Problems (Part 1 of 5)

```
                                                  CROSS-REFERENCE                                                    PAGE    1

SYMBOL    LEN   VALUE  DEFN

CDAA0001 00004 000004 0009    0030  0044  0046  0093
CDAC0001 00004 000050 0030    0014  0028
CDAD0001 00080 0C0C0C 0032    0056
CDAE0001 00080 00005C 0033    0066
CDAF0001 00002 0000B0 0043    0099  0102  0104
CDAG0001 00002 0000B4 0045    0103
CDAH0001 00002 0000B8 0047    0016  0018  0044
CDAJ0001 00008 0000C8 0056    0019  0021  0053
CDAK0001 00002 0000D0 0057    0022  0024  0046
CDAL0001 00008 0000EC 0066    0025  0027  0063
CDAN0001 00004 0000F8 0069    0015  0029  0096
CDAP0001 00004 00010C 0071    0097
CDAQ0001 00004 00010A 0074    0072
CDAR0001 00004 00011E 0080    0075
CDAS0001 00004 00012E 0084    0079
CDAT0001 00004 000160 0099    0069  0083
CDAU0001 00002 000166 0101
CDBA0001 00004 00000C 0011    0031
CDBM0001 00004 000158 0098    0084  0085  0091  0092
CDCM0001 00004 0000E8 0067    0012  0095  0106
CDEN0001 00004 000176 0106    0087
CDRE0001 00002 00013C 0089    0010
CDRZ0001 00004 00014E 0096    0070  0078
CDR00001 00001 000000 0034    0076  0076  0081
CDR10001 00001 000001 0035    0071  0074  0080  0084  0086  0091  0099  0100
CDR20001 00001 000002 0036    0012  0013  0014  0017  0020  0023  0026  0077  0077  0080  0081  0082  0086  0095  0106
CDR30001 00001 000003 0037    0016  0017  0018  0019  0020  0021  0022  0023  0024  0025  0026  0027  0069  0083  0105
CDR40001 00001 000004 0038    0102  0103
CDR50001 00001 000005 0039    0012  0095  0102  0104  0106
CDSW0001 00004 000008 0010    0011
CDXX0001 00001 00017A 0108
CD140001 00001 00000E 0040    0087
CD150001 00001 00000F 0041    0007  0008  0009  0009  0013  0028  0085  0089  0090  0092  0093  0094  0094  0100  0107
PRAA0002 00004 00017C 0115    0138  0153  0154
PRAC0002 00004 000188 0118    0139
PRAD0002 00004 0001CC 0138    0120
PRAE0002 00080 000188 0140    0163
PRAF0002 00080 0001D8 0141    0171
PRAH0002 00004 000238 0153    0200  0202  0203
PRAJ0002 00004 00023C 0154    0189  0201  0228
PRAK0002 00004 00024C 0155    0116  0224  0231
PRAQ0002 00002 0002AC 0184    0117  0121  0137
PRAR0002 00004 0002CC 0191
PRAU0002 00001 0002DC 0199    0195
PRAV0002 00004 0002E8 0203
```

Figure 44. SYSLST Output for Multiprogramming Sample Problems (Part 2 of 5)

```
                                                CROSS-REFERENCE                                                  PAGE    2

SYMBOL     LEN   VALUE  DEFN

PRAW0002 00001 000318 0216     0207
PRAZ0002 00004 0C031A C218
PRB1C002 00006 00026C 0156     0122  0124  0154
PRB20002 00006 0C0278 0164     0125  0127  0153
PRB30002 00006 0C029C C172     0134  0136  0196
PRDA0002 00004 0C0320 0220
PRDB0002 00006 CC0324 0221     0220
PRDC0002 00002 0C032A 0222
PRDD0002 00004 00032E 0224     0187  0233
PRDE0002 00001 0C0334 0227     0185
PREM0002 00008 00C230 0152     0212
PRENC002 00001 00034A 0236
PRFB0002 00004 0002A8 0182     0206
PRMH0002 00004 000228 0151     0210  0214
PRSV0002 00002 00032C 0223
PRSW0002 00004 C00184 0117     0118
PRS10002 00004 0002FA 0208     0209
PRS30002 00004 000314 0215     0208
PRT90002 00004 0C02CA 0194
PRWA0002 00008 000270 0163     0128  0130  0160
PRWB0002 00008 000288 0171     0131  0133  0168
PRW10002 00001 0C02A0 0179
PRW200C2 00001 0002AC 0180
PRW300C2 C0008 0002A0 0181     0176  0197
PR000002 00001 0C0000 0142     0184  0184  0210  0211  0214  0217
PR010002 00001 C00001 0143     0186  0186  0188  0189  0190  0191  0194  0196  0210  0212  0214  0217  0221  0222  0228
                               0229  0230  0231  0232  0232
PR020002 000C1 CC0002 0144     0119  0120  0123  0126  0129  0132  0135  0188  0206  0215  0218  0219  0220
PR030002 00001 000003 0145     0122  0123  0124  0125  0126  0127  0128  0129  0130  0131  0132  0133  0134  0135  0136
PR040002 00001 C000C4 0146     0116  0224
PR130002 000C1 C0000D 0147     0116  0200  0201  0205  0221  0224
PR140002 00001 00000E 0148     0200  0202  0203  0204  0205  0218  0222  0225
PR150002 00001 C0000F 0149     0113  0114  0115  0115  0119  0190  0204  0229  0235


NO STATEMENTS FLAGGED IN THIS ASSEMBLY
```

Figure 44. SYSLST Output for Multiprogramming Sample Problems (Part 3 of 5)

```
// EXEC LNKEDT
```

```
JOB  CATALOG              DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN      MAP
LIST   PHASE CARDPRNT,+0
LIST   ENTRY
```

```
        PHASE  XFR-AD  LOCORE  HICORE  DSK-AD   ESD TYPE  LABEL     LOADED  REL-FR

      CARDPRNT  000000  000000  000349  28 0 1   CSECT               000000  000000
```

```
SYSTEM DIRECTORY                    CORE-IMAGE                 RELOCATABLE                SOURCE-STATEMENT
                                    ---------------------------DECIMAL---------------------------------------

                                      C  H  R  E                 C  H  R  E                 C  H  R  E
DIRECTORY STARTING ADDRESS            01 00 01                   64 00 01                   101 00 01
DIRECTORY NEXT ENTRY                  01 02 07 11                64 01 02 00                101 00 14 06
DIRECTORY LAST ENTRY                  01 04 08 17                64 04 09 22                101 01 16 09

LIBRARY   STARTING ADDRESS            01 05 01                   64 05 01                   101 02 01
LIBRARY   NEXT AVAILABLE ENTRY        43 00 02                   83 08 09                   142 06 07
LIBRARY   LAST AVAILABLE ENTRY        63 09 02                  100 09 09                   149 09 16
                                    ----------------------STATUS  INFORMATION---------------------------------

DIRECTORY ENTRIES ACTIVE              403                        230                        136

LIBRARY   BLOCKS ALLCCATED            1250                       3285                       7808
LIBRARY   BLOCKS ACTIVE               827                        1745                       6630
LIBRARY   BLOCKS DELETED               04                          00                         00
LIBRARY   BLOCKS AVAILABLE            419                        1540                       1178

AUTOMATIC CONDENSE LIMIT               00                        4000                       6000

LIBRARY   ALLOCATED CYLINDERS          63                          37                         49
DIRECTORY ALLOCATED TRACKS             05                          05                         02
```

```
EOJ CATALOG
```

Figure 44. SYSLST Output for Multiprogramming Sample Problems (Part 4 of 5)

```
// JOB EXECUTE SAMPLE FOREGROUND/BACKGROUND PROGRAM
// ASSGN SYS001,X'00C'          CARD READER
// ASSGN SYS002,X'00E'          PRINTER
// PAUSE CARD READER END-OF-FILE SIGNALS END OF CARD INPUT
// EXEC CARDPRNT
```

```
* THESE COMMENTS AND THE FOLLOWING STATEMENTS ARE TEST INPUT
* NOW IN A PAUSE STATUS, UNASSIGN THE PRINTER (X'00E') AND THE CARD
*      READER (X'00C') TO PERMIT SUBSEQUENT ASSIGNMENT TO FOREGROUND
*      AREA. TO DO THIS, TYPE THE FOLLOWING
ASSGN SYSRDR,UA    (REQUIRED ONLY IF SYSRDR ASSIGNED TO X'00C')
ASSGN SYSIPT,UA    (REQUIRED ONLY IF SYSIPT ASSIGNED TO X'00C')
ASSGN SYSLST,UA    (REQUIRED ONLY IF SYSLST ASSIGNED TO X'00E')
* TO SUSPEND BACKGROUND (BG) AREA PROCESSING, TYPE
STOP
* TO INITIATE PROGRAM NAMED CARDPRNT IN FOREGROUND 1 (F1) AREA, THE
*      FOLLOWING FOUR STATEMENTS ARE TO BE TYPED ON SYSLOG (1052)
START F1      (THIS COMMAND IS PROCESSED BY THE 1052 ATTENTION ROUTINE)
ASSGN SYS001,X'00C'          CARD READER
ASSGN SYS002,X'00E'          PRINTER
EXEC CARDPRNT
```

```
EOJ EXECUTE
```

```
// PAUSE REFER TO PRINTER FOR ADDITIONAL INSTRUCTIONS
ASSGN SYSRDR,UA
ASSGN SYSIPT,UA
ASSGN SYSLST,UA
```

```
* THESE COMMENTS AND THE FOLLOWING STATEMENTS ARE TEST INPUT
* NOW IN A PAUSE STATUS, UNASSIGN THE PRINTER (X'00E') AND THE CARD
*      READER (X'00C') TO PERMIT SUBSEQUENT ASSIGNMENT TO FOREGROUND
*      AREA. TO DO THIS, TYPE THE FOLLOWING
ASSGN SYSRDR,UA    (REQUIRED ONLY IF SYSRDR ASSIGNED TO X'00C')
ASSGN SYSIPT,UA    (REQUIRED ONLY IF SYSIPT ASSIGNED TO X'00C')
ASSGN SYSLST,UA    (REQUIRED ONLY IF SYSLST ASSIGNED TO X'00E')
* TO SUSPEND BACKGROUND (BG) AREA PROCESSING, TYPE
STOP
* TO INITIATE PROGRAM NAMED CARDPRNT IN FOREGROUND 1 (F1) AREA, THE
*      FOLLOWING FOUR STATEMENTS ARE TO BE TYPED ON SYSLOG (1052)
START F1      (THIS COMMAND IS PROCESSED BY THE 1052 ATTENTION ROUTINE)
ASSGN SYS001,X'00C'          CARD READER
ASSGN SYS002,X'00E'          PRINTER
EXEC CARDPRNT
```

Figure 44. SYSLST Output for Multiprogramming Sample Problems (Part 5 of 5)

Magnetic Character Readers

The first 1412/1419 Magnetic Character Reader sample problem processes 500 documents from one magnetic character reader using GET logic. The documents are read into pocket 3 in groups of approximately 50. After each group is read, the 1412/1419 Magnetic Character Reader is disengaged, the batch number is updated, and the pocket light is turned on. All documents are listed on the printer. If the Selective Tape List feature is present on the printer, the documents are listed on the leftmost tape. The controls on the magnetic character reader that must be depressed, if present, are: BATCH NUMBER ON, PROG SORT, and at least one field for the VALIDITY CHECK & READ OUT control.

The second 1412/1419 Magnetic Character Reader sample problem processes 250 documents from each of two magnetic character readers, (both of the same type: both with a single address adapter or both with a dual address adapter), using READ, CHECK, and WAITF logic. The only controls that must be depressed are PROG SORT and at least one field for the VALIDITY CHECK, & READ OUT control. Documents from one reader are selected into pockets one or zero depending on whether or not the selected field is present. Documents read in error are rejected and all data is listed on a printer. If the Selective Tape List feature is present on the printer, data is printed on the leftmost tape.

Documents from the second reader are selected according to a digit in the field read and printed on SYSLST. If manual intervention is required on the second reader, a message is printed on SYSLOG. The message is:

INTERVENTION REQUIRED ON FILE2.

The program names are Z.MCR1 and Z.MCR2. The 1412/1419 sample problems support the 1259.

Instructions show how to remove the following optional features:

1. Dual addressing adapter feature
2. Batch numbering
3. Pocket lights
4. Selective Tape List Feature (Printer)

The SYSLOG output is shown in Figure 17. The SYSLST output in Figure 45 contains:

- Job control cards
- External symbol dictionary
- Source program listing
- Relocation dictionary
- Linkage editor input diagnostics
- Linkage editor storage map
- Document data listings
- EOJ MCR
- // PAUSE END OF MCR SAMPLE PROBLEMS

```
// JOB MCR SAMPLE PROBLEMS
// ASSGN SYS005,X'004'
// ASSGN SYS006,X'005'
// ASSGN SYS009,X'012'
// OPTION LINK,NOXREF
// EXEC ASSEMBLY
```

```
                                              EXTERNAL SYMBOL DICTIONARY                          PAGE    1
 SYMBOL   TYPE ID  ADDR   LENGTH LD ID

          PC   01 000000 002834
IJUDZZZZ  ER   02
IJUDZZZZ  SD   03 002838 00048C
IJDFSZZZ  ER   04
IJDFSZZZ  SD   05 002CE8 00002B
```

```
$477    MCR1 SAMPLE PROBLEM                                                                        PAGE    1

  LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                               DOS CL3-0 03/12/69

                                        2            PRINT NOGEN                                    L4770002
000000 0540                             3 START      BALR 4,0                                       L4770003
000002                                  4            USING *,4                                      L4770004
                                        5            OPEN  FILE1,FILE         OPEN ALL FILES        L4770005
000016 5850 42AE              002B0    14            L     DOCNT,TOTDOC       INITIALIZE DOCUMENT COUNTER L4770006
                                       15 NEXT       GET   FILE               GET DOCUMENT FROM MICR DEVICE L4770007
000026 9101 42AC        002AE          20            TM    DISSW,X'01'        DISENGAGE             L4770008
00002A 4710 4066              00068    21            BO    DISENG             YES                   L4770009
00002E 9110 2000        00000          22 DISCOM     TM    0(2),X'10'         READY FOR POCKET LIGHT L4770010
000032 4710 407E              00080    23            BO    LITE               YES                   L4770011
000036 9108 2000        00000          24            TM    0(2),X'08'    IF INT. REQD., RE-ISSUE GET L4770012
00003A 4710 4018              0001A    25            BO    NEXT                                     L4770013
00003E 4650 4044              00046    26 LASDOC     BCT   DOCNT,MVC          DECREMENT DOCUMENT COUNTER L4770014
000042 47F0 4092              00094    27            B     EOF                                      L4770015
000046 D209 422E 2C56 00230 00056      28 MVC        MVC   AREA1(10),86(2)    MOVE DATA TO OUTPUT AREA L4770016
                                       29            PUT   FILE1,STLSP=BKT    PRINT                 L4770017
                                       37 * OMIT 'STLSP=BKT' PARAMETER IF SELECTIVE TAPE LIST FEATURE NOT PRESENT L4770018
000064 47F0 4018              0001A    38            B     NEXT                                     L4770019
                                       39 DISENG     DISEN FILE               DISENGAGE MICR DEVICE L4770020
000074 9230 42A9        002AB          44            MVI   CTR+1,X'30'        REINITIALIZE DOCUMENT COUNTER L4770021
000078 9200 42AC        002AE          45            MVI   DISSW,X'00'        RESET DISENGAGE SWITCH L4770022
00007C 47F0 402C              0002E    46            B     DISCOM                                   L4770023
000080                                 47 LITE       EQU   *                                        L4770024
                                       48            LITE  FILE,SWAREA        LIGHT POCKET SPECIFIED IN SWAREA L4770025
                                       54 *          REMOVE ABOVE STATEMENT IF POCKET LIGHT FEATURE IS ABSENT L4770026
000090 47F0 403C              0003E    55            B     LASDOC                                   L4770027
000094 D201 4032 409C 00034 0009E      56 EOF        MVC   DISCOM+6(2),EOJADD ALTER EOJ BRANCH      L4770028
00009A 47F0 4066              00068    57            B     DISENG                                   L4770029
00009E 409E                            58 EOJADD     DC    S(EOJ)             EOJ ADCON             L4770030
                                       59 EOJ        CLOSE FILE1,FILE         CLOSE ALL FILES       L4770031
                                       68            EOJ                                            L4770032
                                       71 *                                                         L4770033
                                       72 *          STACKER SELECTION ROUTINE                      L4770034
                                       73 *                                                         L4770035
0000B4                                 74            USING *,5                                      L4770036
0000B4 923F 7004        00004          75 USSR       MVI   4(7),X'3F'         ARBITRARILY SELECT POCKET THREE L4770037
0000B8 4320 51F7              002AB    76            IC    2,CTR+1            DECREMENT DOCUMENT COUNTER L4770038
0000BC 4620 5014              000C8    77            BCT   2,NODIS                                  L4770039
0000C0 9601 51FA        002AE          78            OI    DISSW,X'01'        TURN ON DISENGAGE SWITCH L4770040
0000C4 9680 7001        00001          79            OI    1(7),X'80'         UPDATE BATCH NUMBERER L4770041
                                       80 *          REMOVE ABOVE STATEMENT IF BATCH NUMBERING NOT AVAILABLE L4770042
0000C8 4220 51F7              002AB    81 NODIS      STC   2,CTR+1                                  L4770043
                                       82            EXIT  MR                                       L4770044
                                       85            DROP  5                                        L4770045
                                       86 *                                                         L4770046
                                       87 FILE       DTFMR BUFFERS=100,                            XL4770047
                                                           EXTADDR=USSR,                           XL4770048
                                                           IOAREA1=AREA,                           XL4770049
                                                           DEVADDR=SYS005,                         XL4770050
                                                           RECSIZE=80,                             XL4770051
                                                           IOREG=2,                                XL4770052
```

Figure 45. SYSLST Output for 1412/1419 Sample Problems (Part 1 of 7)

```
$477   MCR1 SAMPLE PROBLEM                                                                                   PAGE    2

 LOC   OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          DOS CL3-0 03/12/69

                                                          ADDRESS=DUAL,      REMOVE IF DUAL ADAPTER NOT PRESENT   XL4770053
                                                          SECADDR=SYS006,    REMOVE IF DUAL ADAPTER NOT PRESENT   XL4770054
                                                          ADDAREA=10                                               L4770055
                                     157           MRMOD  ADDRESS=DUAL,BUFFERS=100                                 L4770056
                                     521 *         IF DUAL ADDRESSING ADAPTER FEATURE IS ABSENT,                   L4770057
                                     522 *         REMOVE (ADDRESS=DUAL) PARAMETER FROM THE ABOVE STATEMENT.       L4770058
                                     523 FILE1     DTFPR  DEVADDR=SYS009,                                         XL4770059
                                                          STLIST=YES, REMOVE IF NO SELECTIVE TAPE FEATURE PRESENT XL4770060
                                                          IOAREA1=AREA1                                            L4770061
                                     545           PRMOD                                                          XL4770062
                                                          STLIST=YES, REMOVE IF NO SELECTIVE TAPE FEATURE PRESENT XL4770063
                                                          RECFORM=FIXUNB                                           L4770064
000210                               613           LTORG                                                           L4770065
000210 5B5BC2D6D7C5D540              614                  =C'$$BOPEN '
000218 5B5BC2C3D3D6E2C5              615                  =C'$$BCLOSE'
000220 000000DC                      616                  =A(FILE)
000224 000001D8                      617                  =A(FILE1)
000228 000002A9                      618                  =A(BKT)
00022C 000002AC                      619                  =A(SWAREA)
000005                               620 DOCNT     EQU    5                  DOCUMENT COUNTER                      L4770066
000230 4040404040404040              621 AREA1     DC     121CL1' '          OUTPUT AREA                           L4770067
0002A9 01                            622 BKT       DC     X'01'              SELECTIVE TAPE LISTER TAPE INDICATOR  L4770068
0002AA 0030                          623 CTR       DC     H'48'              DOCUMENT COUNTER                      L4770069
0002AC 0400                          624 SWAREA    DC     X'0400'            POCKET SELECTION FOR POCKET LIGHTING  L4770070
0002AE 00                            625 DISSW     DC     X'00'              DISENGAGE SWITCH                      L4770071
0002AF 00
0002B0 000001F4                      626 TOTDOC    DC     F'500'             TOTAL NO. OF DOCUMENTS TO BE READ     L4770072
0002B4                               627 AREA      DS     CL9600             MICR INPUT AREA                       L4770073
000000                               628    END      START                                                         L4770074
```

```
                                        RELOCATION DICTIONARY                                    PAGE    1

 POS.ID    REL.ID    FLAGS    ADDRESS

   01        01       0C      00000C
   01        01       0C      000010
   01        01       0C      0000A8
   01        01       0C      0000AC
   01        01       0C      0000D8
   01        02       18      0000E1
   01        01       0C      000108
   01        01       08      000129
   01        01       08      00012D
   01        01       0C      000130
   01        01       0C      000134
   01        01       0C      000138
   01        01       0C      000144
   01        01       0C      00014C
   01        01       0C      000150
   01        01       0C      00015C
   01        01       08      000171
   01        01       08      000179
   01        01       08      000181
   01        01       08      000189
   01        01       08      000191
   01        01       08      000199
   01        01       08      0001A1
   01        01       08      0001A9
   01        01       08      0001B1
   01        01       08      0001B9
   01        01       08      0001C1
   01        01       08      0001C9
   01        01       08      0001D1
   03        03       08      002CA1
   03        03       08      002CA9
   01        01       0C      0001E0
   01        04       18      0001E9
   01        01       0C      0001F0
   01        01       08      000201
   01        01       08      000209
   01        01       0C      000220
   01        01       0C      000224
   01        01       0C      000228
   01        01       0C      00022C
```

Figure 45. SYSLST Output for 1412/1419 Sample Problems (Part 2 of 7)

```
// EXEC LNKEDT
```

```
JOB  MCR                    DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN     MAP
LIST   ENTRY
```

```
06/19/68   PHASE   XFR-AD  LOCORE  HICORE  DSK-AD   ESD TYPE  LABEL     LOADED  REL-FR

        PHASE***   004000  004000  006C9F  15 04 3  CSECT               004C00  004000

                                                    CSECT     IJUDZZZZ  006830  004000

                                                    CSECT     IJDFSZZZ  006C78  004000
```

```
// EXEC
```

```
000009437$
005298816$
000002642$
000009437$
000009437$
005298816$
000009437$
000009437$
000002642$
000002642$
000002642$
000009437$
000002642$
000002642$
000002642$
000002642$
000002642$
000002642$

000002642$
000002642$
000009437$
000002642$
```

NOTE: This is a portion of the tape output for MCR1.

```
000009437$
005298816$
000009437$
000002642$
000002642$
000002642$
000002642$
000002642$
000002642$
000002642$
000002642$
000002642$
000002642$
000009437$
```

Figure 45. SYSLST Output for 1412/1419 Sample Problems (Part 3 of 7)

```
// PAUSE   END OF MCR1
// ASSGN SYS007,X'006'
// EXEC ASSEMBLY
```

```
                                              EXTERNAL SYMBOL DICTIONARY                                         PAGE    1
  SYMBOL   TYPE ID   ADDR   LENGTH LD ID

           PC   01 000000 003928
IJUDZZZZ   ER   02
IJDFSZZZ   ER   03
IJDFZZZZ   ER   04
IJUDZZZZ   SD   05 003928 0004B0
IJDFSZZZ   SD   06 003DD8 000028
IJDFZZZZ   SD   07 003E00 000028
IJ2M0026   SD   08 003E28 00001A
```

```
$477    MCR2 SAMPLE PROBLEM                                                                                    PAGE    1

  LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                            DOS CL3-0 03/12/69

                                        2            PRINT NOGEN                                                 $4770002
000000 0540                             3 START      BALR  4,0                                                   $4770003
000002                                  4            USING *,4                                                   $4770004
                                        5            OPEN  FILE1,FILE2,FILE3,FILE4 OPEN ALL FILES                $4770005
00001F 5850 447A             0047C     16            L     DOCNT1,TOTDOC      INITIALIZE FILE1 DOCUMENT COUNTER  $4770006
000022 1865                            17            LR    DOCNT2,DOCNT1      INITIALIZE FILE2 DOCUMENT COUNTER  $4770007
                                       18 NEXT       READ  FILE1,MR            GET DOCUMENT FROM FIRST MICR DEVICE  $4770008
                                       23            CHECK FILE1,NEXT1        IS DOCUMENT READY FOR PROCESSING   $4770009
000040 4650 404E             00050     29            BCT   DOCNT1,MVC1        DECREMENT DOCUMENT COUNTER FOR FILE1 $4770010
                                       30            DISEN FILE1              DISENGAGE FILE1                    $4770011
000050 D209 447E 2010 00480 00010      35 MVC1       MVC   AREA(10),16(2)     MOVE DATA TO OUTPUT AREA           $4770012
                                       36            PUT   FILE3,STLSP=CNTAD1 PRINT                              $4770013
                                       44 * OMIT 'STLSP=CNTAD1' PARAMETER IF NO SELECTIVE TAPE FEATURE           $4770014
                                       45 NEXT1      READ  FILE2,MR           GET DOCUMENT FROM SECOND MICR DEVICE $4770015
                                       50            CHECK FILE2,NEXT2        IS DOCUMENT READY FOR PROCESSING   $4770016
00008A 4660 4098             0009A     56            BCT   DOCNT2,MVC2        DECREMENT DOCUMENT COUNTER FOR FILE2 $4770017
                                       57            DISEN FILE2              DISENGAGE FILE2                    $4770018
00009A D263 45CA 3006 0050C 00006      62 MVC2       MVC   PRNTAR(100),6(3)   MOVE DATA TO OUTPUT AREA           $4770019
                                       63            PUT   FILE4              PRINT                              $4770020
                                       68 NEXT2      WAITF FILE1,FILE2        ANY MICR FILES ACTIVE              $4770021
0000C0 47F0 4022             00024     77            B     NEXT               YES                                $4770022
0000C4 18E0                            78 ERROUT1    LR    14,0               INITIALIZE RETURN REGISTER         $4770023
0000C6 9130 2000       00000           79            TM    0(2),X'30' UNRECOVERABLE I/O ERROR OR ALL DOCS PROCESSED $4770024
0000CA 4770 40EE             000F0     80            BNZ   CLOSE1             YES                                $4770025
0000CE 07FE                            81            BR    14                 RETURN TO NEXT READ                $4770026
0000D0 18A0                            82 ERROUT2    LR    RETURN,0           INITIALIZE RETURN REGISTER         $4770027
0000D2 9130 3000       00000           83            TM    0(3),X'30' UNRECOVERABLE I/O ERROR OR ALL DOCS PROCESSED $4770028
0000D6 4770 4106             00108     84            BNZ   CLOSE2             YES                                $4770029
0000DA 9108 3000       00000           85            TM    0(3),X'08'         INTERVENTION REQUIRED              $4770030
0000DE 07EA                            86            BCR   14,RETURN          NO                                 $4770031
                                       87            PUT   MSGWRT                                                $4770032
0000EC 07FA                            92            BR    RETURN                                                $4770033
                                       93 CLOSE1     CLOSE FILE1              CLOSE FILE1                        $4770034
0000FE 9601 446F       00471          101            OI    CLOSID,X'01'       SET FILE1 CLOSED INDICATOR         $4770035
000102 47F0 4118             0011A    102            B     ENDTST                                                $4770036
                                      103 CLOSE2     CLOSE FILE2              CLOSE FILE2                        $4770037
000116 9602 446F       00471          111            OI    CLOSID,X'02'       SET FILE2 CLOSED INDICATOR         $4770038
00011A 9103 446F       00471          112 ENDTST     TM    CLOSID,X'03'       BOTH FILES CLOSED                  $4770039
00011E 47E0 4022             00024    113            BNO   NEXT               NO                                 $4770040
                                      114            CLOSE FILE3,FILE4        CLOSE PRINTER FILES                $4770041
                                      123            EOJ                                                         $4770042
                                      126 *                                                                      $4770043
                                      127 *          STACKER SELECTION ROUTINE FOR FILE1                         $4770044
                                      128 *                                                                      $4770045
000138                                129            USING *,5                                                   $4770046
000138 910C 7002       00002          130 USSR1      TM    2(7),X'0C'         DATA CHECK OR OVERRUN              $4770047
00013C 4770 5022             0015A    131            BNZ   DERR1              YES                                $4770048
000140 D509 533A 7010 00472 00010     132            CLC   BLANK(10),16(7)    IS FIELD BLANK                     $4770049
000146 4770 501A             00152    133            BNE   PKT1               NO                                 $4770050
00014A 920F 7004       00004          134            MVI   4(7),X'0F'         SELECT POCKET ZERO                 $4770051
00014E 47F0 5026             0015E    135            B     EXIT1                                                $4770052
```

Figure 45. SYSLST Output for 1412/1419 Sample Problems (Part 4 of 7)

```
$477   MCR2 SAMPLE PROBLEM                                                                              PAGE    2

  LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                     DOS CL3-0 03/12/69

000152 921F 7004        00004         136 PKT1      MVI   4(7),X'1F'           SELECT POCKET ONE                $4770053
000156 47F0 5026              0015E   137           B     EXIT1                                                 $4770054
00015A 92CF 7004        00004         138 DERR1     MVI   4(7),X'CF'           SELECT REJECT POCKET             $4770055
                                      139 EXIT1     EXIT  MR                                                    $4770056
                                      142 *                                                                     $4770057
                                      143 *         STACKER SELECTION ROUTINE FOR FILE2                         $4770058
                                      144 *                                                                     $4770059
000160                                145           USING *,5                                                   $4770060
000160 910C 7002        00002         146 USSR2     TM    2(7),X'CC'           DATA CHECK OR OVERRUN            $4770061
000164 4770 5022              00182   147           BNZ   REJECT               YES                              $4770062
000168 95F9 7062        00062         148           CLI   98(7),X'F9'                                           $4770063
00016C 4720 5022              00182   149           BH    REJECT               IS DATA A DIGIT                  $4770064
000170 95F0 7062        00062         150           CLI   98(7),X'FC'                                           $4770065
000174 4740 5022              00182   151           BL    REJECT                                                $4770066
000178 F200 7004 7062 00004 00062     152           PACK  4(1,7),98(1,7)       SELECT POCKET ACCORDING DIGIT ON DOC. $4770067
00017E 47F0 5026              00186   153           B     EXIT2                                                 $4770068
000182 92CF 7004        00004         154 REJECT    MVI   4(7),X'CF'           SELECT REJECT POCKET             $4770069
                                      155 EXIT2     EXIT  MR                                                    $4770070
                                      158           DROP  5                                                     $4770071
                                      159 *                                                                     $4770072
                                      160 FILE1     DTFMR DEVADDR=SYS005,                                      X$4770073
                                                          ADDRESS=DUAL, REMOVE IF DUAL ADDRESSING ADAPTER ABSENT X$4770074
                                                          ADDAREA=0,                                           X$4770075
                                                          IOREG=2,                                             X$4770076
                                                          IOAREA1=AREA1,                                       X$4770077
                                                          RECSIZE=20,                                          X$4770078
                                                          ERROPT=ERROUT1,                                      X$4770079
                                                          EXTADDR=USSR1,                                       X$4770080
                                                          SECADDR=SYS006, REMOVE IF DUAL ADDRESS ADAPTER ABSENT C$4770081
                                                          BUFFERS=100                                           $4770082
                                      230 FILE2     DTFMR DEVADDR=SYS007,                                      X$4770083
                                                          ADDRESS=DUAL, REMOVE IF DUAL ADDRESSING ADAPTER ABSENT X$4770084
                                                          RECSIZE=80,                                          X$4770085
                                                          IOAREA1=AREA2,                                       X$4770086
                                                          IOREG=3,                                             X$4770087
                                                          ADDAREA=20,                                          X$4770088
                                                          BUFFERS=100,                                         X$4770089
                                                          ERROPT=ERROUT2,                                      X$4770090
                                                          SECADDR=SYS008, REMOVE IF DUAL ADDRESS ADAPTER ABSENT C$4770091
                                                          EXTADDR=USSR2                                         $4770092
                                      300 FILE3     DTFPR DEVADDR=SYS009,                                      X$4770093
                                                          STLIST=YES, REMOVE IF NO SELECTIVE TAPE FEATURE PRESENT X$4770094
                                                          IOAREA1=AREA                                          $4770095
                                      322 FILE4     DTFPR DEVADDR=SYSLST,                                      X$4770096
                                                          IOAREA1=PRNTAR                                        $4770097
                                      343           MRMOD ADDRESS=DUAL,BUFFERS=100                              $4770098
                                      707 *         REMOVE (ADDRESS=DUAL) PARAMETER FROM THE ABOVE STATEMENT.   $4770099
                                      708 *         IF DUAL ADDRESSING ADAPTER FEATURE IS ABSENT,               $4770100
                                      709           PRMOD                                                      X$4770101
                                                          STLIST=YES,       REMOVE IF NO SELECTIVE TAPE FEATURE X$4770102
                                                          RECFORM=FIXUNB                                        $4770103
```

```
$477   MCR2 SAMPLE PROBLEM                                                                              PAGE    3

  LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                     DOS CL3-0 03/12/69

                                      777           PRMOD RECFORM=FIXUNB                                        $4770104
                                      844 MSGWRT    DTFCN BLKSIZE=32,DEVADDR=SYSLOG,IOAREA1=MESSG,TYPEFLE=OUTPUT $4770105
000420                                888           LTORG                                                       $4770106
000420 5B5BC2D6D7C5D540               889                 =C'$$BOPEN '
000428 5B5BC2C3D3D6E2C5               890                 =C'$$BCLOSE'
000430 00000188                       891                 =A(FILE1)
000434 0000006E                       892                 =A(NEXT1)
000438 00000398                       893                 =A(FILE3)
00043C 00000470                       894                 =A(CNTAD1)
000440 00000290                       895                 =A(FILE2)
000444 000000AC                       896                 =A(NEXT2)
000448 000003D0                       897                 =A(FILE4)
00044C 00000400                       898                 =A(MSGWRT)
000005                                899 DOCNT1    EQU   5                    DOCUMENT COUNTER FOR FILE1       $4770107
000006                                900 DOCNT2    EQU   6                    DOCUMENT COUNTER FOR FILE2       $4770108
00000A                                901 RETURN    EQU   10                                                    $4770109
000450 40C9D5E3C5D9E5C5               902 MESSG     DC    C' INTERVENTION REQUIRED ON FILE2 '                   $4770110
000470 01                             903 CNTAD1    DC    X'01'                SELECTIVE TAPE LIST TAPE INDICATOR $4770111
000471 00                             904 CLOSID    DC    X'00'                CLOSED INDICATOR                 $4770112
000472 00000000000000000              905 BLANK     DC    10X'00'                                               $4770113
00047C 000000FA                       906 TOTDOC    DC    F'250'               TOTAL NO. OF DOCUMENTS TO BE READ $4770114
000480 4040404040404040               907 AREA      DC    140C' '              OUTPUT AREA FOR FILE3            $4770115
00050C 4040404040404040               908 PRNTAR    DC    140C' '              OUTPUT AREA FOR FILE4            $4770116
000598                                909 AREA1     DS    CL2600               INPUT AREA FOR FILE1             $4770117
000FC0                                910 AREA2     DS    CL10600              INPUT AREA FOR FILE2             $4770118
000000                                911           END   START                                                 $4770119
```

Figure 45. SYSLST Output for 1412/1419 Sample Problems (Part 5 of 7)

RELOCATION DICTIONARY PAGE 1

POS.ID	REL.ID	FLAGS	ADDRESS
01	01	0C	00000C
01	01	0C	000010
01	01	0C	000014
01	01	0C	000018
01	01	0C	0000B8
01	01	0C	0000BC
01	01	0C	0000F8
01	01	0C	00011C
01	01	0C	00012C
01	01	0C	00013C
01	01	0C	00019C
01	02	18	000199
01	01	0C	0001C0
01	01	08	0001F1
01	01	08	0001E5
01	01	0C	0001F8
01	01	0C	0001EC
01	01	0C	0001F0
01	01	0C	0001FC
01	01	0C	000204
01	[illegible]	0C	000208
01	01	08	[illegible]
01	01	08	0002ED
01	01	0C	0002F0
01	01	0C	0002F4
01	01	0C	0002F8
01	01	0C	000304
01	01	0C	00030C
01	01	0C	00031C
01	01	0C	00031C
01	01	08	000331
01	01	08	000339
01	01	08	000341
01	01	08	000349

RELOCATION DICTIONARY PAGE 2

POS.ID	REL.ID	FLAGS	ADDRESS
01	01	08	000351
01	01	08	000359
01	01	08	000361
01	01	08	000369
01	01	08	000371
01	01	08	000379
01	01	08	000381
01	01	08	000389
01	01	08	000391
01	01	0C	0003A0
01	03	18	0003A9
01	01	CC	0003BC
01	01	08	0003C1
01	01	08	0003C9
01	01	0C	0003D8
01	04	18	0003E1
01	01	0C	0003E8
01	01	08	0003F9
05	05	08	003D91
05	05	08	003D99
01	01	0C	000408
01	08	08	000411
01	01	08	000419
01	01	0C	000430
01	01	0C	000434
01	01	CC	000438
01	01	0C	00043C
01	01	0C	000440
01	01	0C	000444
01	01	0C	000448
01	01	0C	00044C

Figure 45. SYSLST Output for 1412/1419 Sample Problems (Part 6 of 7)

```
// EXEC LNKEDT
```

```
JOB  MCR                DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN    MAP
LIST   ENTRY
```

```
        PHASE   XFR-AD  LOCORE  HICORE  DSK-AD   ESD TYPE  LABEL     LOADED  REL-FR

     PHASE***   004000  004000  007DD9  15 06 2  CSECT               004000  004000

                                                 CSECT     IJUDZZZZ  007928  004000

                                                 CSECT     IJDFSZZZ  007D70  004000

                                                 CSECT     IJDFZZZZ  007D98  004000

                                                 CSECT     IJ2M0026  007DC0  004000
```

```
// EXEC
                                                              )29010007)55458,3210$0001334469$
                                                              )29010007)55917,3210$0001334469$
                                                          )28361407)5283691407,5283$5283691407$
                                                          )28361407)5283691407,5283$5283691407$
                                                          )28361407)5283691407,5283$5283691407$
                                                          )28361407)5283691407,5283$5283691407$
                                                          )28361407)5283691407,5283$5283691407$
                                                          )28361407)5283691407,5283$5283691407$
                                                          )28361407)5283691407,5283$5283691407$
                                                          )28361407)5283691407,5283$5283691407$
                                                           )28361407)5283691407,52835283691407$
                                                           )28361407)528391407,5283$5283691407$
                                                          )28361407)5283691407,5283$5283691407$
                                                          )12346789)0123456789,0123$0123456789$
                                                          )28361407)5283691407,5283$5283691407$
                                                          )28361407)5283691407,5283$5283691407$
                                                          )28361407)5283691407,5283$5283691407$
                                                          )28361407)5283691407,5283$5283691407$
                                                          )28361407)5283691407,5283$5283691407$
                                                           )28361407)528369147,5283$5283691407$
                                                           )2361407)5283691407,5283$5283691407$
                                                          )28361407)5283691407,5283$5283691407$
                                                          )28361407)5283691407,5283$5283691407$
                                                          )06131009)0000061757,8801$0000013104$
                                                          )05020991)0000091132,8031$0012000000$
                                                          )05020991)0000091132,8031$0012000000$
                                                          )05020991)0000091132,8031$0012000000$
                                                          )05020991)0000091132,8031$0012000000$
                                                          )05020991)0000091132,8031$0012000000$
                                                          )05020991)0000091132,8031$0012000000$
                                                          )05020991)0000091132,8031$0012000000$
                                                          )05020991)0000091132,8031$0012000000$
                                                          )05020991)0000091132,8031$0012000000$
```

```
EOJ MCR
```

```
// PAUSE   END OF MCR SAMPLE PROBLEMS
```

Figure 45. SYSLST Output for 1412/1419 Sample Problems (Part 7 of 7)

Optical Readers

The 1285 sample problem is obtained by retrieving the sample problem for the 1287 journal tape mode of operation (Z.ORJT) and changing the cards with the sequence numbers 1058 and 1077 as follows. For both cards change the parameter DEVICE=1287T to DEVICE=1285. (Refer to listing containing the source statements for the sample test program for journal tape processing.)

The sample problem illustrating document processing for the 1287 Optical Reader (Z.ORDC) reads documents for data input into an input area, and then prints this data on SYSLST. Whenever the 1287 document hopper empties, press end-of-file or replenish the stack.

Any approved document type may be used because the reference mark and data field coordinates are entered at program execution time. However, the data field chosen to be read may not exceed six characters in length.

Figure 19 is a sample input document, and Figure 46 is the SYSLST output listing for document mode processing. The listing contains examples of keyed-in error corrections identified by comments.

The sample problem illustrating journal tape mode processing for the 1287 or 1285 Optical Readers reads undefined records from the 1287 or 1285 Optical Reader into a work area from two I/O areas, and then prints these records on SYSLST.

Any journal tape with a maximum record size of 38 characters is suitable for this sample problem.

Figure 20 is a portion of a sample input journal tape, and Figure 47 is the SYSLST output listing for journal tape processing.

The SYSLST output in Figures 46 and 47 contain:

- Job control cards
- External symbol dictionary
- Source program listing
- Relocation dictionary
- Linkage editor input diagnostics
- Linkage editor storage map
- List of fields read from documents (if sample problem for document processing was used).
  <u>or</u>
  Journal tape listing (If sample problem for journal tape processing was used.)
- EOJ Optical Reader
- // PAUSE END OF SAMPLE PROBLEM

```
// JOB OPTICAL READER SAMPLE PROBLEM
// ASSGN SYS009,X'055'
// OPTION LINK,NOXREF
// EXEC ASSEMBLY
```

```
                                              EXTERNAL SYMBOL DICTIONARY
 SYMBOL   TYPE ID  ADDR  LENGTH LD ID

SAMPLE     SD  01 000000 000600
IJMFCZDZ   ER  02
IJMFCZDZ   SD  03 000600 000600
IJMFZZDZ   LD     000600           03
IJDFZZZZ   ER  04
IJDFZZZZ   SD  05 000000 00001A
```

```
          SAMPLE TEST PROGRAM FOR DOCUMENT PROCESSING

  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                     DOS CL3-0

                                      2          PRINT NOGEN                                         $4780002
000000                                3 SAMPLE   START                                               $4780003
000000 0530                           4 BASERG   BALR  3,0                                           $4780004
000002                                5          USING *,R3                                          $4780005
                                      6          OPEN  ORDTF,PRDTF            OPEN ORFILE,PRINTFILE  $4780006
000016 9240 3472      00474          15          MVI   DATAPRT,C' '           CLEAR                  $4780007
00001A D277 3473 3472 00475 00474    16          MVC   DATAPRT+1(120),DATAPRT      OUTPUT AREA       $4780008
                                     17          PUT   CNDTF                  WRITE MSG ON SYSLOG    $4780009
                                     22          GET   CNDTF                  READ 16 CHARS INPUT    $4780010
000038 58B0 3242            00244    27          L     R11,=F'4'              LOAD R11 WITH LOOP CNTR $4780011
00003C 0700                          28          CNOP  2,4                                           $4780012
00003E D203 3456 33CE 00458 003D0    29 ALTER1   MVC   BEGIN(4),CALBRTNS      MOVE 4 BYTES AT A TIME $4780013
000044 58A0 303E            00040    30          L     R10,ALTER1+2           INCR ABOVE DISPLACE BY $4780014
000048 5AA0 3242            00244    31          A     R10,=F'4'              FOUR BYTES AND STORE   $4780015
00004C 50A0 303E            00040    32          ST    R10,ALTER1+2           BACK                   $4780016
000050 DC03 3456 335C 00458 0035E    33          TR    BEGIN,TABLE0-193       CONVERT 4-BYTE         $4780017
000056 F233 345A 3456 0045C 00458    34          PACK  TO(4),BEGIN(4)         UNPACKED FIELD TO TWO  $4780018
00005C 5820 345A            0045C    35          L     R2,TO                  BYTE-LEFT-ADJUSTED     $4780019
000060 8820 0004            00004    36          SRL   R2,4                   PACKED FIELD           $4780020
000064 8920 0010            00010    37          SLL   R2,16                  IN R2                  $4780021
000068 5020 3462            00464    38          ST    R2,SAVE                STORE 2 PACKED BYTES IN SAVE $4780022
00006C 0700                          39          CNOP  2,4                                           $4780023
00006E D201 3226 3462 00228 00464    40 ALTER2   MVC   LNDMRK1(2),SAVE        MOVE TWO BYTES INTO COORDNTS $4780024
000074 58A0 306E            00070    41          L     R10,ALTER2+2           INCR DISPLACEMENT OF FILDMRK1 $4780025
000078 5AA0 345E            00460    42          A     R10,CONST              BY TWO AND THEN        $4780026
00007C 50A0 306F            00070    43          ST    R10,ALTER2+2           RESTORE INTO INSTRUCTION $4780027
000080•46B0 303C            0003E    44          BCT   R11,ALTER1             LOOP BACK TO ALTER1    $4780028

                                     46 *                                                            $4780030
                                     47 *                                                            $4780031
                                     48 READDOCS READ  ORDTF,OR,CCWADR        READ DOCUMENTS         $4780032
                                     54          WAITF ORDTF                  WAIT I/O               $4780033
0000A0 D205 3473 34EB 00475 004ED    59          MVC   DATAPRT+1(6),INAREA    MOVE OR DATA TO PRINT AREA $4780034
                                     60          CNTRL ORDTF,ESD,1            EJECT AND SS DOC POK 1 $4780035
                                     66          PUT   PRDTF                  PRINT LINE             $4780036
0000C0 9240 34EB      004ED          71          MVI   INAREA,C' '            CLEAR                  $4780037
0000C4 D204 34EC 34EB 004EE 004ED    72          MVC   INAREA+1(5),INAREA          INPUT AREA        $4780038
0000CA 9240 3472      00474          73          MVI   DATAPRT,C' '           CLEAR                  $4780039
0000CE D277 3473 3472 00475 00474    74          MVC   DATAPRT+1(120),DATAPRT      OUTPUT AREA       $4780040
0000D4 47F0 3082            00084    75          B     READDOCS               GO TO READ NEXT DOCUMENT $4780041
                                     76 *                                                            $4780042
                                     77 *                                                            $4780043


0000D8 90E1 3462            00464    79 CORRECT  STM   R14,R1,SAVE            SAVE REGS              $4780045


0000DC 9101 3546      00548          81 DATACK   TM    ORDTF80,X'01'          Q- DATA CHECK          $4780047
```

Figure 46. SYSLST Output for Optical Reader Sample Problem -- Document Mode Processing (Part 1 of 5)

```
          SAMPLE TEST PROGRAM FOR DOCUMENT PROCESSING

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  DOS CL3-0

0000E0 4780 30EC             000EE    82            BZ     WRLNGTH              NO, BR TO WRONG LNTH REC TEST     $4780048
0000E4 D22E 3486 3370 00488 00372    83            MVC    DATAPRP(47),DATAMSG  YES, MOVE DATA CK MSG             $4780049
0000EA 47F0 3180             00182    84            B      RESCAN               BR TO RESCAN ROUTINE              $4780050

0000EE 9104 3546       00548          86 WRLNGTH   TM     ORDTF80,X'04'        Q- WAS IT A WRONG LNGTH RECORD    $4780052
0000F2 4780 30FE             00100    87            BZ     EQUIPC               NO, GO TEST EQUIP CHECK           $4780053
0000F6 D22E 3486 3312 00488 00314    88            MVC    DATAPRP(47),WLENMSG  YES, MOVE WRONG LNGTH REC MSG     $4780054
0000FC 47F0 31FA             001FC    89            B      ENDIPT               TERMINATE PGM                     $4780055

000100 9108 3546       00548          91 EQUIPC    TM     ORDTF80,X'08'        Q- EQUIPCHECK                     $4780057
000104 4780 3110             00112    92            BZ     NONRECOV             NO, BR TO TEST NONRECOVERY        $4780058
000108 D22E 3486 339F 00488 003A1    93            MVC    DATAPRP(47),EQUIPMSG YES, MOVE EQUIP CK MSG            $4780059
00010E 47F0 31CE             001D0    94            B      DISPLAY              BR TO DISPLAY ROUTINE             $4780060

000112 9110 3546       00548          96 NONRECOV  TM     ORDTF80,X'10'        Q- NON RECOV                      $4780062
000116 4780 312E             00130    97            BZ     LATESS               NO,BR TO LATE SS TEST             $4780063
00011A D22E 3486 3256 00488 00258    98            MVC    DATAPRP(47),NONREC   YES,MOVE NONREC MSG               $4780064
                                      99            PUT    PRDTF                PRINT                             $4780065
00012C 47F0 31FA             001FC   104            B      ENDIPT               TERMINATE PROGRAM                 $4780066

000130 9120 3546       00548         106 LATESS    TM     ORDTF80,X'20'        Q- LATE STACKER SELECT            $4780068
000134 4780 314C             0014E   107            BZ     BADREFMK             NO, BR TO LOST REF MRK TEST       $4780069
000138 D22E 3486 3285 00488 00287   108            MVC    DATAPRP(47),LATESTAK YES,MOVE LATE STAK SEL MSG        $4780070
                                     109            PUT    PRDTF                PRINT                             $4780071
00014A 47F0 31FA             001FC   114            B      ENDIPT               TERM--HARDWARE ERROR OCCURRED     $4780072

00014E 9140 3546       00548         116 BADREFMK  TM     ORDTF80,X'40'        Q- BAD LD FRMT OF REF MARK        $4780074
000152 4780 316A             0016C   117            BZ     ILLBIT               NO, ILLGL BIT WAS SET             $4780075
000156 D22E 3486 32B4 00488 002B6   118            MVC    DATAPRP(47),RECALIBR YES, MOVE RECALIBRATE MSG         $4780076
                                     119            PUT    PRDTF                PRINT                             $4780077
000168 47F0 31FA             001FC   124            B      ENDIPT               TERM PGM                          $4780078

00016C D22E 3486 32E3 00488 002E5   126 ILLBIT    MVC    DATAPRP(47),BADSENSE MOVE MSG FOR ILL BIT SETTING      $4780080
                                     127            PUT    PRDTF                PRINT                             $4780081
00017E 47F0 31FA             001FC   132            B      ENDIPT               TERM PGM                          $4780082


000182 4180 320E             00210   134 RESCAN    LA     R11,CCWADR           LOAD ADR OF COORD OF LNDMRK       $4780084
000186 58A0 3516             00518   135            L      R10,ORDTF+32         LOAD ADR OF RDBKWRD CCW           $4780085
00018A 5BA0 3252             00254   136            S      R10,=F'8'            POINT TO PRECEDING LD FORMAT      $4780086

                                     138 *          REREAD THE FIELD FIVE MORE TIMES AND FORCE ON-LINE CORRECTION $4780088
                                     139 *          IF NOT SUCCESSFUL                                             $4780089
```

```
          SAMPLE TEST PROGRAM FOR DOCUMENT PROCESSING

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  DOS CL3-0

                                     141            RESCN ORDTF,(10),(11),5,F                                     $4780091


0001AA 9108 3546       00548         150            TM     ORDTF80,X'08'        Q- EQUIPMENT CHECK                $4780093
0001AE 4710 31CE             001D0   151            BO     DISPLAY              YES, GO TO DISPLAY FIELD          $4780094

0001B2 9101 3546       00548         153            TM     ORDTF80,X'01'        NO,  Q- DATA CHECK                $4780096
0001B6 4780 31BC             001BE   154            BZ     ANYERRS              NO, BR TO TEST OTHER ERRORS       $4780097
0001BA 47F0 31CE             001D0   155            B      DISPLAY              YES, GO TO DISPLAY FIELD          $4780098

0001BE 91F6 3546       00548         157 ANYERRS   TM     ORDTF80,X'F6'        Q- ANY OTHER ERRORS               $4780100
0001C2 4780 31F2             001F4   158            BZ     RETRN                NO, GO TO RETRN TO PGM            $4780101
0001C6 D22E 3486 3341 00488 00343   159            MVC    DATAPRP(47),RETRYERR YES, MOVE RETRY ERR MSG           $4780102
0001CC 47F0 31F2             001F4   160            B      RETRN                RETRN TO PGM                      $4780103


0001D0 4180 320E             00210   162 DISPLAY   LA     R11,CCWADR           LD ADR OF COORD OF LNDMRK         $4780105
0001D4 58A0 3516             00518   163            L      R10,ORDTF+32         LD ADR OF RDBKWRD CCW             $4780106
0001D8 5BA0 3252             00254   164            S      R10,=F'8'            POINT TO PRECEDING LD FORMAT      $4780107

                                     166            DSPLY ORDTF,(10),(11)       DISPLAY FIELD FOR KYBRD CORR      $4780109
```

Figure 46. SYSLST Output for Optical Reader Sample Problem -- Document Mode Processing (Part 2 of 5)

```
0001F4 98E1 3462             00464    174 RETRN     LM    R14,R1,SAVE             RESTORE REGS                       $4780111
0001F8 07FE                           175           BR    R14                     RETURN FROM COREXIT TO NORM PROC   $4780112


                                      177 ENDIPT    CLOSE ORDTF,PRDTF             CLOSE ORDTF AND PRINT DTF          $4780114
                                      186           EOJ                           JOB EXIT                           $4780115


000210 5300022860C00004               190 CCWADR    CCW   X'53',LNDMRK1,X'60',4   LANDMARK LOCATION                  $4780117
000218 5300022C60C00004               191           CCW   X'53',FILDMRK1,X'60',4  1ST FIELD LOCATION                 $4780118
000220 2C0004F220C00006               192           CCW   X'2C',INAREA+5,X'20',6  READ BKWRDS W/ DATA CK 6 BYTES     $4780119


                                      194 *                                                                          $4780121
000228                                195 LNDMRK1   DS    F                                                          $4780122
00022C                                196 FILDMRK1  DS    F                       FIELD DEF-4TH BYTE FOR ON-LINE     $4780123
                                      197 *                                                                          $4780124


000230                                199           LTORG                                                            $4780126
000230 5B5BC2D6D7C5D540               200                 =C'$$BOPEN '
000238 5B5BC2C3D3D6E2C5               201                 =C'$$BCLOSE'
```

```
           SAMPLE TEST PROGRAM FOR DOCUMENT PROCESSING

  LOC  OBJECT CODE      ADDR1 ADDR2  STMT    SOURCE STATEMENT                                            DOS CL3-0

000240 000005B0                       202                 =A(CNDTF)
000244 00000004                       203                 =F'4'
000248 000004F8                       204                 =A(ORDTF)
00024C 00000210                       205                 =A(CCWADR)
000250 00000580                       206                 =A(PRDTF)
000254 00000008                       207                 =F'8'

000258 40D5D6D560D9C5C3               209 NONREC    DC    C' NON-RECOVERY OCCURRED.RELOAD AND RESTART 1287 '     $4780128
000287 40D3C1E3C540E2E3               210 LATESTAK  DC    C' LATE STACKER SELECT. DOC IS SS IN REJECT POCK '     $4780129
0002B6 40C3D6E4D3C440D5               211 RECALIBR  DC    C' COULD NOT FIND LANDMARK- RECALIBRATE DOCUMENT '     $478013C
0002E5 40E4D5D9C5C3D6C7               212 BADSENSE  DC    C' UNRECOGNIZABLE BIT WAS FOUND IN COREXIT TESTS '     $4780131
000314 40E6D9D6D5C740D3               213 WLENMSG   DC    C' WRONG LENGTH RECORD WAS DETECTED. UNRECOV ERR.'     $4780132
000343 40C1C6E3C5D940D9               214 RETRYERR  DC    C' AFTER RESCN-AN ERR OTHER THAN DATACK,EQUIP LOG'     $4780133
000372 40C4C1E3C140C3C8               215 DATAMSG   DC    C' DATA CHECK                                    '     $4780134
0003A1 40C5D8E4C9D7D4C5               216 EQUIPMSG  DC    C' EQUIPMENT CHECK (INCOMPLETE READ)             '     $4780135
                                      217 CALBRTNS  DC    C'ENTER 16 HEX DIGITS FOR REF MARK AND DATA FIELD COORDIX$4780136
0003D0 C5D5E3C5D940F1F6                                   NATES, ALPHA- UPPER CASE.'                             $4780137
00041F FAFBFCFDFEFF                   218 TABLE0    DC    X'FAFBFCFDFEFF'                                        $4780138
000425                                219           DS    CL41                                                   $4780139
00044E F0F1F2F3F4F5F6F7               220 TABLE1    DC    C'0123456789'                                          $4780140
000458                                221 BEGIN     DS    F                                                      $4780141
00045C                                222 TO        DS    F                                                      $4780142
000460 00020000                       223 CONST     DC    X'00020000'                                            $4780143

000464                                225 SAVE      DS    4F                                                     $4780145

000474                                227 DATAPRT   DS    0CL121                                                 $4780147
000474                                228           DS    CL20                                                   $4780148
000488                                229 DATAPRP   DS    CL101                                                  $4780149

0004ED 404040404040                   231 INAREA    DC    CL6' '                                                 $4780151

                                      233 ***DTFS AND MODULES                                                      $4780153
                                      234 ORDTF     DTFOR BLKSIZE=6,                                              X$4780154
                                                          COREXIT=CORRECT,                                        X$4780155
                                                          DEVADDR=SYS009,                                         X$4780156
                                                          DEVICE=1287D,                                           X$4780157
                                                          CONTROL=YES,                                            X$4780158
                                                          EOFADDR=ENDIPT,                                         X$4780159
                                                          IOAREA1=INAREA                                           $4780160
                                      281           ORMOD CONTROL=YES,                                            X$4780161
                                                          DEVICE=1287D                                             $4780162
                                      824 PRDTF     DTFPR BLKSIZE=121,                                            X$4780163
                                                          DEVADDR=SYSLST,                                         X$4780164
                                                          IOAREA1=DATAPRT                                          $4780165
                                      845           PRMOD                                                          $4780166
                                      907 CNDTF     DTFCN DEVADDR=SYSLOG,                                         X$4780167
```

Figure 46. SYSLST Output for Optical Reader Sample Problem -- Document Mode Processing (Part 3 of 5)

```
          SAMPLE TEST PROGRAM FOR DOCUMENT PROCESSING

  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                   DOS CL3-0

                                                  IOAREA1=CALBRTNS,                          X$4780168
                                                  BLKSIZE=80                                  $4780169
000548                             940 ORDTF80  EQU   ORDTF+80                                $478017C
000001                             941 R1       EQU   1                                       $4780171
000002                             942 R2       EQU   2                                       $4780172
000003                             943 R3       EQU   3                                       $4780173
00000A                             944 R10      EQU   10                                      $4780174
00000B                             945 R11      EQU   11                                      $4780175
00000C                             946 R12      EQU   12                                      $4780176
00000E                             947 R14      EQU   14                                      $4780177

000000                             949          END   BASERG                                  $4780179
```

```
                              RELOCATION DICTIONARY

 POS.ID   REL.ID   FLAGS   ADDRESS

   01       01      0C     00000C
   01       01      0C     000010
   01       01      0C     000204
   01       01      0C     000208
   01       01      08     000211
   01       01      08     000219
   01       01      08     000221
   01       01      0C     000240
   01       01      0C     000248
   01       01      0C     00024C
   01       01      0C     000250
   01       01      08     000501
   01       01      08     000505
   01       02      18     000509
   01       01      08     000519
   01       01      08     00051D
   01       01      08     000521
   01       01      08     00054D
   01       01      08     000551
   01       01      08     000559
   01       01      08     000561
   01       01      0C     000568
   01       01      0C     00056C
   01       01      0C     000570
   01       01      0C     000588
   01       04      18     000591
   01       01      0C     000598
   01       01      08     0005A9
   01       01      0C     0005B8
   01       01      08     0005C1
   01       01      08     0005C9

 NO STATEMENTS FLAGGED IN THIS ASSEMBLY
```

```
 // EXEC LNKEDT
```

Figure 46. SYSLST Output for Optical Reader Sample Problem -- Document Mode Processing (Part 4 of 5)

```
JOB  OPTICAL              DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN     MAP
LIST    ENTRY
```

```
        PHASE   XFR-AD  LOCORE  HICORE  DSK-AD   ESD TYPE  LABEL     LOADED  REL-FR

      PHASE***  002800  002800  003419  13 5 1   CSECT     SAMPLE    002800  002800

                                                 CSECT     IJMFCZDZ  002E00  002800
                                                 * ENTRY   IJMFZZDZ  002E00

                                                 CSECT     IJDFZZZZ  003400  002800
```

```
// EXEC
 021057
 021057
 021057
 021057
 021057
 021057
 021057
 021057
 021057
 021057
 021057
 021057
 021057
 021057
 021057
 021057
 021057
 021057
 021057
 02105X          DATA CHECK
 021057
 021057
  XXXXX          EQUIPMENT CHECK
 021057
```

```
EOJ OPTICAL
```

Figure 46. SYSLST Output for Optical Reader Sample Problem -- Document Mode Processing (Part 5 of 5)

```
// JOB OPTICAL READER SAMPLE PROBLEM
// ASSGN SYS009,X'055'
// OPTION LINK,NOXREF
// EXEC ASSEMBLY
```

```
                                        EXTERNAL SYMBOL DICTIONARY
 SYMBOL   TYPE ID  ADDR   LENGTH LD ID


          PC  01 000000 0007BC
IJMDCBTZ  ER  02
IJMDCBTZ  SD  03 0007C0 0005F0
IJMDZBTZ  LD     0007C0          03
IJDFCZIW  ER  04
IJDFCZIW  SD  05 000DBC 000072
IJDFZZIW  LD     000DB0          05
```

```
    SAMPLE TEST PROGRAM FOR JOURNAL TAPE PROCESSING


  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                            DOS CL3-0

                                      2          PRINT NOGEN                                              $4781002
000000                                3          START                                                    $4781003
000000 0560                           4 BEGIN    BALR  6,0                                                $4781004
000002                                5          USING *,6                                                $4781005
                                      6          OPEN  INPUT,OUTPUT           OPEN OPTICAL READER AND PRINTER $4781006
                                     15 LOOP     GET   INPUT,ORWORKA          GET RECORD FROM OPTICAL READER  $4781007
000026 5B50 67AE            007B0    21          S     5,=F'1'                SUBTRACT ONE FROM RECORD COUNT  $4781008
00002A 4250 602D            0002F    22          STC   5,*+5                  PUT RECORD LENGTH IN MOVE       $4781009
00002E D20D 64D4 654D 004D6 0054F    23          MVC   PRWORKA(14),ORWORKA  MOVE RECORD TO OUTPUT AREA        $4781010
                                     24          PUT   OUTPUT,PRWORKA         PUT RECORD TO PRINTER           $4781011
000044 9240 64D4      004D6          30          MVI   PRWORKA,C' '           ZERO  PRINTER                   $4781012
000048 D277 64D5 64D4 004D7 004D6    31          MVC   PRWORKA+1(120),PRWORKA  IOAREA                         $4781013
00004E 47F0 6014            00016    32          B     LOOP                                                   $4781014
000052 90EF 60E2            000E4    33 CORRECT  STM   14,15,SAVE             SAVE REGISTERS                  $4781015
000056 9110 65EE      005F0          34          TM    INPUT+80,X'10'         TEST FOR NONRECOVERY            $4781016
00005A 4710 60CE            000D0    35          BO    ENDIPT                                                 $4781017
00005E 9108 65EE      005F0          36          TM    INPUT+80,X'08'         EQUIP CHECK TEST                $4781018
000062 4710 6074            00076    37          BO    CHECK1                 IF YES, BRANCH TO CHECK1        $4781019
000066 9101 65EE      005F0          38 REJECT   TM    INPUT+80,X'01'         DATA CHECK TEST                 $4781020
00006A 4710 60A2            000A4    39          BO    KEYBOARD               IF YES,BRANCH TO 'KEYBOARD'     $4781021
00006E 9102 65EE      005F0          40          TM    INPUT+80,X'02'         KEYBOARD CORRECTION TEST        $4781022
000072 4710 60C8            000CA    41          BO    RESTORE                IF YES,BRANCH AND RESTORE REGS  $4781023
000076 D21F 64FC 6573 004FE 00575    42 CHECK1   MVC   PRWORKA+40(32),MSG1    MOVE EQUIP MSG TO IO AREA       $4781024
                                     43          CNTRL INPUT,READKB           DSPLY INCOMPLETE READ           $4781025
00008A 91FF 65EE      005F0          49          TM    INPUT+80,X'FF'         TEST FOR ANY OTHER ERRORS       $4781026
00008F 4780 60C8            000CA    50          BZ    RESTORE                IF NO,BRANCH AND RESTORE REGS   $4781027
                                     51          CNTRL INPUT,MARK             MARK THE ERROR LINE             $4781028
0000A0 47F0 60C8            000CA    57          B     RESTORE                BRANCH AND RESTORE REGISTERS    $4781029
0000A4 9102 65EE      005F0          58 KEYBOARD TM    INPUT+80,X'02'         KEYBOARD CORRECTION TEST        $4781030
0000A8 4710 6074            00076    59          BO    CHECK1                 IF YES,BRANCH AND RE-READ LINE  $4781031
0000AC D209 64FC 6593 004FE 00595    60          MVC   PRWORKA+40(10),MSG2    MOVE DATA CK MSG TO IO AREA     $4781032
                                     61          RDLNE INPUT                  RE-READ CHARACTER               $4781033
0000BE 91FF 65EE      005F0          66          TM    INPUT+80,X'FF'         TEST FOR ANY OTHER ERRORS       $4781034
0000C2 4780 60C8            000CA    67          BZ    RESTORE                IF NO,BRANCH AND RESTORE REGS   $4781035
0000C6 47F0 6074            00076    68          B     CHECK1                 BRANCH AND RE-READ LINE         $4781036
0000CA 98EF 60E2            000E4    69 RESTORE  LM    14,15,SAVE             RESTORE REGISTERS               $4781037
0000CE 07FE                          70          BR    14                     RETURN TO MAIN LINE             $4781038
                                     71 ENDIPT   CLOSE INPUT,OUTPUT           CLOSE OPTICAL READER AND PRINTER $4781039
                                     80          EOJ                                                      $4781040
0000E4 00000000                      83 SAVE     DC    1F'0'                                              $4781041
0000E8 00000000                      84          DC    1F'0'                                              $4781042
0000EC 4040404040404040              85 FILINPT1 DC    CL256' '                                           $4781043
0001EC 4040404040404040              86          DC    CL124' '                                           $4781044
000268 4040404040404040              87 FILINPT2 DC    CL256' '                                           $4781045
000368 4040404040404040              88          DC    CL124' '                                           $4781046
0003E4 4040404040404040              89 FILOUPT1 DC    CL121' '                                           $4781047
00045D 4040404040404040              90 FILOUPT2 DC    CL121' '                                           $4781048
0004D6 4040404040404040              91 PRWORKA  DC    CL121' '                                           $4781049
00054F 4040404040404040              92 ORWORKA  DC    CL38' '                                            $4781050
000575 C5D8E4C9D7D4C5D5              93 MSG1     DC    C'EQUIPMENT CHECK(INCOMPLETE READ)'                $4781051
000595 C4C1E3C140C3C8C5              94 MSG2     DC    C'DATA CHECK'                                      $4781052
```

Figure 47. SYSLST Output for Optical Reader Sample Problem (Journal Tape Processing) (Part 1 of 4)

```
    SAMPLE TEST PROGRAM FOR JOURNAL TAPE PROCESSING

LOC   OBJECT CODE      ADDR1 ADDR2  STMT    SOURCE STATEMENT                                          DOS CL3-0

                                      95 ***DTFS AND MODULES                                              $4781053
                                      96 INPUT     DTFOR BLKSIZE=380,                                    X$4781054
                                                         BLKFAC=10,                                      X$4781055
                                                         COREXIT=CORRECT,                                X$4781056
                                                         DEVADDR=SYS009,                                 X$4781057
                                                         DEVICE=1287T,                                   X$4781058
                                                         CONTROL=YES,                                    X$4781059
                                                         EOFADDR=ENDIPT,                                 X$4781060
                                                         IOAREA1=FILINPT1,                               X$4781061
                                                         IOAREA2=FILINPT2,                               X$4781062
                                                         RECFORM=UNDEF,                                  X$4781063
                                                         RECSIZE=5,                                      X$4781064
                                                         WORKA=YES                                        $4781065
                                     188           ORMOD RECFORM=UNDEF,                                  X$4781066
                                                         BLKFAC=YES,                                     X$4781067
                                                         WORKA=YES,                                      X$4781068
                                                         CONTROL=YES,                                    X$4781069
                                                         IOAREA2=YES,                                    X$4781070
                                                         DEVICE=1287T                                     $4781071
                                     734 OUTPUT    DTFPR BLKSIZE=121,                                    X$4781072
                                                         CONTROL=YES,                                    X$4781073
                                                         DEVADDR=SYSLST,                                 X$4781074
                                                         IOAREA1=FILOUPT1,                               X$4781075
                                                         IOAREA2=FILOUPT2,                               X$4781076
                                                         WORKA=YES                                        $4781077
                                     755           PRMOD CONTROL=YES,                                    X$4781078
                                                         WORKA=YES,                                      X$4781079
                                                         IOAREA2=YES                                      $4781080
000000                               838           END   BEGIN                                            $4781081
000798 5B5BC2D6D7C5D540              839                 =C'$$BOPEN '
0007A0 5B5BC2C3D3D6E2C5              840                 =C'$$BCLOSE'
0007A8 000005A0                      841                 =A(INPUT)
0007AC 0000054F                      842                 =A(ORWORKA)
0007B0 00000001                      843                 =F'1'
0007B4 00000768                      844                 =A(OUTPUT)
0007B8 000004D6                      845                 =A(PRWORKA)
```

RELOCATION DICTIONARY

POS.ID	REL.ID	FLAGS	ADDRESS
01	01	0C	00000C
01	01	0C	000010
01	01	0C	0000D8
01	01	0C	0000DC
01	01	08	0005A9
01	01	08	0005AD
01	02	18	0005B1
01	01	08	0005C1
01	01	08	0005C5
01	01	08	0005C9
01	01	08	0005F5
01	01	0C	0005F8
01	01	0C	0005FC
01	01	08	000609
01	01	0C	000610
01	01	0C	000614
01	01	0C	000618
01	01	08	000629
01	01	08	000631
01	01	08	000639
01	01	08	000641
01	01	08	000649
01	01	08	000651
01	01	08	000659
01	01	08	000661
01	01	08	000669
01	01	08	000671
01	01	08	000679
01	01	08	000681
01	01	08	000689
01	01	08	000691
01	01	08	000699
01	01	08	0006A1
01	01	08	0006A9
01	01	08	0006B1
01	01	08	0006B9
01	01	08	0006C1
01	01	08	0006C9
01	01	08	0006D1

Figure 47. SYSLST Output for Optical Reader Sample Problem (Journal Tape Processing) (Part 2 of 4)

```
01        01        08        0006D9
01        01        08        0006E1
01        01        08        0006E9
01        01        08        0006F1
01        01        08        0006F9
01        01        08        000701
01        01        08        000709
01        01        08        000711
01        01        08        000719
01        01        08        000721
01        01        08        000729
01        01        08        000731
```

```
                                        RELOCATION DICTIONARY

POS.ID   REL.ID   FLAGS   ADDRESS

  01       01      08     000739
  01       01      08     000741
  01       01      08     000749
  01       01      08     000751
  01       01      08     000759
  01       01      08     000761
  01       01      0C     000770
  01       04      18     000779
  01       01      0C     000780
  01       01      08     000791
  01       01      0C     0007A8
  01       01      0C     0007AC
  01       01      0C     0007B4
  01       01      0C     0007B8

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
```

```
// EXEC LNKEDT
```

```
JOB  OPTICAL               DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN    MAP
LIST   ENTRY
```

```
10/20/67    PHASE   XFR-AD  LOCORE  HICORE  DSK-AD   ESD TYPE  LABEL     LOADED  REL-FR

         PHASE***   002800  002800  003621  13 5 1   CSECT               002800  002800

                                                     CSECT     IJMDCBTZ  002FC0  002800
                                                     * ENTRY   IJMDZBTZ  002FC0

                                                     CSECT     IJDFCZIW  0035B0  002800
                                                     * ENTRY   IJDFZZIW  0035B0
```

Figure 47. SYSLST Output for Optical Reader Sample Problem (Journal Tape Processing) (Part 3 of 4)

```
// EXEC
012 3456 789C
123 4567 890S
234 5678 901T
345 6789 012N
456 7890 123S
567 8901 234X
678 9012 345C
789 0123 456T
890 1234 567Z
901 2345 678-
012 3456 789C
123 4567 890S
234 5678 901T
345 6789 012N
456 7890 123S
567 8901 234X
678 9012 345C
789 0123 456T
890 1234 567Z
901 2345 678-
012 3456 789C
123 4567 890S
234 5678 901T
345 6789 012N
456 7890 123S
567 8901 234X
678 9012 345C
789 0123 456T
890 1234 567Z
901 2345 678-
012 3456 789C
123 4567 890S
234 5678 901T
345 6789 012N
456 7890 123S
567 8901 234X
678 9012 345C
789 0123 456T
890 1234 567Z
901 2345 678-
012 3456 789C
123 4567 890S
234 5678 901T
345 6789 012N
456 7890 123S
567 8901 234X
678 9012 345C
789 0123 456T
890 1234 567Z
901 2345 678-
012 3456 789C
123 4567 890S
234 5678 901T
345 6789 012N
456 7890 123S
567 8901 234X
678 9012 345C
789 0123 456T
890 1234 567Z
901 2345 678-
012 3456 789C
```

```
EOJ OPTICAL
```

Figure 47. SYSLST Output for Optical Reader Sample Problem (Journal Tape Processing) (Part 4 of 4)

Emulator (1401/1460 and 1440)

Figure 48 shows the SYSLST output of the IBM Model 2025, and IBM Model 2030 emulator sample problems for the IBM Model 1440. The SYSLST output for IBM Model 1401/1460 sample problem is identical except for the sequence numbers 0088-0107 (see Figure 48). Figure 49 shows the SYSLST output of the IBM Model 2040 emulator sample problem for the IBM Model 1440. The SYSLST output for the IBM Model 1401/1460 is identical except for the sequence numbers 0029-0048 (see Figure 49). The sample problems list 20 records on the printer, and if a tape is available and assigned it will write them out on tape. Card numbers (cols. 73-76) 004, 006, and 051 must be removed from the deck and replaced by cards as instructed. For a complete description of the Emulator sample program and its output, refer to the Emulator Program manual listed in the Preface.

```
001 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0088
002 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0089
003 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0090
004 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0091
005 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0092
006 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0093
007 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0094
008 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0095
009 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0096
010 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0097
011 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0098
012 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0099
013 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0100
014 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0101
015 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0102
016 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0103
017 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0104
018 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0105
019 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0106
020 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0107
```

Figure 48. SYSLST Output for Emulator Sample Problem (IBM Model 2030)

```
001 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0029
002 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0030
003 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0031
004 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0032
005 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0033
006 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0034
007 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0035
008 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0036
009 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0037
010 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0038
011 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0039
012 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0040
013 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0041
014 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0042
015 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0043
016 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0044
017 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0045
018 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0046
019 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0047
020 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 1401/1440/1460 SAMPLE PROGRAM     0048
```

• Figure 49. SYSLST Output for Emulator Sample Problem (IBM Model 2040)

Appendix A. IPL Control and ASSGN Statements for System Generation

The formats for the ADD and SET statements are given in this section. These statements are used at IPL time. If standard physical unit description and assignments are made when assembling the supervisor and these correspond to the configuration used when the system was generated, no ADD or ASSGN statements are required during subsequent IPL procedures. The SET statement is always required.

ADD (ADD A DEVICE)

Operation	Operand
ADD	X'cuu'[(k)],devicetype[,X'ss']

The entries in the operand field represent the following:

<u>X'cuu'</u>

Channel and unit numbers.

<u>(k)</u>

Specify S if the device is to be switchable (the device is physically attached to two adjacent channels). The designated channel is the lower of the two channels. If the device is not switchable, K=0-254. This indicates the priority on the channel of the device, with 0 the highest priority.

<u>devicetype</u>

Actual device (2311, 2400T9, 1443, etc). See Figure 5.

<u>X'ss'</u>

Device specifications: If absent for 7- or 9-track tapes, X'90' or X'C0' is assumed. (See <u>ASSGN</u> statement for the proper entries.)

The device specifications for an IBM 2702 Transmission Control Unit are:

X'00' for SAD0 X'02' for SAD2
X'01' for SAD1 X'03' for SAD3

The device specification for a 1053 attached to a 2848 is: X'01'.

If the device type is omitted for a 2702 Transmission Control Unit, X'00' is assumed.

SET (SET THE DATE)

Operation	Operand
SET	DATE= mm/dd/yy [,CLOCK=hh/mm/ss] dd/mm/yy

Select one of the formats for the date. Replace mm with the month, dd with the day, yy with the year. The format selected must concur with the option selected at system generation. Systems delivered by IBM use the mm/dd/yy format.

If the Timer Feature was specified during supervisor generation, include CLOCK=hh/mm/ss where hh is hours, mm is minutes, and ss is seconds.

ASSGN (ASSIGN LOGICAL NAME)

The ASSGN command assigns a logical input/output unit to a physical device.

Operation	Operand
ASSGN	SYSnnn,X'cuu' ,X'ss' ,ALT

The entries in the operand field represent the following:

SYSnnn

The symbolic unit name. It may be one of the following:

SYSRDR	SYSLST	SYSIN
SYSIPT	SYSLOG	SYSOUT
SYSPCH	SYSLNK	SYS000-SYS221

X'cuu'

Indicates the hexadecimal channel (c) and unit (uu) number. C=0 for the multiplexor channel. C=1-6 for selector channels 1-6.

X'ss'

Device specifications for 7- and 9-track tape are:

ss	Bytes Per Inch	Parity	Trans-late Feature	Convert Feature
10	200	odd	off	on
20	200	even	off	off
28	200	even	on	off
30	200	odd	off	off
38	200	odd	on	off
50	556	odd	off	on
60	556	even	off	off
68	556	even	on	off
70	556	odd	off	off
78	556	odd	on	off
90	800	odd	off	on
A0	800	even	off	off
A8	800	even	on	off
B0	800	odd	off	off
B8	800	odd	on	off
C0	800	9-track single density		
C0	1600	9-track dual density		
C8	800	9-track dual density		
C0	1600	9-track single density		

ALT

Alternate tape unit is to be used when the capacity of the original assignment is reached.

Appendix B. IOCS Modules for COBOL, RPG and PL/I

This appendix lists the preassembled IOCS modules used by COBOL, PL/I, and RPG object programs. No IOCS modules are required by FORTRAN. For assembler language programs, the user can preassemble IOCS modules as described in the Supervisor and Input/Output Macros publication referenced on the cover of this publication. The following preassembled modules for COBOL, PL/I, and RPG can be used by any other program whenever applicable. Each module name begins with a 3-character prefix and consists of a 5-character field corresponding to the option permitted in generation of the module.

CDMOD name = IJCabcde

```
a = F if RECFORM=FIXUNB (always for INPUT CMBND files)
  = V if RECFORM=VARUNB
  = U if RECFORM=UNDEF
```

```
b = A if CTLCHR=ASA is specified (not specified in CMBND)
  = Y if CTLCHR=YES is specified
  = C if CONTROL=YES is specified
  = Z if neither CTLCHR nor CONTROL is specified
```

```
c = I if TYPEFLE=INPUT
  = O if TYPEFLE=OUTPUT
  = C if TYPEFLE=CMBND
```

```
d = Z if neither WORKA nor IOAREA2 is specified
  = W if WORKA=YES is specified
  = I if IOAREA2=YES
  = B if both WORKA and IOAREA2 are specified
```

```
For CMBND files:
  If WORKA=YES is specified, d = W
  If WORKA=YES is not specified, d = Z
```

```
e = 0 if DEVICE=2540
  = 1 if DEVICE=1442
  = 2 if DEVICE=2520
  = 3 if DEVICE=2501
  = 4 if DEVICE=2540 and CRDERR is specified
  = 5 if DEVICE=2520 and CRDERR is specified
```

COBOL	RPG	PL/I
IJCFZIZ0*	IJCFCCZ0	IJCFAOI1*
IJCFZIZ1*	IJCFCCZ1	IJCFAOI2*
IJCFZIZ2*	IJCFCCZ2	IJCFAOI4*
IJCFZIZ3**	IJCFCIZ0	IJCFAOZ1*
IJCFZII0**	IJCFCIZ1	IJCFAOZ2*
IJCFZII1**	IJCFCIZ2	IJCFAOZ4*
IJCFZII2**	IJCFYOI0	IJCFYOI1***
IJCFZII3**	IJCFYOI1***	IJCFYOI2***
IJCFAOI0	IJCFYOI2***	IJCFYOI4
IJCFAOI1*	IJCFYOZ0	IJCFYOZ1***
IJCFAOI2*	IJCFYOZ1***	IJCFYOZ2***
IJCFAOI4*	IJCFYOZ2***	IJCFYOZ4
IJCFAOI5	IJCFZII0**	IJCFZII0**
IJCFAOZ0	IJCFZII1**	IJCFZII1**
IJCFAOZ1*	IJCFZII2**	IJCFZII2**
IJCFAOZ2*	IJCFZII3**	IJCFZII3**
IJCFAOZ4*	IJCFZIZ3**	IJCFZIZ0*
IJCFAOZ5		IJCFZIZ1*
IJCFZOI1*		IJCFZIZ2*
IJCFZOI2*		IJCFZIZ3**
IJCFZOI4*		IJCFZOI1*
IJCFZOI5		IJCFZOI2*
IJCFZOZ1*		IJCFZOI4*
IJCFZOZ2*		IJCFZOZ1*
IJCFZOZ4*		IJCFZOZ2*
IJCFZOZ5		IJCFZOZ4*

*common to COBOL and PL/I
**common to COBOL, RPG, and PL/I
***common to RPG and PL/I

PRMOD name = IJDabcde

a = F if RECFORM=FIXUNB
= V if RECFORM=VARUNB
= U if RECFORM=UNDEF

b = A if CTLCHR=ASA is specified
= Y if CTLCHR=YES is specified
= C if CONTROL=YES is specified
= Z if neither CTLCHR nor CONTROL is specified

c = P if PRINTOV=YES is specified
= Z if PRINTOV=YES is <u>not</u> specified

d = I if IOAREA2=YES is specified
= Z if IOAREA2=YES is <u>not</u> specified

e = W if WORKA=YES is specified
= Z if WORKA=YES is not specified

COBOL	RPG	PL/I
IJDFAPIZ	IJDFYPZW	IJDFAZIZ
IJDFAPZZ	IJDFYPZZ	IJDFAZZZ
IJDFZPIZ*		IJDFYZZZ
IJDFZPZZ*		IJDFYZIZ
		IJDFZPZZ*
		IJDFZPIZ*

*common to COBOL and PL/I

MTMOD name = IJFabcde

a = F if RECFORM=FIXUNB (or FIXBLK)
  = V if RECFORM=VARUNB (or VARBLK)
  = U if RECFORM=UNDEF

b = B if READ=BACK is specified
  = Z if READ=FORWARD or is not specified

c = C if CKPTREC=YES is specified
  = Z if CKPTREC=YES is not specified

d = W if WORKA=YES is specified
  = Z if WORKA=YES is not specified

e = Z always

COBOL	RPG	PL/I
IJFUBCZZ	IJFVZCWZ	IJFFBZZZ
IJFVBCZZ	IJFFZCZZ	IJFFZZZZ
IJFFBCZZ		IJFVZZZZ
IJFVBCWZ		IJFUBZZZ
		IJFUZZZZ

SDMODxx name = IJGabcde

a = F if records are fixed unblocked or fixed blocked
  = V if records are variable unblocked or variable blocked
  = U if records are undefined

b = U if file is an update file
  = I if file is an input file
  = O if file is an output file

c = E if ERROPT=YES is specified
  = Z if ERROPT=YES is not specified

d = T if TRUNCS=YES is specified
  = Z if TRUNCS=YES is not specified

e = C if CONTROL=YES is specified
  = Z if CONTROL=YES is not specified

COBOL	RPG	PL/I
IJGFIETZ*	IJGFIZZZ**	IJGFIETZ*
IJGFIZZZ**	IJGFOZZZ**	IJGFIEZZ
IJGFOZZZ**	IJGFUZZZ**	IJGZIEZZ
IJGFUZZZ**	IJGVIZZZ**	IJGFOEZZ
IJGUIEZZ*	IJGVOZZZ**	IJGFUEZZ
IJGUIZZZ	IJGVUZZZ**	IJGUIEZZ*
IJGUOZZZ		IJGUOEZZ
IJGUUZZZ		IJGUUEZZ
IJGVIEZZ*		IJGVIEZZ*
IJGVIZZZ**		IJGVOEZZ
IJGVOZZZ**		IJGVUEZZ
IJGVUZZZ**		

*common to COBOL and PL/I
**common to COBOL and RPG

ISMOD name = IJHabcde

a = A if RECFORM=BOTH and IOROUT specifies ADD or ADDRTR
  = B if RECFORM=FIXBLK and IOROUT specifies ADD or ADDRTR
  = U if RECFORM=FIXUNB and IOROUT specifies ADD or ADDRTR
  = Z if IOROUT specifies LOAD or RETRVE

b = L if IOROUT=LOAD
  = I if IOROUT=ADD
  = R if IOROUT=RETRVE
  = A if IOROUT=ADDRTR

c = R if TYPEFLE=RANDOM is specified
  = S if TYPEFLE=SEQNTL is specified
  = B if TYPEFLE=RANSEQ is specified
  = Z if TYPEFLE is not specified

d = C if CORINDX=YES is specified
  = Z if CORINDX is not specified

e = P if CORDATA=YES is specified
  = Z if CORDATA is not specified

COBOL	RPG	PL/I
IJHAABZZ	IJHZLZZZ**	IJHAARCP***
IJHAARCP***	IJHZRBZZ**	IJHAARCZ***
IJHAARCZ***	IJHZRRZZ**	IJHAARZP***
IJHAARZP***	IJHZRSZZ**	IJHAARZZ***
IJHAARZZ***		IJHBARCP***
IJHAASZZ		IJHBARCZ***
IJHAIZZZ		IJHBARZP***
IJHBABZZ		IJHBARZZ***
IJHBARCP***		IJHUARCP***
IJHBARCZ***		IJHUARCZ***
IJHBARZP***		IJHUARZP***
IJHBARZZ***		IJHUARZZ***
IJHBASZZ		IJHZLZZZ**
IJHBIZZZ		IJHZRRCZ***
IJHUABZZ		IJHZRRZZ**
IJHUARCP***		IJHZRSZZ**
IJHUARCZ***		
IJHUARZP***		
IJHUARZZ***		
IJHUASZZ		
IJHUIZZZ		
IJHZLZZZ**		
IJHZRBZZ*		
IJHZRRCZ***		
IJHZRRZZ**		
IJHZRSZZ**		

*common to COBOL and RPG
**common to COBOL, RPG, and PL/I
***common to COBOL and PL/I

DAMOD name = IJIabcde

a = F if RECFORM=FIXUNB
  = B if RECFORM=UNDEF (handles both UNDEF and FIXUNB)

b = A if AFTER=YES is specified
  = Z if AFTER is not specified

c = I if IDLOC is specified
  = Z if IDLOC is not specified

d = Z always

e = Z always

COBOL	RPG	PL/I
IJIBAIZZ	IJIBZZZZ**	IJIFAZZZ*
IJIBAZZZ		IJIFZZZZ*
IJIBZIZZ		
IJIBZZZZ**		
IJIFAIZZ		
IJIFAZZZ*		
IJIFZIZZ		
IJIFZZZZ*		

*common to COBOL and PL/I
**common to COBOL and RPG

Name list for MTMOD Work File Type Modules (TYPEFLE=WORK).

MTMOD name = IJFWabcd

a = E if ERROPT=YES is specified
  = Z if ERROPT is not specified

b = N if NOTEPNT=YES is specified
  = Z if NOTEPNT=YES is not specified
  = S if NOTEPNT=POINTS is specified

c = Z always

d = Z always

System I/O Modules

For MTMOD:

IJFWEZZZ
IJFWZNZZ
IJFWZZZZ

Name List for SDMOD Work File Type Modules

SDMODW name = IJGWabcd

a = E if ERROPT=YES
  = Z if ERROPT is not specified

b = R if NOTEPNT=POINTRW is specified
  = N if NOTEPNT=YES is specified
  = Z if NOTEPNT is not specified

c = Z always

d = Z always

System I/O Modules

IJGWEZZU
IJGWEZZZ
IJGWZNZZ
IJGWZRZZ

DIMOD name = IJJabcde

a = F for fixed unblocked record format

b = C for ASA and System/360 control character support for printers and punches

c = B if an output file is specified (processes both input and output)
= I if an input file is specified

d = I if IOAREA2=YES is specified
= Z if IOAREA2=YES is not specified

e = D for DOS DIMOD

COBOL	RPG	PL/I
		IJJFCBID
		IJJFCBZD
		IJJFCIID
		IJJFCIZD

Appendix C. IBM System Components Identification

Core Image Phases

All program phase names in the core image library are composed of two four-character parts. The first four characters uniquely identify the program. The next four characters identify the phase of the program. The first phase of a program to be executed from the core image library (cataloged on SYSRES) must be identical to the name specified in the // EXEC control statement. For RPG, the first 4 characters of the phase names are RPG1, although the processor is invoked by

// EXEC RPG.

All IBM-supplied phase names begin with an alphabetic character (A-Z). Three classes of programs are exceptions:

1. Transient routines. Type A transient routines (device error routines) have the prefix $$A. ($ is an alphabetic character in System/360.) Type B transient routines (OPEN, CLOSE, CHKPT, etc) have the prefix $$B.

Device Error Routines and OPEN/CLOSE Phases for Disk Operating System

$$ANERAE	Message Writer
$$ANERRA	Error recovery monitor
$$ANERRB	Error recovery monitor
$$ANERRC	Error recovery monitor
$$ANERRD	Tape-error recovery
$$ANERRE	Tape-error recovery
$$ANERRF	Tape-error recovery
$$ANERRG	Data cell (2321)-error recovery
$$ANERRH	Data cell error recovery
$$ANERRI	Data cell (2321)-error recovery
$$ANERRJ	Data cell (2321)-error recovery
$$ANERRK	Data cell (2321)-error recovery
$$ANERRL	Tape-error recovery
$$ANERRM	Message writer
$$ANERRN	Message writer
$$ANERRO	Attention routine
$$ANERRP	Message writer
$$ANERRQ	Message writer
$$ANERRR	Message writer
$$ANERRS	System control
$$ANERRT	1412/1419 error recovery
$$ANERRU	Unit record error recovery
$$ANERRV	Unit record error recovery
$$ANERRW	1419D error recovery
$$ANERRX	Paper tape error recovery
$$ANERRY	Physical attention routine
$$ANERRZ	Physical attention routine
$$ANERR0	Physical attention routine
$$ANERR1	Modify communication region
$$ANERR6	2495 Tape Cartridge Reader Error Recovery Phase
$$ANERR7	2495 Tape Cartridge Reader
$$ANERR8	2495 Tape Cartridge Reader
$$ANERR9	Optical character reader-error recovery procedure
$$BATST1	Autotest (2311 only)
$$BATST3	Autotest (2311 only)
$$BATTNA	Supervisor-program terminator

$$BATTNB	Supervisor-program terminator
$$BATTNC	Supervisor-initiator
$$BATTND	Supervisor-nonresident attention routine
$$BATTNE	Supervisor-nonresident attention routine
$$BATTNF	Supervisor-nonresident attention routine
$$BATTNG	Supervisor-nonresident attention routine
$$BATTNH	Supervisor-nonresident attention routine
$$BATTNI	Supervisor-foreground initiator
$$BATTNJ	Supervisor-foreground initiator
$$BATTNK	Supervisor-foreground initiator
$$BATTNL	Supervisor-foreground initiator
$$BATTNM	Supervisor-foreground initiator
$$BATTNN	Supervisor-nonresident attention
$$BATTNO	Supervisor-nonresident attention
$$BATTNP	Supervisor-nonresident attention
$$BCBLIS	COBOL
$$BCBLOP	COBOL
$$BCBODA	COBOL
$$BCBUSR	COBOL
$$BCBUSW	COBOL
$$BCCPT1	DTFCP tape open
$$BCEOV1	Monitor-EOV/EOF
$$BCHKPD	Disk-checkpoint
$$BCHKPE	Disk-checkpoint
$$BCHKPF	Disk-checkpoint
$$BCHKPT	Tape-checkpoint
$$BCHKP2	Tape-checkpoint
$$BCIS0A	ISFMS - CLOSE
$$BCLOSE	Close monitor
$$BCLOSP	Punch file close
$$BCLOS2	Close
$$BCMR01	Magnetic Character Reader-close
$$BCMT01	Tape EOF/EOV input-forward
$$BCMT02	Tape CLOSE-alternate switching
$$BCMT03	Tape CLOSE input-backward
$$BCMT04	Tape EOV output-forward
$$BCMT05	Tape CLOSE
$$BCMT06	Tape CLOSE-workfiles
$$BCMT07	Tape-alternate switching
$$BCTC01	CLOSE routine--BTAM
$$BDRSTR	Disk restart phase
$$BDUMP	Supervisor-program terminator
$$BDUMPB	Supervisor-program terminator
$$BDUMPD	Supervisor-program terminator
$$BENDFF	ISFMS-ENDFL (Load phase 2)
$$BENDFL	ISFMS-ENDFL (Load phase 1)
$$BEOJ	Supervisor-program terminator
$$BEOJ1	Supervisor-program terminator
$$BEOJ2	Supervisor-program terminator
$$BEOJ3	Supervisor-program terminator
$$BEOJ4	Supervisor-program terminator
$$BEOJ5	Supervisor-program terminator
$$BERRTN	CPMOD-error recovery
$$BETPRT	Error threshold message
$$BHDRCK	Terminal test request validation and comparison
$$BILSVC	Supervisor-program terminator
$$BINDEX	Cylinder index in core (ISFMS)
$$BJCOPT	Job Control-OPEN TAPE routine
$$BLEPRT	Line error print routine
$$BLISTV	List VTOC
$$BLOPEN	OPEN routine
$$BLSTIO	System control
$$BMMR20	Magnetic Character Reader Message
$$BMSGWR	Tape open/close
$$BMU100	MPS utility

$$BMU200	MPS utility
$$BMU300	MPS utility
$$BCOP12	DTFCP-OPEN
$$BOCPM1	Message writer
$$BOCPM2	Message writer
$$BOCPT1	Open DTFCP or DTFDI
$$BOCPT2	Open-Output tape
$$BOCPT3	Open-Output tape
$$BOCP01	Open-Disk labels
$$BOCP02	Open-Tape
$$BOCP11	OPEN - IBM Processor Programs
$$BOCP12	OPEN - IBM Processor Programs
$$BODACL	Close Routine-DA
$$BODAI1	Open input-DA
$$BODAIN	Direct access input-OPEN (Input)
$$BODAO1	Direct access-OPEN OUTPUT Phase 1
$$BODAO2	Direct access-OPEN OUTPUT Phase 2
$$BODAO3	Direct access OPEN-OUTPUT Phase 3
$$BODAO4	Direct access OPEN-OUTPUT Phase 4
$$BODAU1	Direct access-OPEN (user)
$$BODQUE	Dequeue JIB's
$$BODSPV	Disk VTOC display routine
$$BODSPW	Disk VTOC display routine
$$BOFLPT	DASD file protect
$$BOIS01	ISFMS - OPEN I/O Phase 1
$$BOIS02	ISFMS - OPEN I/O Phase 2
$$BOIS03	ISFMS - OPEN I/O Phase 3
$$BOIS04	ISFMS - OPEN I/O Phase 4
$$BOIS05	ISFMS - OPEN I/O Phase 5
$$BOIS06	ISFMS - OPEN I/O Phase 6
$$BOIS07	ISFMS - OPEN I/O Phase 7
$$BOIS08	ISFMS - OPEN I/O Phase 8
$$BOIS09	Index sequential independent overflow area integrity
$$BOIS10	ISFMS - OPEN
$$BOMR01	Magnetic Character Reader-open
$$BOMSG1	Message non-abort types
$$BOMSG2	Message writer - abort types
$$BOMSG3	Message writer
$$BOMSG4	Message writer
$$BOMSG5	Message writer
$$BOMSG6	Message writer
$$BOMTOM	Tape open message
$$BOMTOW	Tape open message
$$BOMT01	Tape OPEN input-forward-standard labels
$$BOMT02	Tape OPEN input-backward-standard labels
$$BOMT03	Tape OPEN output forward-standard label
$$BOMT04	Tape OPEN output-standard labels
$$BOMT05	Tape OPEN I/O-forward/backward Non-standard/unlabeled
$$BOMT06	Tape OPEN workfiles
$$BOOR01	Optical character reader-OPEN
$$BOPEN	Open monitor
$$BOPENC	OPENC
$$BOPENR	Relocation Phase 1
$$BOPEN2	Open monitor
$$BOPIGN	COBOL-open
$$BOPNLB	Locates source statement library. Directory also locates source statement library (for Assembler and COBOL)
$$BOPNR2	Relocation Phase 2
$$BORTV1	ISFMS Open
$$BORTV2	ISFMS Open
$$BOSDC1	Sequential disk I/O - CLOSE
$$BOSDC2	Sequential disk-close
$$BOSDI1	Sequential disk input - OPEN
$$BOSDI2	Sequential disk input - OPEN
$$BOSDI3	Sequential disk input - OPEN

$$BOSDO1	Sequential disk output Phase 1 - OPEN
$$BOSDO2	Sequential disk output Phase 2 - OPEN
$$BOSDO3	Sequential disk OPEN output Phase 3
$$BOSDO4	Sequential disk output Phase 4 - OPEN
$$BOSDO5	Sequential disk output Phase 5 - OPEN
$$BOSDO6	Sequential disk open - Phase 6
$$BOSDO7	Sequential disk output - OPEN
$$BOSDO8	Sequential disk output - OPEN
$$BOSDW1	Sequential disk workfiles Phase 1 - OPEN
$$BOSDW2	Sequential disk workfiles Phase 2 - OPEN
$$BOSDW3	Sequential workfile Phase 3 - OPEN
$$BOSD00	Sequential disk OPEN - output Phase 0
$$BOSD01	Sequential disk OPEN - output Phase 1
$$BOSIGN	COBOL-Open
$$BOTC01	OPEN routine
$$BOUR01	Unit record-OPEN routine
$$BOVDMP	Disk VTOC dump
$$BPCHK	Supervisor - program terminator
$$BPDUMP	Supervisor - program terminator
$$BPLOSE	PL/I OPEN/CLOSE
$$BPSW	Supervisor - program terminator
$$BRMSG1	Message writer CHKPT-RSTRT
$$BRMSG2	CHECKPOINT-RESTART message writer
$$BRSTRB	Tape Restart
$$BRSTRT	RESTART message writer - Phase 2
$$BRSTR2	Tape and DASD verify for restart
$$BRSTR3	DASD verify for restart
$$BRSTR4	Tape Reposition
$$BSETFF	ISFMS load Phase 2 of SETFL
$$BSETFG	ISFMS load Phase 3 of SETFL
$$BSETFH	ISFMS load Phase 4 of SETFL
$$BSETFL	ISFMS load Phase 1 of SETFL
$$BSETL	ISMOD - SETL
$$BSOPEN	OLTEP OPEN
$$BSYSWR	Supervisor-nonresident attention
$$BTCNCL	Cancel routine
$$BTERM	Supervisor - program terminator
$$BTMEBG	Terminal test request - IBM 1030 Manual Entry and Badge Reader
$$BT1030	Terminal test module - IBM 1030
$$BT1050	Terminal test module - IBM 1050
$$BT1060	Terminal test module - IBM 1060
$$BT2260	Terminal test module - IBM 2260
$$BT2740	Terminal test module - IBM 2740
$$BT2848	Terminal test module - IBM 2848
$$BZCE01	CE Serviceability Programs
$$BZTIME	OLTEP

2. Job control and linkage editor are preferred core image library routines and begin with a dollar sign ($) followed by an alphabetic character.

Job Control and Linkage Editor Routines for Disk Operating System

$JOBCTLA	Job control program
$JOBCTLD	Job control program
$JOBCTLF	Job control program
$JOBCTLG	Job control program
$JOBCTLJ	Job control program
$JOBCTLK	Job control program
$LNKEDT	Linkage editor - initialize/overhead - Phase 1
$LNKEDTA	Linkage editor - pass 2 processor Phase 7
$LNKEDTC	Linkage editor - catalog processor Phase 8
$LNKEDT0	Linkage editor 12-2-9 processor (ESD Only) Phase 2
$LNKEDT2	Linkage editor 12-2-9 processor (other than ESD) Phase 3
$LNKEDT4	Linkage editor - control card processor Phase 4
$LNKEDT6	Linkage editor - control card processor Phase 5
$LNKEDT8	Linkage editor - MAP processor Phase 6
$MAINEOJ	Library routine update program

3. The names of the supervisor and IPL programs are $$A$SUPx and $$A$IPLx. An additional character (x) is used to distinguish phases of these programs.

Supervisor and IPL Programs for Disk Operating System

$IPLRT2	IPL Program
$$A$SUP1	Supervisor control program

Relocatable Module Identification

Note: All components of the DOS Relocatable Library are listed here. For each component that has module and phase names listed alongside each other, a one-to-one relationship between the modules and phases is not necessarily intended or implied.

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
CONTROL PROGRAM				DELETR IJB.ALL
Job Control	IJBJC	$JOBCTLA	INCLUDE IJBJC	
	IJBJC1	$JOBCTLD	// EXEC LNKEDT	
	IJBJC2	$JOBCTLF		
	IJBJC3	$JOBCTLG		
	IJBJC4	$JOBCTLJ		
	IJBJC5	$JOBCTLK		
	IJBJC6			
IPL	IJBIPL	$IPLRT2	INCLUDE IJBIPL	
	IJBIPL3		// EXEC LNKEDT	
Linkage Editor	IJBLE		INCLUDE IJBLE	
	IJBLE1	$LNKEDT	// EXEC LNKEDT	
	IJBLBI	$LNKEDT0		
		$LNKEDT2		
		$LNKEDT4		
		$LNKEDT6		
		$LNKEDT8		
		$LNKEDTA		
		$LNKEDTC		
Standard System Dump	IJBDUMPS		INCLUDE IJBDUMPS	
	IJBDMPS	$$BDUMP	// EXEC LNKEDT	
	IJBDMPBS	$$BDUMPB		
	IJBDMPDS	$$BDUMPD		
	IJBDMPFS	$$BDUMPF		
	IJBPDMPS	$$BPDUMP		
	IJBPDUMS	$$BPDUM1		
Translating System Dump	IJBDUMPT		INCLUDE IJBDUMPT	
	IJBDMPT	$$BDUMP	// EXEC LNKEDT	
	IJBDMPBT	$$BDUMPB		
	IJBDMPDT	$$BDUMPD		
	IJBPDMPT	$$BPDUMP		

Librarian

DSERV	IJBSL1	DSERV	INCLUDE IJBSL1
			// EXEC LNKEDT
RESTART		IJBRSTRT	
MAINT	IJBSL2	MAINT	INCLUDE IJBSL2
	IJBLBA	MAINTA	// EXEC LNKEDT
	IJBLBC	MAINTCL	
	IJBLBD	MAINTCN	
	IJBLBE	MAINTDR	
	IJBLBF	MAINTEJP	
	IJBLBG	MAINTR2	
	IJBLBH	MAINTS2	
	IJBLBL	MAINTUP	
	IJBLBM	$MAINEOJ	
	IJBLBQ		
	IJBLBZ		
	IJBMCS		
	IJBMDS		
	IJBMDU		
	IJBMIN		
	IJBMIO		
	IJBMUP		
CSERV	IJBLBP	CSERV	INCLUDE IJBLBP
			// EXEC LNKEDT
RSERV	IJBSL3	RSERV	INCLUDE IJBSL3
			// EXEC LNKEDT

Component	Module Name	Phase Name	Linkage Edit Control Statments	Delete Statements
SSERV	IJBSL4	SSERV	INCLUDE IJBSL4 // EXEC LNKEDT	
CORGZ	IJBSI5 IJBLBJ IJBLBK IJBLBS IJBLBT IJBLBU IJBLBV	CORGZ CORGZ2 CORGZ1 CORGZ3 CORGZ4 CORGZ5	INCLUDE IJBSL5 // EXEC LNKEDT	
Common Librarian Modules	IJBLBA IJBLBC IJBLBD IJBLBE IJBLBF IJBLBG IJBLBH IJBLBI IJBLBJ IJBLBL			
CE Serviceability Programs	ILCEAID1 ILCEAID2	CEAIDLST CEAID CEAID001 CEAID002 CEAID003 CEAID004 CEAID005 CEAID006		
LANGUAGE TRANSLATORS				
Disk Work file assembler	IJQD16DW		INCLUDE IJQD16DW // EXEC LNKEDT	Disk Work Files DELETR IJQD16DW
Tape Work file assembler	IJQD16TW		INCLUDE IJQD16TW // EXEC LNKEDT	Tape Work Files DELETR IJQD16TW
Additional Core, 14K assembler	IJQD32		INCLUDE IJQD32 // EXEC LNKEDT	14K Tape and Disk Work Files DELETR IJQD32
Common Assembler Module	IJQDCOMN			All Assemblers DELETR IJQ.ALL
Additional Core, 44K Assembler	IJYASM		INCLUDE IJYASM // EXEC LNKEDT	44K Tape and Disk Work Files DELETR IJY.ALL

Note: The assembler relocatable modules are descriptive modules used by the linkage editor to incorporate appropriate assembler modules from the relocatable library into corresponding assembler phases for the core image library. The descriptive modules IJQD16DW, IJQD16TW, IJQD32 and IJYASM are used to determine the appropriate corresponding variant of the assembler to be linkage edited from the assembler modules in the relocatable library. IBM-supplied systems with 6K supervisors contain the IJQD16DW variant. IBM-supplied systems with 8K supervisors contain the IJQD32 variant.

The descriptive module IJQDCOMN is nested within the 10K and 14K descriptive modules. This module describes the appropriate relocatable library modules and the corresponding core image library phase structure for phases common to all three assembler variants.

The following is a complete list of relocatable library modules and core image library phases for the assembler.

Component	Module Name	Phase Name	Delete Statements
	IJQDCOMN	ASSEMBLY	(Sample problems)
	*IJQD16DW	ASSEM00A*+	DELETS Z.AS1
	+IJOD16TW	ASSEM00B*+	DELETS Z.AS2
	#IJQD32	ASSEM02	DELETS Z.AS3
	IJQABT	ASSEN11D	
	IJQDIA	ASSEN11C	
	IJQDPP	ASSEN11B	
	IJQD0$	ASSEM02A	
	IJAD0A	ASSEM03	
	IJQD2$	ASSEM03A	
	IJQD2A	ASSEM04	
	IJQD3$	ASSEM04A	
	IJQD3A	ASSEM04B	
	IJQD4P	ASSEM05	
	IJQD4M	ASSEM05A	
	IJQD4A	ASSEM05B	
	IJQD5A	ASSEM06	
	IJQD5P	ASSEN07	
	IJQD5M	ASSEN07A	
	IJQD7$	ASSEN07I	
	IJQD7I	ASSEN08	
	IJQD8$	ASSEN08A	
	IJQD9$	ASSEN088	
	IJQD9I	ASSEN09	
	IJQRTA	ASSEN07C	
	IJQRTB	ASSEN08C	
	IJQ10$	ASSEN09I	
	IJQ10B	ASSEN10	
	IJQ21A	ASSEN10B	
	IJQ21B	ASSEN11A	
		ASSEN11E	
		ASSEN12	
		ASSEN13	
		ASSEN14	

* required only for DWF variant

+ required only for TWF variant

required only for additional core storage variant S4

The following list is a complete list of the relocatable library modules and the core image library phases for the 44K assembler.

Module Name	Phase Name
IJYASM	ASSEMBLY
IJYF0	ASSEMABT
IJYABT	ASSEMFI
IJYCM	ASSEMFPP
IJYFD	ASSEMF8
IJYFI0	ASSEM3
IJYFPP	ASSEM3E
IJYF1	ASSEM7
IJYF2	
IJYF3	
IJYF3E	
IJYF7C	
IJYF7D	
IJYF7E	
IJYF7G	
IJYF7I	
IJYF7L	
IJYF7N	
IJYF7S	
IJYF7V	
IJYF7X	
IJYF8A	
IJYF8C	
IJYF8D	
IJYF8I	
IJYF8L	
IJYF8M	
IJYF8N	
IJYF8P	
IJYF8S	
IJYF8V	
IJYIN	
IJYRTA	
IJYRTB	

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
Autotest	IJVPT	ATLECONT	ACTION CLEAR	DELETR IJV.ALL
	IJVTAB	ATLEDT	INCLUDE IJVPT	(Sample
	IJVTC110	ATLEDT10	// EXEC LNKEDT	problem)
	IJVTC210	ATLEDT12		DELETS Z.AT1
	IJVTC310	ATLEDT14		
	IJVTC410	ATLEDT16		
	IJVTC510	ATLEDT18		
	IJVTC710	ATLEDT1A		
	IJVTE110	ATLEDT1B		
	IJVTE210	ATLEDT1C		
	IJVTF110	ATLEFC1		
	IJVTG110	ATLEFC2		
	IJVTH210	ATLEFC3		
	IJVTH310	ATLEFC4		
	IJVTD110	ATLEFC5		
	IJVTD210	ATLEFC7		
	IJVTI110	ATLEFD1		
	IJVTJ110	ATLEFD2		
	IJVTA010	ATLEFE1		
	IJVLE	ATLEFE2		
	IJVSS110	ATLEFF1		
	IJVSS310	ATLEFG1		
		ATLEFH2		
		ATLEFH3		
		ATLEGO1		
		ATLEJCTV		
		$$BATST1		
		$$BATST3		

Note: The autotest calling book (IJVPT) will include the Autotest transients in the core image library.

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
COBOL	IJSCBD	IJSCBD	INCLUDE IJSCBD	DELETR IJS.ALL
	IJSCBL01	COBOL	// EXEC LNKEDT	This deletes
	IJSCBL02	COBOL000		the debug
	IJSCBL03	COBOL001		program also.
	IJSCBL04	COBOL002		(Sample
	IJSCBL05	COBOL003		problem)
	IJSCBL06	COBOL004		DELETES Z.CB1
	IJSCBL07	COBOL005		
	IJSCBL08	COBOL006		
	IJSCBL09	COBOL007		
	IJSCBL10	COBOL008		
	IJSCBL11	COBOL009		
	IJSCBL12	COBOL010		
	IJSCBL13	COBOL011		
	IJSCBL14	COBOL012		
	IJSCBL15	COBOL013		
	IJSCBL16	COBOL014		
	IJSCBL17	COBOL015		
	IJSCBL18	COBOL016		
	IJSCBL19	COBOL017		
	IJSCBL20	COBOL018		
	IJSCBL21	COBOL019		

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
	IJSCBL22	COBOL020		
	IJSCBL23	COBOL021		
	IJSCBL24	COBOL022		
	IJSCBL25	COBOL023		
	IJSCBL26	COBOL024		
	IJSCBL27	COBOL025		
	IJSCBL28	COBOL027		
	IJSCBL29	COBOL028		
	IJSCBL31	COBOL029		
	IJSCBL32	COBOL030		
	IJSCBL33	COBOL031		
	IJSCBL34	COBOL032		
	IJSCBL35	COBOL033		
	IJSCBL36	COBOL034		
	IJSCBL37	COBOL035		
	IJSCBL38	COBOL036		
	IJSCBL39	COBOL037		
	IJSCBL40	COBOL038		
	IJSCBL41	COBOL039		
	IJSCBL42	COBOL040		
	IJSCBL43	COBOL041		
	IJSCBL44	COBOL042		
	IJSCBL45	COBOL043		
	IJSCBL46	COBOL044		
	IJSCBL47	COBOL050		
	IJSCBL48	COBOL055		
	IJSCBL49			
	IJSCBL50			
	IJSCBL55			
COBOL Debug	IJSDDB	DEBUG	INCLUDE IJSDDB	
	IJSCBL60		// EXEC LNKEDT	
FORTRAN	ILFFO		INCLUDE ILFFO	DELETR ILFFO
	ILFFORT	FFORTRAN	// EXEC LNKEDT	DELETR ILFFORT
	ILFPAR	FFORT1		DELETR ILFPAR
	ILFALL	FFORT2		DELETR ILFALL
	ILFUNF	FFORT3		DELETR ILFUNF
	ILFGEN	FFORT4		DELETR ILFGEN
	ILFEXT	FFORT5		DELETR ILFEXT
	ILFROL	FFORT9		DELETR ILFROL
	ILFTRBK	FFORTRBK		DELETR ILFTRBK
				DELETS Z.ILFSAMPL (Sample Problem)
				DELETS Z.ILFMERGE (Compatibility Merge Book)
Basic FORTRAN	IJTFO	FORTRAN	INCLUDE IJTFO	DELETR IJTFO
	IJTFO1	FORTREL	// EXEC LNKEDT	DELETR IJTFO1
	IJTFO2	FORTRGE		DELETR IJTFO2
	IJTFO3	FORTRPU		DELETR IJTFO3
	IJTFO4			DELETR IJTFO4
				(Sample problem) DELETS Z.FO1

PL/I (SYSIPT, SYSPCH, SYSLST never on 2311)	IJXA00	PL/I	INCLUDE IJXPLI // EXEC LNKEDT	DELFTR IJX.ALL (Sample problem) DELETS Z.PL1
PL/I (SYSIPT, SYSPCH, SYSLST may be on 2311)	IJXA00D	PL/I	INCLUDE IJXPLID // EXEC LNKEDT	

Note: The two compiler versions differ in the first phase only. The following list applies to both versions.

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
	IJXA10	PL/IA10		
	IJXA20	PL/IA20		
	IJXA25	PL/IA25		
	IJXA27	PL/IA27		
	IJXA30	PL/IA30		
	IJXA35	PL/IA35		
	IJXA45	PL/IA45		
	IJXA50	PL/IA50		
	IJXA60	PL/IA60		
	IJXA65	PL/IA65		
	IJXB10	PL/IB10		
	IJXB15	PL/IB15		
	IJXB20	PL/IB20		
	IJXB25	PL/IB25		
	IJXB30	PL/IB25A		
	IJXB40	PL/IB30		
	IJXB70	PL/IB40		
	IJXB75	PL/IB70		
	IJXB80	PL/IB75		
	IJXB85	PL/IB80		
	IJXB87	PL/IB85		
	IJXB90	PL/IB85A		
	IJXB92	PL/IB87		
	IJXB95	PL/IB90		
	IJXB97	PL/IB92		
	IJXC00	PL/IB95		
	IJXC03	PL/IB97		
	IJXC25	PL/IC00		
	IJXC30	PL/IC03		
	IJXC31	PL/IC25		
	IJXC32	PL/IC30		
	IJXC33	PL/IC31		
	IJXC34	PL/IC32		
	IJXC34A	PL/IC33		
	IJXC34B	PL/IC34		
	IJXC35	PL/IC34A		

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
	IJXC37	PL/IC34B		
	IJXC40	PL/IC35		
	IJXC50	PL/IC37		
	IJXC55	PL/IC40		
	IJXC60	PL/IC50		
	IJXC65	PL/IC55		
	IJXC85	PL/IC60		
	IJXC86	PL/IC65		
	IJXC95	PL/IC85		
	IJXD00	PL/IC86		
	IJXD03	PL/IC95		
	IJXD05	PL/ID00		
	IJXD10	PL/ID03		
	IJXD11	PL/ID05		
	IJXD15	PL/ID10		
	IJXD17	PL/ID11		
	IJXD20	PL/ID15		
	IJXD20A	PL/ID17		
	IJXD40	PL/ID20		
	IJXD70	PL/ID20A		
	IJXD75	PL/ID40		
	IJXD80	PL/ID70		
	IJXE25	PL/ID75		
	IJXE25A	PL/ID80		
	IJXE25B	PL/IE25		
	IJXE25C	PL/IE25A		
	IJXE25D	PL/IE25B		
	IJXE25E	PL/E25C		
	IJXE25F	PL/IE25D		
	IJXE25G	PL/IE25E		
	IJXE25H	PL/IE25F		
	IJXE25I	PL/IE25G		
	IJXE25J	PL/IE25H		
	IJXE25K	PL/IE25I		
	IJXE50	PL/IE25J		
	IJXE55	PL/IE25K		
	IJXE60	PL/IE50		
	IJXE61	PL/IE55		
	IJXF25	PL/IE60		
	IJXF35	PL/IE60A		
	IJXF50	PL/IF25		
	IJXF75	PL/IF35		
	IJXF90	PL/IF50		
	IJXF95	PL/IF75		
	IJXG00	PL/IF90		
	IJXG01	PL/IF95		
	IJXG15	PL/IG00		
	IJXG17	PL/IG01		
	IJXG17B	PL/IG15		
	IJXG17D	PL/IG17		
	IJXG17E	PL/IG17B		
	IJXG17R	PL/IG17D		
	IJXG17S	PL/IG17E		
	IJXG17X	PL/IG17R		
	IJXG17Y	PL/IG17S		
	IJXG20	PL/IG17X		
	IJXG25	PL/IG17Y		
	IJXG30	PL/IG20		
	IJXG31	PL/IG25		
	IJXG40	PL/IG30		
	IJXG55	PL/IG31		

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
	*IJXPLOSE	PL/IG40		
	*IJXS00	PL/IG55		
	*IJXS10	$$BPLOSE		
	*IJXS20	$IJKS00		
	*IJXS30	$IJKS10		
	*IJXS40	$IJKS20		
	*IJXS50	$IJKS30		
	*IJXS60	$IJKS40		
	*IJXS70	$IJKS50		
		$IJKS60		
		$IJKS70		
Object program I/O modules for 2311**	IJXSYSA IJXSYSI			

*At system generation time, these phases are cataloged into the core image library along with the PL/I compiler.

**These routines are either renamed to replace IJKSYSA or IJKSYSI respectively or deleted from the operational volume.

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
RPG	IJRRG	RPG1	INCLUDE IJRRG	DELETR IJR.ALL
	IJR000	RPG10010	// EXEC LNKEDT	(Sample
	IJR010	RPG10020		problem)
	IJR020	RPG10025		DELETS Z.RG1
	IJR025	RPG10030		
	IJR030	RPG10040		
	IJR039	RPG10050		
	IJR040	RPG10060		
	IJR049	RPG10070		
	IJR050	RPG10080		
	IJR059	RPG10090		
	IJR060	RPG10100		
	IJR069	RPG10110		
	IJR070	RPG10120		
	IJR079	RPG10130		
	IJR080	RPG10140		
	IJR089	RPG10150		
	IJR090	RPG10160		
	IJR099	RPG10170		
	IJR100	RPG1018A		
	IJR109	RPG1018G		
	IJR110	RPG10180		
	IJR119	RPG10190		
	IJR120	RPG10200		
	IJR129	RPG10210		
	IJR130	RPG10220		
	IJR139	RPG10230		
	IJR140	RPG10240		
	IJR149			
	IJR150			
	IJR159			
	IJR160			
	IJR169			
	IJR170			
	IJR179			
	IJR18A			
	IJR18F			
	IJR18G			
	IJR18H			
	IJR180			
	IJR189			
	IJR190			
	IJR199			
	IJR200			
	IJR209			
	IJR210			
	IJR219			
	IJR220			
	IJR229			
	IJR230			
	IJR239			
	IJR240			
	IJR241			
	IJR242			
	IJR243			
	IJR244			
	IJR245			
	IJR246			
	IJR247			
	IJR249			

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
SORT/MERGE		(see Appendix D for Disk Sort/Merge)		
Tape	IJPSM	TSRT	INCLUDE IJPSM	DELETR IJP.ALL
	IJPSM001	TSRTP002	// LBLTYP TAPE	(Sample
	IJPSM002	TSRTP003	// EXEC LNKEDT	problems)
	IJPSM003	TSRTP004		DELETS Z.SM1
	IJPSM004	TSRTP005		
		TSRTP006		
		TSRTP007		
		TSRTP008		
		TSRTP101		
		TSRTP102		
		TSRTP103		
		TSRTP104		
		TSRTP105		
		TSRTP201		
		TSRTP202		
		TSRTP203		
		TSRTP204		
		TSRTP301		
		TSRTP302		
		TSRTP303		
UTILITIES				DELETR IJW.ALL
				(Sample
				problems)
				DELETS Z.UTDKPR1
				DELETS Z.UTDCPR1
				DELETS Z.UTTPRR1
Alternate	IJWALTM	ATAM	INCLUDE IJWALTM	DELETR IJWALTM
Track	IJWAM1	ATAM2	// EXEC LNKEDT	DELETR IJWAM1
Assignment	IJWAM2	ATAM3		DELETR IJWAM2
Data Cell	IJWAM3	ATAM4		DELETR IJWAM3
	IJWAM4	ATAM5		DELETR IJWAM4
	IJWAM5			DELETR IJWAM5
Alternate	IJWAD	ATAD	INCLUDE IJWAD	DELETR IJWAD
Track	IJWAD1	ATAD2	// EXEC LNKEDT	DELETR IJWAD1
Assignment-	IJWAD2	ATAD3		DELETR IJWAD2
Disk	IJWAD3	ATAD4		DELETR IJWAD3
	IJWAD4	ATAD5		DELETR IJWAD4
	IJWAD5			DELETR IJWAD5
Card-to-Disk	IJWCD	CDDK	INCLUDE IJWCD	DELETR IJWCD
	IJWCD1	CDDK2	PHASE CDDK5, IJWCDCS2,NOAUTO	DELETR IJWCD1
	IJWCD3	CDDK3	INCLUDE IJWLAB	DELETR IJWCD3
	IJWCD4	CDDK4	// EXEC LNKEDT	DELETR IJWCD4
		CDDK5		
Card-to-	IJWCP	CDPP	INCLUDE IJWCP	DELETR IJWCP
Printer/Punch	IJWCP1	CDPP2	PHASE CDPP5, IJWCPCS2,NOAUTO	DELETR IJWCP1
	IJWCP3	CDPP3	INCLUDE IJWLAB	DELETR IJWCP2
	IJWCP4	CDPP4	// EXEC LNKEDT	DELETR IJWCP3
		CDPP5		DELETR IJWCP4

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
Card-to-Tape	IJWCT	CDTP	INCLUDE IJWCT	DELETR IJWCT
	IJWCT1	CDTP2	PHASE CDTP5, IJWCTCS2,NOAUTO	DELETR IJWCT1
	IJWCT3	CDTP3	INCLUDE IJWLAB	DELETR IJWCT3
	IJWCT4	CDTP4	// LBLTYP TAPE	DELETR IJWCT4
	IJWLAB	CDTP5	// EXEC LNKEDT	
Clear Data Cell	IJWCLM	CLDC	INCLUDE IJWCLM	DELETR IJWCLM
	IJWCLM1	CLDC2	// EXEC LNKEDT	DELETR IJWCLM1
	IJWCLD2	CLDC3		
	IJWCLD3			
Clear Disk	IJWCLD	CLRDSK	INCLUDE IJWCLD	DELETR IJWCLD
	IJWCLD1	CLRD2	// EXEC LNKEDT	DELETR IJWCLD1
	IJWCLD2	CLRD3		DELETR IJWCLD2
	IJWCLD3			DELETR IJWCLD3
Copy Disk to Card	IJWKC	CRDC	INCLUDE IJWKC	DELETR IJWKC
	IJWKC1	CRDC2	// LBLTYP NSD (nn)	DELETR IJWKC1
	IJWKC2		// EXEC LNKEDT	DELETR IJWKC2
Copy Disk to Disk	IJWRD	CRDD	INCLUDE IJWRD	DELETR IJWRD
	IJWRD1	CRDD2	// LBLTYP NSD (nn)	DELETR IJWRD1
	IJWRD2		// EXEC LNKEDT	DELETR IJWRD2
Copy Disk or Data Cell to Tape	IJWKT	CRDT	INCLUDE IJWKT	DELETR IJWKT
	IJWKT1	CRDT2	// LBLTYP NSD(nn)	DELETR IJWKT1
	IJWKT2		// EXEC LNKEDT	DELETR IJWKT2
Data Cell-to-Data Cell	IJWMM	DCDC	INCLUDE IJWMM	DELETR IJWMM
	IJWMM1	DCDC2	PHASE DCDC5,	DELETR IJWMM1
	IJWDD3		IJWMMCS2,NOAUTO	
	IJWDD4	DCDC3	INCLUDE IJWLAB	
		DCDC4	// EXEC LNKEDT	
		DCDC5		
Data Cell-to-Disk	IJWMD	DCDK	INCLUDE IJWMD	DELETR IJWMD
	IJWMD1	DCDK2	PHASE DCDK5,	DELETR IJWMD1
	IJWDD3		IJWMDCS2,NOAUTO	
	IJWDD4	DCDK3	INCLUDE IJWLAB	
		DCDK4	// EXEC LNKEDT	
		DCDK5		
Data Cell-to-Printer	IJWMP	DCPR	INCLUDE IJWMP	DELETR IJWMP
	IJWMP1	DCPR2	PHASE DCPR5,	DELETR IJWMP1
	IJWDP3		IJWMPCS2,NOAUTO	
	IJWDT4	DCPR3	INCLUDE IJWLAB	
		DCPR4	// EXEC LNKEDT	
		DCPR5		

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
Data Cell-to-	IJWMT	DCTP	INCLUDE IJWMT	DELETR IJWMT
Tape	IJWMT1	DCTP2	PHASE DCTP5,	DELETR IJWMT1
	IJWDT3		IJWMTCS2,NOAUTO	
	IJWDT4	DCTP3	INCLUDE IJWLAB	
		DCTP4	// LBLTYP TAPE	
		DCTP5	// EXEC LNKEDT	
Disk-to-Card	IJWDC	DKCD	INCLUDE IJWDC	DELETR IJWDC
	IJWDC1	DKCD2	PHASE DKCD5,	DELETR IJWDC1
			IJWDCCS2,NOAUTO	
	IJWDC3	DKCD3	INCLUDE IJWLAB	DELETR IJWDC3
	IJWDC4	DKCD4	// EXEC LNKEDT	DELETR IJWDC4
		DKCD5		
Disk-to-	IJWDM	DKDC	INCLUDE IJWDM	DELETR IJWDM
Data Cell	IJWDM1	DKDC2	PHASE DKDC5,	DELETR IJWDM1
	IJWDD3		IJWDMCS2,NOAUTO	
	IJWDD4	DKDC3	INCLUDE IJWLAB	
		DKDC4	// EXEC LNKEDT	
		DKDC5		
Disk-to-Disk	IJWDD	DKDK	INCLUDE IJWDD	DELETR IJWDD
	IJWDD1	DKDK2	PHASE DKDK5,	DELETR IJWDD1
			IJWDDCS2,NOAUTO	
	IJWDD3	DKDK3	INCLUDE IJWLAB	DELETR IJWDD3
	IJWDD4	DKDK4	// EXEC LNKEDT	DELETR IJWDD4
		DKDK5		
Disk-to-	IJWDP	DKPR	INCLUDE IJWDP	DELETR IJWDP
Printer	IJWDP1	DKPR2	PHASE DKPR5,	DELETR IJWDP1
			IJWDPCS2,NOAUTO	
	IJWDP3	DKPR3	INCLUDE IJWLAB	DELETR IJWDP3
	IJWDP4	DKPR4	// EXEC LNKEDT	DELETR IJWDP4
		DKPR5		
Disk-to-	IJWDT	DKTP	INCLUDE IJWDT	DELETR IJWDT
Tape	IJWDT1	DKTP2	PHASE DKTP5,	
			IJWDTCS2,NOAUTO	DELETR IJWDT1
	~~IJWDT3~~	~~DKTP3~~	~~INCLUDE IJWLAB~~	~~DELETR IJWDT3~~
	IJWDT4	DKTP4	// LBLTYP TAPE	DELETR IJWDT4
		DKTP5	// EXEC LNKEDT	
Initialize	IJWIM	INTM	INCLUDE IJWIM	DELETR IJWIM
Data Cell	IJWIM1	INTM2	// EXEC LNKEDT	DELETR IJWIM1
	IJWIM2	INTM3		DELETR IJWIM2
	IJWIM3	INTM4		DELETR IJWIM3
	IJWIM4			DELETR IJWIM4
Initialize	IJWID	INTD	INCLUDE IJWID	DELETR IJWID
Disk	IJWID1	INTD2	// EXEC LNKEDT	DELETR IJWID1
	IJWID2	INTD3		DELETR IJWID2
	IJWID3			DELETR IJWID3
	IJWID4			DELETR IJWID4
Restore Card	IJWRC	CRCD	INCLUDE IJWRC	DELETR IJWRC
to Disk			// LBLTYP NSD(nn)	
	IJWRC1		// EXEC LNKEDT	DELETR IJWRC1
Restore Tape	IJWRT	CRTD	INCLUDE IJWRT	DELETR IJWRT
to Disk or	IJWRT1		// LBLTYP NSD(nn)	DELETR IJWRT1
Data Cell			// EXEC LNKEDT	

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
Tape-to-Card	IJWTC	TPCD	INCLUDE IJWTC	DELETR IJWTC
	IJWTC1	TPCD2	PHASE TPCD5, IJWTCCS2,NOAUTO	DELETR IJWTC1
	IJWTC3	TPCD3	INCLUDE IJWLAB	DELETR IJWTC3
	IJWTC4	TPCD4	// LBLTYP TAPE	DELETR IJWTC4
		TPCD5	// EXEC LNKEDT	
Tape Compare	IJWTPCP			DELETR IJWTPCP
	IJWTCP	TPCP	PHASE TPCP,*, NOAUTO	DELETR IJWTCP
	IJWTCP2	TPCP2	INCLUDE IJWTCP	DELETR IJWTCP2
	IJWTCP3	TPCP3	INCLUDE IJJCPD1	DELETR IJWTCP3
	IJWXIT		INCLUDE IJWXIT	DELETR IJWXIT
			INCLUDE IJWTPCP	
			// LBLTYP TAPE	
			// EXEC LNKEDT	
Tape-to-Data Cell	IJWTM	TPDC	INCLUDE IJWTM	DELETR IJWTM
	IJWTM1	TPDC2	PHASE TPDC5,	DELETR IJWTM1
	IJWTD3		IJWTMCS2,NOAUTO	
	IJWTD4	TPDC3	INCLUDE IJWLAB	
		TPDC4	// LBLTYP TAPE	
		TPDC5	// EXEC LNKEDT	
Tape to Disk	IJWTD	TPDK	INCLUDE IJWTD	DELETR IJWTD
	IJWTD1	TPDK2	PHASE TPDK5, IJWTDCS2,NOAUTO	DELETR IJWTD1
	IJWTD3	TPDK3	INCLUDE IJWLAB	DELETR IJWTD3
	IJWTD4	TPDK4	// LBLTYP TAPE	DELETR IJWTD4
		TPDK5	// EXEC LNKEDT	
Tape to Printer	IJWTP	TPPR	INCLUDE IJWTP	DELETR IJWTP
	IJWTP1	TPPR2	PHASE TPPR5, IJWTPCS2,NOAUTO	DELETR IJWTP1
	IJWTP3	TPPR3	INCLUDE IJWLAB	DELETR IJWTP3
	IJWTP4	TPPR4	// LBLTYP TAPE	DELETR IJWTP4
		TPPR5	// EXEC LNKEDT	
Tape to Tape	IJWTT	TPTP	INCLUDE IJWTT	DELETR IJWTT
	IJWTT1	TPTP2	PHASE TPTP5, IJWTTCS2,NOAUTO	DELETR IJWTT1
	IJWTT3	TPTP3	INCLUDE IJWLAB	DELETR IJWTT3
	IJWTT4	TPTP4	// LBLTYP TAPE	DELETR IJWTT4
		TPTP5	// EXEC LNKEDT	
VTOC Display (BJS)	IJWLVB	LISTVTOC	INCLUDE IJWLVB	DELETR IJWLVB
	IJWLV1	$$BLISTV	// EXEC LNKEDT	
	IJWLVT			
VTOC Display (MPS)	IJWLVM	LISTVTOC	INCLUDE IJWLVM	DELETR IJWLVM
	IJWLV1	$$BLISTV	// EXEC LNKEDT	
	IJWLVT			

Note: Each of the above file to file utility programs share two common modules. These are:

	IJWGEN			DELETR IJWGEN
	IJWLAB			DELETR IJWLAB

The file-to-file data cell programs share the third and fourth modules from the counterpart disk programs.

The Clear Data Cell program shares the modules IJWCLD2 and IJWCLD3 with the clear disk program.

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements

Note: If a user label-checking routine is to be included instead of the IBM-supplied label routine, this module is included instead of the IBM-supplied module (i.e., IJWLAB). See the complete description of module names, phase names and the procedure for including user modules in IBM System/360 Disk and Tape Operating Systems, Utility Programs: Specifications, Form C24-3465.

Vocabulary File Utility Program	IJNVBL	VOC72BL	INCLUDE IJNVOC	DELETR IJN.ALL
	IJNVCT	VOC72BM	// LBLTYP NSD(01)	
	IJNVER	VOC72BN	// EXEC LNKEDT	
	IJNVIO	VOC72CR		
	IJNVLI	VOC72ER		
	IJNVLO	VOC72LI		
	IJNVOC	VOC72LO		
	IJNVUP	VOC72PR		
		VOC72UP		
		VOC72UQ		
		VOC72UR		
		VOC72UT		
COBCL SUBROUTINE				
	IHD00000			DELETR IHD.ALL
	IHD00100			
	IHD00200			
	IHD00300			
	IHD00400			
	IHD00500			
	IHD00600			
	IHD00700			
	IHD00800			
	IHD00900			
	IHD01000			
	IHD01100			
	IHD01200			
	IHD01300			
	IHD01400			
	IHD01500			
	IHD01600			
	IHD01700			
	IHD01800			
	IHD01900			
	IHD02000			
	IHD02100			
	IHD02200			
	IHD02300			
	IHD02400			
	IHD02500			
	IHD02600			
	IHD02700			

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
	IHD02800			
	IHD02900			
	IHD03000			
	IHD03100			
	IHD03200			
	IHD03300			
	IHD03400			
	IHD03500			
	IHD03600			
	IHD03700			
	IHD03800			
	IHD03900			
	IHD04000			
	IHD04100			
Basic FORTRAN SUBROUTINE				
	IJTAAFR			DELETR IJT.ALL
	IJTACOM			
	IJTACON			
	IJTADIR			
	IJTADXD			
	IJTADXI			
	IJTAIXI			
	IJTAPST			
	IJTARBE			
	IJTARXI			
	IJTARXR			
	IJTDVCK			
	IJTEXPN			
	IJTFDMP			
	IJTFIOS			
	IJTFXIT			
	IJTHEXC			
	IJTIFIX			
	IJTLEXP			
	IJTLLOG			
	IJTLSCN			
	IJTLSQT			
	IJTLTAN			
	IJTLTNH			
	IJTMAXD			
	IJTMODI			
	IJTMODR			
	IJTOVRF			
	IJTSINT			
	IJTSLIT			
	IJTSLOG			
	IJTSMX0			
	IJTSMX1			
	IJTSSCN			
	IJTSSQT			
	IJTSTAN			
	IJTSTNH			

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
FORTRAN SUBROUTINE				
	ILFACOM			DELETR ILF.ALL
	ILFADCON			
	ILFCLABS			
	ILFCLAS			
	ILFCLEXP			
	ILFCILOG			
	ILFCLSCN			
	ILFCLSQT			
	ILFCSABS			
	ILFCSAS			
	ILFCSEXP			
	ILFCSLOG			
	ILFCSSCN			
	ILFCSSQT			
	ILFDEBUG			
	ILFDIOCS			
	ILFFCDXI			
	ILFFCXPI			
	ILFFDUMP			
	ILFFDVCH			
	ILFFDXPD			
	ILFFDXPI			
	ILFFEXIT			
	ILFFINT			
	ILFFIOCS			
	ILFFIXPI			
	ILFFMAXD			
	ILFFMAXI			
	ILFFMAXR			
	ILFFOVER			
	ILFFRXPI			
	ILFFRXPR			
	ILFFSLIT			
	ILFGHTAB			
	ILFIBCOM			
	ILFIBERR			
	ILFLASCN			
	ILFLATN2			
	ILFLERF			
	ILFLEXP			
	ILFLGAMA			
	ILFLLOG			
	ILFLSCN			
	ILFLSCNH			
	ILFLSQRT			
	ILFLTANH			
	ILFLTNCT			
	ILFNAMEL			
	ILFSASCN			
	ILFSATN2			
	ILFSERF			
	ILFSEXP			
	ILFSGAMA			
	ILFSLOG			
	ILFSSCN			
	ILFSSCNH			
	ILFSSQRT			
	ILFSTANH			
	ILFSTNCT			
	ILFTRBK			
	ILFUNTAB			

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
PL/I SUBROUTINES				
	IJKEXHC			
	IJKQALM			DELETR IJK.ALL
	IJKQASM			
	IJKQBLA			
	IJKQBSA			
	IJKQCLA			
	IJKQCSA			
	IJKQDLA			
	IJKQDSA			
	IJKQLLA			
	IJKQLSA			
	IJKQNLD			
	IJKQNSD			
	IJKQQLM			
	IJKQQSM			
	IJKQRLB			
	IJKQRSB			
	IJKQSLD			
	IJKQSSD			
	IJKQTLB			
	IJKQTSB			
	IJKRBBM			
	IJKRBIM			
	IJKRBKA			
	IJKREBM			
	IJKRELM			
	IJKREPM			
	IJKRESM			
	IJKRGIM			
	IJKRGKM			
	IJKRMBX			
	IJKRMLX			
	IJKRMPX			
	IJKRMSX			
	IJKRSBM			
	IJKRSLM			
	IJKRSPM			
	IJKRSSM			
	IJKRUBM			
	IJKRWBM			
	IJKRWLM			
	IJKRWPM			
	IJKRWSM			
	IJKRXLM			
	IJKRXSA			
	IJKSDMP			
	IJKSDTM			
	IJKSTMM			
	IJKSYSA			
	IJKSYSI			
	IJKSZBA			
	IJKSZCA			

Component	Module Name	Phase Name	Linkage Edit Control Statements	Delete Statements
	IJKSZLM			
	IJKTCBM			
	IJKTCUM			
	IJKTDIM			
	IJKTDPD			
	IJKTFDM			
	IJKTFMM			
	IJKTGDI			
	IJKTICM			
	IJKTLIM			
	IJKTIOM			
	IJKTLTB			
	IJKTPSM			
	IJKTRGM			
	IJKTRON			
	IJKTSIM			
	IJKTSTM			
	IJKTXCF			
	IJKTXRM			
	IJKTXRN			
	IJKVBCM			
	IJKVBTM			
	IJKVCBM			
	IJKVCEM			
	IJKVCFM			
	IJKVCPM			
	IJKVCTM			
	IJKVECM			
	IJKVFCM			
	IJKVGIM			
	IJKVIGM			
	IJKVIIM			
	IJKVNPM			
	IJKVPCM			
	IJKVPNM			
	IJKVPRM			
	IJKVRPM			
	IJKVTBM			
	IJKVTCM			
	IJKXTBM			
	IJXSYSA			
	IJXSYSI			

Telecommunications

BTAM RELOCATABLE MODULE AND CORE IMAGE PHASE IDENTIFICATION

Relocatable Module Name

IJLASC	Table of special characters in ASCII for BSC
IJLEBD	Table of special characters in EBCDIC for Binary Synchronous Communication (BSC)
IJLTCD	Table of special characters in 6-bit transcode (for Binary Synchronous Communications)

IJL0BY	BSC Channel Program Models for Switched lines using No ID Verification (CPU-to-CPU)
IJL0BZ	BSC Channel Program Models for Nonswitched lines (CPU-to-CPU)
IJL0EZ	World Trade Telegraph Terminals (WTTA)
IJL00Y	IBM 7770
IJL10Y	IBM 7772
IJL01J	IBM 1030 (Auto Poll)
IJL01Z	IBM 1030
IJL02J	IBM 1060 (Auto Poll)
IJL02Z	IBM 1060
IJL03Z	IBM 2848 Remote
IJL04Z	AT&T 83B3
IJL05Z	Western Union 115A
IJL07J	IBM 1050 Nonswitched (Auto Poll)
IJL07Y	IBM 1050 Switched
IJL07Z	IBM 1050 Nonswitched
IJL08H	IBM 2740 with Station Control (Auto Poll)
IJL08M	IBM 2740 with Dial, Transmit Control, and Checking
IJL08P	IBM 2740 with Station Control and Checking
IJL08Q	IBM 2740 with Dial and Checking
IJL08R	IBM 2740 with Checking
IJL08U	IBM 2740 with Dial and Transmit Control
IJL08X	IBM 2740 with Station Control
IJL08Y	IBM 2740 with Dial
IJL08Z	IBM 2740 Basic
IJL089	IBM 2740 with Station Control and Checking (Auto Poll)
IJL09Y	TWX 33
IJLWTZ	WTTA Subroutines
IJL@6Z	Table Generation for 2260 Local Channel Program

The following relocatable modules are BSC channel program models for switched lines using ID verification (CPU-to-CPU). The correspondence between relocatable module name and the codes which appear in the FEATURE operand sublist in the DTFBT macro instruction is indicated.

IJL0BA	RIW
IJL0BC	SIW,RXW
IJL0BE	SXW,RXW
IJL0BG	SIX,RXW
IJL0BI	RXW
IJL0BK	SIW,RIX
IJL0BM	SXW,RIX
IJL0BO	SIX,RIX
IJL0BQ	RIX
IJL0BS	SIW
IJL0BU	SXW
IJL0BW	SIX
IJL0B4	SIW,RIW
IJL0B6	SXW,RIW
IJL0B8	SIX,RIW
IJL0CY	Channel Program Models for IBM 1130 on a point to point switched line
IJL0CZ	Channel Program Models for IBM 1130 on a point to point non-switched line
IJL0DY	IBM 2780 switched point to point
IJL0DZ	IBM 2780 non-switched point to point
IJL1BZ	BCS S/360 Model 20 Multipoint
IJL1CZ	Channel Program Models for a multidropped IBM 1130 on a multipoint line
IJL1DZ	IBM 2780 multipoint for EBCDIC
IJL2DZ	IBM 2780 Multipoint for USASCII or Transcode

The following relocatable modules are translate tables for the devices indicated.

IJLRASA	Translates from USASCII to EBCDIC for BSC
IJLRCTW	Translates from ITA2 to EBCDIC (World Trade Teletype)
IJLRCT1	Translates from BAUDOT to EBCDIC
IJLRCT2	Translates from TWX to EBCDIC
IJLRCT3	Translates from ZSC3 to EBCDIC (World Trade Teletype)
IJLRC30	Translates from 1080 to EBCDIC

IJLRC40	Translates from 2740 to EBCDIC (lowercase)
IJLRC50	Translates from 1050 to EBCDIC (lowercase)
IJLRC60	Translates from 1060 to EBCDIC
IJLRC80	Translates from 6-Bit TRANSCODE to EBCDIC
IJLRF40	Translates from 2740 to EBCDIC (uppercase)
IJLRF50	Translates from 1050 to EBCDIC (uppercase)
IJLRSCI	Translates from USASCII to EBCDIC with 2848 attached to 2701 via IBM Terminal Control Type III.
IJLSASA	Translates from EBCDIC to USASCII for BSC
IJLSCTW	Translates from EBCDIC to ITA2 (World Trade Teletype)
IJLSCT1	Translates from EBCDIC to BAUDOT
IJLSCT2	Translates from EBCDIC to TWX
IJLSCT3	Translates from EBCDIC to ZSC3 (World Trade Teletype)
IJLSD30	Translates from EBCDIC to 1030
IJLSD40	Translates from EBCDIC to 2740
IJLSD50	Translates from EBCDIC to 1050
IJLSD60	Translates from EBCDIC to 1060
IJLSD80	Translates from EBCDIC to 6-Bit TRANSCODE
IJLSSCI	Translates from EBCDIC to USASCII with 2848 attached to 2701 via IBM Terminal Control Type III

Transient Phase Names

$$ANERP2	Terminal Test Module to Write out Results of BSC On-Line
$$ANERR2	BTAM Error Recovery Message Writer, Phase 1
$$ANERR3	BTAM Error Recovery Message Writer, Phase 2
$$ANERR4	BTAM Error Recovery Message Writer, Phase 3
$$ANERR5	BTAM Error Recovery Message Writer, Phase 4
$$BCTC01	Close Routine
$$BETPRT	Error Threshold Message
$$BHDRCK	Terminal Test Request Validation and Comparison
$$BLEPRT	Line Error Print Routine
$$BLOPEN	Line OPEN Routine
$$BOTC01	OPEN Routine

$$BRESPL	Reset Polling Lines
$$BTCNCL	Cancel Routine
$$BTMEBG	Terminal Test Request for IBM 1030 Manual Entry and Badge Reader
$$BT1030	Terminal Test Module for IBM 1030
$$BT1050	Terminal Test Module for IBM 1050
$$BT1060	Terminal Test Module for IBM 1060
$$BT2260	Terminal Test Module for IBM 2260
$$BT2740	Terminal Test Module for IBM 2740
$$BT2848	Terminal Test Module for 2260, 1053 Remote

Core Image Phase Names	Test Patterns for On-Line Terminal Tests
IJLBOT02	Transparant EBCDIC Message
IJLBOT03	USASCII Transparency Message
IJLBOT04	Normal EBCDIC Message
IJLBOT05	Normal USASCII Message
IJLBOT06	Alphameric USASCII Message
IJLBOT07	USASCII Printer Message
IJLBOT08	USASCII Punch Message
IJLBOT09	TRANSCODE Printer Message
IJLBOT10	TRANSCODE Punch Message
IJLBOT11	TRANSCODE Multipoint Message
IJLBOT12	EBCDIC Printer Message
IJLBOT13	EBCDIC Punch Message
IJLBOT14	EBCDIC Alphameric Message
IJLBOT15	EBCDIC Weak Pattern Message for Switched Line
IJLBOT16	EBCDIC Weak Pattern Message for Leased Line
IJLBOT17	TRANSCODE Weak Pattern Message for Switched Line
IJLBOT18	TRANSCODE Weak Pattern Message for Leased Line
IJLBOT19	EBCDIC Weak Pattern for OLE SYN Insertion
IJLT2ALC	All Character Test for IBM 2848
IJLT2ROT	Rotate Test for IBM 2848
IJLT2TLT	Tilt Test for IBM 2848

IJLT2TWS	Twist Test for IBM 2848
IJLT3ALC	All Character Test for IBM 1030
IJLT3ROT	Rotate Test for IBM 1030
IJLT3SLA	Analyzer Test for IBM 1030 (Selectric®)
IJLT3TLT	Tilt Test for IBM 1030
IJLT3TWS	Twist Test for IBM 1030
IJLT5ALC	All Character Test for IBM 1050 or 2740
IJLT5ROT	Rotate Test for IBM 1050 or 2740
IJLT5SLA	SELCTRIC Analyzer Test for IBM 1050 or 2740
IJLT5TLT	Tilt Test for IBM 1050 or 2740
IJLT5TWS	Twist Test for IBM 1050 or 2740
IJLT6ALC	All Character Test for IBM 1060
IJLT6ROT	Rotate Test for IBM 1060
IJLT6SLA	SELECTRIC Analyzer Test for IBM 1060
IJLT6TLT	Tilt Test for IBM 1060
IJLT6TWS	Twist Test for IBM 1060

QTAM RELOCATABLE MODULE AND CORE IMAGE PHASE IDENTIFICATION

Relccatable Module Names

IJLQAA	Audio Line Appendage
IJLQAD	IBM 7772 Vocabulary Disk Appendage
IJLQBO	Breakoff (BREAKOFF)
IJLQCK	Checkpoint
IJLQCL	Change Line (STARTLN and STOPLN)
IJLQCM	Cancel message (CANCELM)
IJLQCP	Change Polling list entry (CHNGP)
IJLQCR	Checkpoint Request (CKREQ)
IJLQCT	Change terminal table entry (CHNGT)
IJLQDA	Disk I/O
IJLQDC	Copy Counters (COPYC)
IJLQDE	Copy terminal table entry (COPYT)

IJLQDL	Distribution list
IJLQDP	Copy polling list entry (COPYP)
IJLQDQ	Copy queue control block status (COPYQ)
IJLQDT	Insert date in message header (DATESTMP)
IJLQEA	End-of-address (EOA)
IJLQEB	End-of-block (EOB)
IJLQEC	End-of-block and line correction (EOBLC)
IJLQEP	Normal line appendage and ERP
IJLQER	Error message (ERRMSG)
IJLQEX	Expand message header
IJLQFL	DTF locator
IJLQGA	Get audio message
IJLQGB	Get audio or non-audio message (GET)
IJLQGC	Get audio message or non-audio message record (GET)
IJLQGD	Get audio message or non-audio message segment (GET)
IJLQGM	Get complete message (GET)
IJLQGR	Get message record (GET)
IJLQGS	Get message segment (GET)
IJLQIP	QTAM Implementation
IJLQIT	Intercept message (INTERCPT)
IJLQLA	Line appendage for PCI and program check
IJLQLG	Audio input message logging (LOGSEG)
IJLQLK	Look-up terminal table entry (DIRECT)
IJLQLO	IBM 2260 local appendage
IJLQLP	Time Procedure Specifications Control Routine
IJLQMC	Conversational mode (MODE)
IJLQMI	Initiate mode (MODE)
IJLQMM	Message-mode interface (MODE)
IJLQMP	Priority mode (MODE)
IJLQMT	Compare message type (MSGTYPE)
IJLQMW	Error Recovery Procedures Message Writer Subtask
IJLQM0	Model channel program for IBM 1030 terminals

IJLQM1	Model channel program for IBM 1060 terminals
IJLQM2	Model channel program for IBM 2260 terminals
IJLQM3	Model channel program for AT&T 83B3 terminals
IJLQM4	Model channel program for Western Union Plan 115A terminals
IJLQM5	Model channel program for IBM 1050 switched and nonswitched terminals
IJLQM6	Model channel program for IBM 1050 nonswitched terminals
IJLQM8	Model channel program for AT&T TWX terminals (Models 33 and 35)
IJLQM9	Model channel program for IBM 2260 local terminals
IJLQN0	Model channel program for IBM 2740 Basic terminals
IJLQN1	Model channel program for IBM 2740 Basic Dial terminals
IJLQN2	Model channel program for IBM 2740 terminals with Station Control
IJLQN3	Model channel program for IBM 2740 terminals with Station Control and Checking
IJLQN4	Model channel program for IBM 2740 Dial terminals with Transient Control and Checking
IJLQN5	Model channel program for IBM 2740 terminals with checking
IJLQN6	Model channel program for IBM 2740 Dial terminals with checking
IJLQN7	Model channel program for IBM 2740 Dial terminals with Transmit Control
IJLQN8	Model channel program for World Trade Telegraph Terminals (WTTA)
IJLQOA	Operator Awareness
IJLQOC	Operator Control (OPCTL)
IJLQPA	Put audio message (PUT)
IJLQPL	Polling limit control (POLLIMIT)
IJLQPM	Put complete message (PUT)
IJLQPR	Put message record (PUT)
IJLQPS	Put message segment (PUT)
IJLQPZ	Pause-transmit idle characters (PAUSE)
IJLQQT	Close message control (CLOSEMC)

IJLQRA	Translate table RCVARU: ARU code to EBCDIC
IJLQRB	Translation table RCVITA2 (EBCDIC to International Telegraph Alphabet No. 2)
IJLQRC	Translation table RCVZSC3 (EBCDIC to Figure Protected Code ZSC3)
IJLQRD	Retrieve message segment by DASD address (RETRIEVE)
IJLQRG	Route message (ROUTE)
IJLQRM	Release message (RELEASEM)
IJLQRR	Re-route message (REROUTE)
IJLQRS	Retrieve message header by sequence number (RETRIEVE)
IJLQRW	Physical input/output control
IJLQR1	Translate table RCV1030: 1030 to EBCDIC
IJLQR2	Translate table RCV1050: 1050 to EBCDIC
IJLQR3	Translate table RCV1050F: 1050 to monocase EBCDIC
IJLQR4	Translate table RCV1060: 1060 to EBCDIC
IJLQR5	Translate table RCV2260: 2260 to EBCDIC
IJLQR6	Translate tables RCV83B3 or RCV115A: AT&T 83B3 or WU Plan 115A to EBCDIC
IJLQR7	Translate table RCVTWX: AT&T Models 33/35 (TWX) to EBCDIC
IJLQR8	Translate table RCV2740: 2740 to EBCDIC
IJLQR9	Translate table RCV2740F: 2740 to monocase EBCDIC
IJLQSB	Translation table SNDITA2 (International Telegraph Alphabet No. 2 to EBCDIC)
IJLQSC	Translation table SNDZSC3 (Figure Protected Code ZSC3 to EBCDIC)
IJLQSH	Scan message header
IJLQSI	Sequence-in number verification (SEQIN)
IJLQSK	Skip-through-character (SKIP)
IJLQSO	Insert sequence-out number in message header (SEQOUT)
IJLQSR	Source terminal name verification (SOURCE)
IJLQSS	Change audio line (STARTARU and STOPARU)
IJLQST	Skip-on-count (SKIP)
IJLQS1	Translate table SND1030: EBCDIC to 1030
IJLQS2	Translate table SND1050: EBCDIC to 1050

IJLQS4	Translate table SND1060: EBCDIC to 1060
IJLQS5	Translate table SND2260: EBCDIC to 2260
IJLQS6	Translate tables SND83B3 or SND115A: EBCDIC to AT&T 83B3 or WU Plan 115A
IJLQS7	Translate table SNDTWXE: EBCDIC to AT&T Models 33/35 (TWX) (Even parity)
IJLQS8	Translate table SND2740: EBCDIC to 2740
IJLQS9	Translate table SNDTWXO: EBCDIC to AT&T Models 33/35 (TWX) (non parity)
IJLQTA	World Trade Telegraph Terminals (WTTA) Line Appendage
IJLQTR	Code translation; used in conjunction with QTAM or user-provided translate table (TRANS)
IJLQTS	Insert time-of-day in message header (TIMESTMP)
IJLQTT	Terminal test recognition (LPSTART)

Transient Phase Names

$$BQC01	QTAM	Close Phase 1
$$BQC02	QTAM	Close Phase 2
$$BQC03	QTAM	Close Phase 3
$$BOQ001	QTAM	QTAM Open Monitor/DASD Message Queues-Phase 1
$$BOQ002	QTAM	Open Line Group and Main-Storage Process/Destination Queues
$$BOQ003	QTAM	Open Checkpoint/Restart-Phase 1
$$BOQ004	QTAM	Open Checkpoint/Restart-Phase 2
$$BOQ006	QTAM	Open Main Storage Process/Destination Queues
$$BOQ007	QTAM	Open IBM 7772 Vocabulary DASD File
$$BOQ008	QTAM	Open Audio Line Group and Output Queue
$$BQCNCL	QTAM	QTAM Cancel
$$BQCNCM	QTAM	QTAM Cancel
$$BQHDCK	QTAM	Terminal Test Header Analysis
$$BQWTRA	QTAM	ARU Message Writer
$$BQWTR1	QTAM	Message Writer-Phase 1
$$BQWTR2	QTAM	Message Writer-Phase 2
$$BQWTR3	QTAM	Message Writer-Phase 3
$$BQ1030	QTAM	Terminal Test Module for IBM 1030

$$BQ1050	QTAM	Terminal Test Module for IBM 1050
$$BQ1060	QTAM	Terminal Test Module for IBM 1060
$$BQ2260	QTAM	Terminal Test Module for IBM 2260
$$BQ2740	QTAM	Terminal Test Module for IBM 2740

To delete the QTAM phases from the core image library, the QTAM modules from the relocatable library, and the QTAM books from the source statement library, the DELET card for the appropriate library and the name of the phase, module or book to be deleted must be supplied. For example:

```
// JOB DELETE
// EXEC MAINT
   DELETC $$BCQC01,$$BQWTR1,etc.
   DELETR IJLQBO,IJLQCL,IJLQCM,etc.
   DELETS A.BREAKOFF,A.BUFFER,A.CANCELM,etc.
/&
```

To delete the BTAM phases from the core image library, the BTAM modules from the relocatable library, and the BTAM books from the source statement library, the DELET card for the appropriate library and the name of the phase, module, or book to be deleted must be supplied. For example:

```
// JOB DELETC
// EXEC MAINT
   DELETC $$BCT01,$$BETPRT,etc.
   DELETC IJLT2ALC,IJLT2ROT,etc.
   DELETR IJL00Y,IJL01Z,IJL02Z,etc.
   DELETS A.CONTROL,A.LERB,etc.
/&
```

To delete both BTAM and QTAM from the core image, relocatable and source statement libraries, the DELET card for the appropriate library and the following entries must be made: For the core image transients, the DELETC card with a separate entry for each transient (QTAM and BTAM) to be deleted must be supplied. For the core image phase names, enter the delete card as follows:

```
   DELETC $$BCTC01,$$BOOO03,$$BETPRT,etc.
   DELETC IJLB.ALL
   DELETC IJLT.ALL
```

To delete from the relocatable library enter the delete card as follows:

```
   DELETR IJL.ALL
```

To delete the BTAM and QTAM books from the source statement library, enter the DELETS card with a separate entry for each book to be deleted. The following example shows the required delete cards and type of entries to be made in those cards.

```
// JOB DELETE
// EXEC MAINT
   DELETS A.CONTROL,A.CANCELM,etc.
/&
```

On Line Test Executive Program (OLTEP)
Relocatable Module Identification

IJZABOOK
IJZAAOLT
IJZACEOM
IJZACKTP
IJZACMNT
IJZACOMP
IJZACONV
IJZADATA
IJZADEVD
IJZADPRT
IJZAEXIO
IJZAHEAD
IJZAJOPT
IJZAOPEN
IJZAOPTN
IJZAOPUT
IJZARATA
IJZARSLT
IJZATEST
IJZATIME
IJZAWAIT

Core Image Phase Names

IJZAAOLT
IJZACEOM
IJZACKTP
IJZACMNT
IJZACOMP
IJZACONV
IJZADATA
IJZADEVD
IJZAHEAD
IJZAJOPT
IJZAOPTN
IJZARATA
IJZARSLT
IJZATEST

Transient Phase Name

$$BSOPEN
$$BZTIME

The linkage editor control statements for cataloging OLTEP into the core image library are:

```
   INCLUDE IJZABOOK
// EXEC LNKEDT
```

To delete OLTEP from the core image and relocatable libraries, the following control cards are required:

```
// JOB DELETE
// EXEC MAINT
   DELETC $$BSOPEN
   DELETC $$BZTIME
   DELETC IJZA.ALL
   DELETR IJZ.ALL
/&
```

PREASSEMBLED MODULES USED TO BUILD IBM COMPONENTS

Module Name
IJGWEZZZ
IJFWEZZZ
IJFWZNZZ
IJFWZZZZ
IJGWZNZZ
IJGWZRZZ
IJJCPDV
IJJCPDV1
IJJCPDV2
IJJCPD0
IJJCPD0N
IJJCPD1
IJJCPD1N
IJJCPD2
IJJCPD3
IJJCPV
IJJCPV1
IJJCPV2
IJJCP0
IJJCP0N
IJJCP1
IJJCP1N
IJJCP2
IJJCP3
IJJFCBID
IJJFCBZD
IJJFCIID
IJJFCIZD

SOURCE STATEMENT LIBRARY MACROS

The following are the names of the macro definitions in the source statement library of the IBM supplied systems residence volume.

	Supervisor Generation	File Definition	Supervisor Communications	IOCS Imperatives	Misc. Macros
A.ALLOC	x				
A.ARUMGTYP					Q
A.ASMTRTAB					B
A.ASSGN	x				
A.ATTACH			x		
A.BREAKOFF					Q
A.BTMOD					B
A.BTRWC					B
A.BTWAIT					B
A.BUFARU					Q
A.BUFFER					Q
A.CALL			x		
A.CANCEL			x		
A.CANCELM					Q
A.CCB				x	
A.CDMOD		x			
A.CHECK				MT,SD,MR	
A.CHECKARU					Q
A.CHGNTRY					B
A.CHKPT			x		

	Supervisor Generation	File Definition	Supervisor Communications	IOCS Imperatives	Misc. Macros
A.CHNG			8k		
A.CHNGP					Q
A.CHNGT					Q
A.CKREQ					Q
A.CLOSE				x	
A.CLOSEMC					Q
A.CLOSER				x	
A.CNTRL				x	
A.COMMN	x				
A.COMRG			x		
A.CONFG	x				
A.CONTROL				B	
A.COPYC					Q
A.COPYP					Q
A.COPYQ					Q
A.COPYT					Q
A.COUNTER					Q
A.CTLTBL					Q
A.DAMOD		x			
A.DATESTMP					Q
A.DEQ			x		
A.DETACH			x		
A.DFTRMLST					B
A.DIAG					E
A.DIMOD		x			
A.DIRECT					Q
A.DISEN					MR
A.DOSCHLV					M
A.DSPLY				x	
A.DTFBG		8k			
A.DTFBT					B
A.DTFBTND					B
A.DTFCD		x			
A.DTFCN		x			
A.DTFDA		x			
A.DTFDI		x			
A.DTFEN		8k			
A.DTFIS		x			
A.DTFIS1		x			
A.DTFIS2		x			
A.DTFIS3		x			
A.DTFMR		x			
A.DTFMT		x			
A.DTFOR		x			
A.DTFPH		x			
A.DTFPR		x			
A.DTFPT		x			
A.DTFQT					Q
A.DTFSD		x			
A.DTFSR		8k			
A.DUMP			x		
A.DVCGEN	x				
A.ENDFL				IS	
A.ENDRCV					Q
A.ENDREADY					Q
A.ENDSEND					Q
A.ENQ			x		
A.EOA					Q
A.EOB					Q
A.EOBLC					Q
A.EOJ			x		

	Supervisor Generation	File Definition	Supervisor Communications	IOCS Imperatives	Misc. Macros
A.ERET				x	
A.ERRMSG					Q
A.ESETL				IS	
A.EU3CG					E
A.EU3DK					E
A.EU3EJ					E
A.EU3ER					E
A.EU3FT					E
A.EU3MS					E
A.EU3OS					E
A.EU3PH					E
A.EU3PT					E
A.EU3RD					E
A.EU3TP					E
A.EU30					E
A.EU4CG					E
A.EU4DK					E
A.EU4EJ					E
A.EU4ER					E
A.EU4FT					E
A.EU4IN					E
A.EU4MS					E
A.EU4OS					E
A.EU4PH					E
A.EU4PT					E
A.EU4RD					E
A.EU4TP					E
A.EU40					E
A.EXCP				x	
A.EXIT			x		
A.FEOV				MT,SR	
A.FETCH			x		
A.FOPT	x				
A.FREE				x	
A.GET				x	
A.GETIME			x		
A.IJKZL					I
A.IJLBTMDS					B
A.IJLQBABD					Q
A.IJLQBFRD	Q				
A.IJLQBRBD					Q
A.IJLQCKPD					Q
A.IJLQCTLD					Q
A.IJLQDEQU					Q
A.IJLQDSCT	Q				
A.IJLQDTFD					Q
A.IJLQIP1D	Q				
A.IJLQLABD					Q
A.IJLQLCBD					Q
A.IJLQMCBD					Q
A.IJLQQCBD	Q				
A.IJLQSTBD	Q				
A.IJLQSVCD					Q
A.IJLQTBLD					Q
A.IJLQTSVC	Q				
A.IJLQVECD	Q				
A.INCARD					U
A.INDISK					U
A.INLOG					U
A.INTAPE					U
A.INTCR					U
A.INTERCPT					Q

	Supervisor Generation	File Definition	Supervisor Communications	IOCS Imperatives	Misc. Macros
A.IOTAB	x				
A.ISMOD		X			
A.ISMOD0		x			
A.ISMOD1		x			
A.ISMOD2		x			
A.ISMOD3		x			
A.ISMOD4		x			
A.ISMOD5		x			
A.ISMOD6		x			
A.ISMOD7		x			
A.ISMOD8		x			
A.ISMOD9		x			
A.LBRET			x		
A.LCBD					Q
A.LERB					B
A.LERPRT					B
A.LINE					Q
A.LINETBL					Q
A.LIST					Q
A.LITE					MR
A.LOAD			x		
A.LOADA					C
A.LODIS					C,P
A.LOGSEG					Q
A.LOPEN					B
A.LPSTART					Q
A.LUBGEN	x				
A.MODE					Q
A.MRMOD		x			
A.MSGTYPE					Q
A.MTMOD		x			
A.MVCOM			x		
A.NOTE				MT,SR	
A.ONLTST					B
A.OPCTL					Q
A.OPEN				x	
A.OPENR				x	
A.OPTION					Q
A.ORMOD		x			
A.OUTAPE					U
A.OUTCARD					U
A.OUTDISK					U
A.OUTLOG					U
A.OUTPRT					U
A.PAUSE					Q
A.PDUMP			x		
A.PIOCS	x				
A.POINTR				MT,SD	
A.POINTS				MT,SD	

	Supervisor Generation	File Definition	Supervisor Communications	IOCS Imperative	Misc. Macros
A.POINTW				MT, SD	
A.POLL					Q
A.POLLIMIT					Q
A.POST			x		
A.POSTARU					Q
A.POSTRCV					Q
A.POSTSEND					Q
A.PREFIXD					Q
A.PRMOD		x			
A.PROCESS					Q
A.PRTOV				PR	
A.PTMOD		x			
A.PUT				x	
A.QCBD					Q
A.RANDA					C
A.RCB			x		
A.RCVHDR					Q
A.RCVITA2					Q
A.RCVSEG					Q
A.RCVZSC3					Q
A.RDLNE				x	
A.READ				MT, SD, IS, DA, B, MR	
A.RELBUF					B
A.RELEASEM					Q
A.RELSE				MT, SD, SR	
A.REPEAT					Q
A.REQBUF					B
A.REROUTE					Q
A.RESCN				x	
A.RESETPL					B
A.RETRIEVE					Q
A.RETURN			x		
A.ROUTE					Q
A.RRUIS					C
A.RUADA					C, P
A.RUAIS					C, P
A.SAVE			x		
A.SDMOD		x			
A.SDMODFI		x			
A.SDMODFO		x			
A.SDMODFU		x			
A.SDMODUI		X			
A.SDMODUO		x			
A.SDMODUU		x			
A.SDMODVI		x			
A.SDMODVO		x			
A.SDMODVU		x			

	Supervisor Generation	File Definition	Supervisor Communications	IOCS Imperative	Misc. Macros
A.SDMODW		x			
A.SEND	x				
A.SENDHDR					Q
A.SENDSEG					Q
A.SEOV				x	
A.SEQDA					C
A.SEQIN					Q
A.SEQOUT					Q
A.SETFL				IS	
A.SETIME			x		
A.SETL				IS	
A.SGDFCH	x				
A.SGDSK	x				
A.SGSVC	x				
A.SGTCHS	x				
A.SGTCON	x				
A.SGTHAP	x				
A.SGUNCK	x				
A.SKIP					Q
A.SMICR	MR				
A.SNDITA2					Q
A.SNDZSC3					Q
A.SOURCE					Q
A.SRUIS					C
A.STARTARU					Q
A.STARTLN					Q
A.STCBD					Q
A.STDJC	x				
A.STOPARU					Q
A.STOPLN					Q
A.STXIT			x		
A.SUPVR	x				
A.TECB			x		
A.TERM					Q
A.TERMTBL					Q
A.TERMTBLD					Q
A.TIMESTMP					Q
A.TRANS					Q
A.TRNSLATE					B
A.TRSRCTW					B
A.TRSRCT3					B
A.TRSSCTW					B
A.TRSSCT3					B
A.TRUNC				MT,SD,SR	
A.TWAIT					B
A.WAIT			x		
A.WAITF				IS,DA,MR	
A.WAITM			x		

	Supervisor Generation	File Definition	Supervisor Communications	IOCS Imperative	Misc. Macros
A.WORD					Q
A.WORDTBL					Q
A.WRITE				MT,SD,IS,DA,B	
A.WRU					Q
Z.AS1					S
Z.AS2					S
Z.AS3					S
Z.AS4					S
Z.AS5					S
Z.AS6					S
Z.AT1					S
Z.CB1					S
Z.DELETECL					D
Z.DELETERL					D
Z.DELETESL					D
Z.EU3SPRGM					S
Z.EU4SPRGM					S
Z.FO1					S
Z.ILFMERGE					F
Z.ILFSAMPL					S
Z.LINKEDIT					L
Z.MCR1					S
Z.MCR2					S
Z.ORDC					S
Z.ORJT					S
Z.PL1					S
Z.RG1					S
Z.SM1					S
Z.SM2					S
Z.SM4					S
Z.SM5					S
Z.SM6					S
Z.UTDCPR1					S
Z.UTDKPR1					S
Z.UTDKPR2					S
Z.UTDKPR3					S
Z.UTTPPR1					S
Z.UTTPPR2					S
Z.VFU1					S

Key:	x	generally applicable
	B	BTAM
	C	COBOL DASD macros (these macros are not necessary for COBOL compilation or execution)
	D	Delete book
	E	emulator macro
	F	Basic FORTRAN Compatability with FORTRAN
	I	PL/I macro
	L	Linkage edit book
	M	display change level book
	P	PL/I DASD macros
	Q	QTAM
	S*	sample problem
	U	utility macros
	MR	magnetic character readers
	MT	magnetic tape
	SD	sequential DASD
	IS	index sequential
	DA	direct access
	SR	serial
	PR	printer
	8k	for BPS and BOS compatibility
	*	for 1285 sample problem change DEVICE on the cards with the sequence numbers 1054 and 1070 from DEVICE=1287 to DEVICE=1285

Appendix D. Disk Sort/Merge Modules

The Disk Sort/Merge program residing in the relocatable library consists of three entries:

1. primary processor generation modules
2. intermediate processor generation modules
3. relocatable object modules.

These modules enable a user to linkage edit into the core image library only those modules required to tailor a generalized sort/merge program to specific job applications. At system generation time, the user can linkage edit any one of seven distinct sort/merge object programs into the core image library.

The following programs can be generated at linkage edit time:

1. The entire sort/merge program.
2. A program that
   a. sorts fixed-length records;
   b. performs the ADDROUT option for fixed or variable-length record; and
   c. merges fixed or variable-length records.
3. A program that
   a. sorts variable-length records;
   b. performs the ADDROUT option for fixed or variable-length records; and
   c. merges fixed or variable-length records.
4. A program that
   a. sorts fixed or variable-length records; and
   b. performs the ADDROUT option for fixed or variable-length records.
5. A program that
   a. sorts fixed-length records; and
   b. performs the ADDROUT option for fixed or variable-length records.

6. A program that

   a. sorts variable-length records; and

   b. performs the ADDROUT option for fixed or variable-length records.

7. A program that merges fixed or variable-length records.

The user should linkage edit only the Sort/Merge program that satisfies his particular requirements. To generate:

```
Program 1.  specify:  INCLUDE IJOSM
                      // EXEC LNKEDT

Program 2.  specify:  INCLUDE IJOSMFOM
                      // EXEC LNKEDT

Program 3.  specify:  INCLUDE IJOSMVOM
                      // EXEC LNKEDT

Program 4.  specify:  INCLUDE IJOSMFVS
                      // EXEC LNKEDT

Program 5.  specify:  INCLUDE IJOSMFOS
                      // EXEC LNKEDT

Program 6.  specify:  INCLUDE IJOSMVOS
                      // EXEC LNKEDT

Program 7.  specify:  INCLUDE IJOSMERG
                      // EXEC LNKEDT
```

Primary Processor Generation Books

```
CATALR    IJOSM

INCLUDE   IJOSMPH0
INCLUDE   IJOSMPH1
INCLUDE   IJOSMF23   Entire Sort/Merge program.
INCLUDE   IJOSMV23
INCLUDE   IJOSMPH4
END

CATALR    IJOSMFOM

INCLUDE   IJOSMPH0   1.  Sort program for fixed-length records.
INCLUDE   IJOSMPH1   2.  ADDROUT option (fixed or variable).
INCLUDE   IJOSMF23   3.  Merge-only for fixed or variable-length
                         records.
INCLUDE   IJOSMPH4
END
```

```
CATALR    IJOSMVOM

INCLUDE   IJOSMPH0   1.  Sort program for variable-length records.
INCLUDE   IJOSMPH1   2.  ADDROUT option (fixed or variable).
INCLUDE   IJOSMV23   3.  Merge-only for fixed or variable-length
                         records.
INCLUDE   IJOSMPH4
END

CATALR    IJOSMFVS

INCLUDE   IJOSMPH0   1.  Sort program for fixed or variable-length
                         records.
INCLUDE   IJOSMPH1   2.  ADDROUT option (fixed or variable).
INCLUDE   IJOSMF23
INCLUDE   IJOSMV23
END

CATALR    IJOSMFOS

INCLUDE   IJOSMPH0   1.  Sort program for fixed-length records.
INCLUDE   IJOSMPH1   2.  ADDROUT option (fixed or variable).
INCLUDE   IJOSMF23
END

CATALR    IJOSMVOS

INCLUDE   IJOSMPH0   1.  Sort program for variable-length records.
INCLUDE   IJOSMPH1   2.  ADDROUT option (fixed or variable).
INCLUDE   IJOSMV23
END

CATALR    IJOSMERG

INCLUDE   IJOSMPH0   1.  Merge-only program for fixed or variable-length
                         records.
INCLUDE   IJOSMPH4
END
```

Intermediate Processor Generation Books

```
CATALR    IJOSMPH0

PHASE     DSORT
INCLUDE   IJOSM001
PHASE     DSORT002
INCLUDE   IJOSM002
PHASE     DSORT003
INCLUDE   IJOSM003
PHASE     DSORT004
INCLUDE   IJOSM004
PHASE     DSORT005
INCLUDE   IJOSM005
PHASE     DSORT006
INCLUDE   IJOSM006
PHASE     DSORT007
INCLUDE   IJOSM007
PHASE     DSORT008
INCLUDE   IJOSM008
PHASE     DSORT009
```

```
INCLUDE   IJOSM009
PHASE     DSORT010
INCLUDE   IJOSM010
END

CATALR    IJOSMV23

PHASE     DSORT203
INCLUDE   IJOSM203
PHASE     DSORT204
INCLUDE   IJOSM204
PHASE     DSORT303
INCLUDE   IJOSM303
PHASE     DSORT304
INCLUDE   IJOSM304
END

CATALR    IJOSMERG

PHASE     DSORT401
INCLUDE   IJOSM401
PHASE     DSORT402
INCLUDE   IJOSM402
END

CATALR    IJOSMPH1

PHASE     DSORT101
INCLUDE   IJOSM101
PHASE     DSORT102
INCLUDE   IJOSM102
PHASE     DSORT103
INLCUDE   IJOSM103
PHASE     DSORT104
INCLUDE   IJOSM104
PHASE     DSORT105
INCLUDE   IJOSM105
END

CATALR    IJOSMF23

PHASE     DSORT201
INCLUDE   IJOSM201
PHASE     DSORT202
INCLUDE   IJOSM202
PHASE     DSORT301
INCLUDE   IJOSM301
PHASE     DSORT302
INCLUDE   IJOSM302
PHASE     DSORT501
INCLUDE   IJOSM501
END
```

1. IJOSMPH0 = Assignment Phase.
2. IJOSMPH1 = Phase 1.
3. IJOSMF23 = Phases 2-3 for fixed-length records.
4. IJOSMV23 = Phase 2-3 for variable-length records.
5. IJOSMPH4 = Merge-only or Phase 4.

Phase Names and Relocatable Module Names

Phase Name	Csect Name	Module Name	Required	Program Phase
DSORT	IJO00110	IJOSM001		
DSORT002	IJO00210	IJOSM002		
DSORT003	IJO00310	IJOSM003		
DSORT004	IJO00410	IJOSM004		Assignment
DSORT005	IJO00510	IJOSM005	Always	Phase
DSORT006	IJO00610	IJOSM006	required	
DSORT007	IJO00710	IJOSM007		(Phase 0)
DSORT008	IJO00810	IJOSM008		
DSORT009	IJO00910	IJOSM009		
DSORT010	IJO01010	IJOSM010		
DSORT101	IJO10110	IJOSM101		
DSORT102	IJO10210	IJOSM102	Required	
DSORT103	IJO10310	IJOSM103	for	Phase 1
DSORT104	IJO10410	IJOSM104	Sort	
DSORT105	IJO10510	IJOSM105		
DSORT201	IJO20110	IJOSM201	Fixed-length	
DSORT202	IJO20210	IJOSM202	records: SORT	Phase 2
DSORT203	IJO20310	IJOSM203	Variable-length	
DSORT204	IJO20410	IJOSM204	records: Sort	
DSORT301	IJO30110	IJOSM301	Fixed-length	
DSORT302	IJO30210	IJOSM302	records: Sort	Phase 3
DSORT303	IJO30310	IJOSM303	Variable-length	
DSORT304	IJO30410	IJOSM304	records: Sort	
DSORT401	IJO40110	IJOSM401	Merge-only	Phase 4
DSORT402	IJO40210	IJOSM402	operations	(Merge)

Appendix E. Tape and Disk Sort/Merge Modules

The Tape and Disk Sort/Merge program residing in the relocatable library consists of three entries:

- primary processor generation modules
- intermediate processor generation modules
- relocatable object modules.

These modules enable a user to linkage edit into the core image library only those modules required to tailor a generalized sort/merge program to specific job applications. At system generation time, the user can linkage edit any one of six distinct sort/merge object programs into the core image library:

1. The entire sort/merge program.
2. The entire sort program.
3. The merge program.
4. A sort program that uses 2400 tape units as intermediate storage.
5. A sort program that uses 2311 direct access devices as intermediate storage.
6. A sort program that uses 2314 direct access devices as intermediate storage.

To conserve library space, the user should linkage edit only the sort/merge program that satisfies his particular requirements. Note that an attempt to execute an option not selected will result in abnormal termination of the sort/merge program. To generate:

```
Program 1.  specify:  INCLUDE ILHSALL
                      // EXEC LNKEDT

Program 2.  specify:  INCLUDE ILHSORT
                      // EXEC LNKEDT

Program 3.  specify:  INCLUDE ILHSMRG
                      // EXEC LNKEDT

Program 4.  specify:  INCLUDE ILHST
                      // EXEC LNKEDT

Program 5.  specify:  INCLUDE ILHSD1
                      // EXEC LNKEDT

Program 6.  specify:  INCLUDE ILHSD4
                      // EXEC LNKEDT
```

If the sort/merge program is generated in a system without multiprogramming support, the following primary processor generation module must be linkage edited immediately after the selected program: .

```
INCLUDE ILHSNMPS
// EXEC LNKEDT
```

Primary Processor Generation Books

```
CATALR    ILHSALL

INCLUDE   ILHSPH0
INCLUDE   ILHSPH1
INCLUDE   ILHSPH2    Entire Sort/Merge Program
INCLUDE   ILHSPH3
END

CATALR    ILHSORT

INCLUDE   ILHSPH0S
INCLUDE   ILHSPH1
INCLUDE   ILHSPH2    Entire Sort Program
INCLUDE   ILHSPH3S
END

CATALR    ILHSMRG

INCLUDE   ILHSPH0M
INCLUDE   ILHSPH3M   Merge Program
END

CATALR    ILHST

INCLUDE   ILHSPH0T
INCLUDE   ILHSPH1T
INCLUDE   ILHSPH2T   2400 Sort Program
INCLUDE   ILHSPH3T
END

CATALR    ILHSD1

INCLUDE   ILHSPH01
INCLUDE   ILHSPH11
INCLUDE   ILHSPH21   2311 Sort Program
INCLUDE   ILHSPH31
END

CATALR    ILHSD4

INCLUDE   ILHSPH04
INCLUDE   ILHSPH14
INCLUDE   ILHSPH24   2314 Sort Program
INCLUDE   ILHSPH34
END

CATALR    ILHSNMPS

INCLUDE   ILHSPHN    Required for systems without multiprogramming
                     support
```

Intermediate Processor Generation Books

```
CATALR    ILHSPH0

PHASE     SORT,+0
INCLUDE   ILHSPPI
INCLUDE   ILHSRTMG
PHASE     SORTRCL,*
```

```
INCLUDE   ILHSRCL
INCLUDE   ILHSRCA
INCLUDE   ILHSRCM    Entire Sort/Merge Program
PHASE     SORTRCB,*  Phase 0
INCLUDE   ILHSRCB
INCLUDE   ILHSRCJ
INCLUDE   ILHSRCI
PHASE     SORTRCK,*
INCLUDE   ILHSRCK
PHASE     SORTRCC,*
INCLUDE   ILHSRCC
PHASE     SORTRCD,*
INCLUDE   ILHSRCD
PHASE     SORTRCE,*
INCLUDE   ILHSRCE
PHASE     SORTRCF,*
INCLUDE   ILHSRCF
PHASE     SORTRCH,*
INCLUDE   ILHSRCH
END

CATALR    ILHSPH0S

PHASE     SORT,+0
INCLUDE   ILHSPP1
INCLUDE   ILHSRTMG
PHASE     SORTRCL,*
INCLUDE   ILHSRCL
INCLUDE   ILHSRCA
INCLUDE   ILHSRCM
PHASE     SORTRCB,*  Entire Sort Program
INCLUDE   ILHSRCB    Phase 0
INCLUDE   ILHSRCJ
INCLUDE   ILHSRCI
PHASE     SORTRCK,*
INCLUDE   ILHSRCK
PHASE     SORTRCC,*
INCLUDE   ILHSRCC
PHASE     SORTRCD,*
INCLUDE   ILHSRCD
PHASE     SORTRCE,*
INCLUDE   ILHSRCE
PHASE     SORTRCF,*
INCLUDE   ILHSRCF
END

CATALR    ILHSPH0M
PHASE     SORT,+0
INCLUDE   ILHSPPI
INCLUDE   ILHSRTMG
PHASE     SORTRCL,*
INCLUDE   ILHSRCL
INCLUDE   ILHSRCA    Merge Program
INCLUDE   ILHSRCM    Phase 0
PHASE     SORTRCB,*
INCLUDE   ILHSRCB
PHASE     SORTRCJ,*
INCLUDE   ILHSRCJ
INCLUDE   ILHSRCI
PHASE     SORTRCK,*
INCLUDE   ILHSRCK
PHASE     SORTRCC,*
INCLUDE   ILHSRCC
PHASE     SORTRCD,*
INCLUDE   ILHSRCD
```

```
CATALR    ILHSPHOM
PHASE     SORTRCH,*
INCLUDE   ILHSRCH
END


CATALR    ILHSPHOT

PHASE     SORT,+0
INCLUDE   ILHSPPI
INCLUDE   ILHSRTMG
PHASE     SORTRCL,*
INCLUDE   ILHSRCL
INCLUDE   ILHSRCA
INCLUDE   ILHSRCM
PHASE     SORTRCB,*   2400 Sort Program
INCLUDE   ILHSRCB     Phase 0
PHASE     SORTRCJ,*
INCLUDE   ILHSRCJ
INCLUDE   ILHSRCI
PHASE     SORTRCK,*
INCLUDE   ILHSRCK
PHASE     SORTRCC,*
INCLUDE   ILHSRCC
PHASE     SORTRCD,*
INCLUDE   ILHSRCD
PHASE     SORTRCE,*
INCLUDE   ILHSRCE
END


CATALR    ILHSPH01

PHASE     SORT,+0
INCLUDE   ILHSPPI
INCLUDE   ILHSRTMG
PHASE     SORTRCL,*
INCLUDE   ILHSRCL
INCLUDE   ILHSRCA
INCLUDE   ILHSRCM
PHASE     SORTRCB,*   2311 Sort Program
INCLUDE   ILHSRCB     Phase 0
PHASE     SORTRCJ,*
INCLUDE   ILHSRCJ
INCLUDE   ILHSRCI
PHASE     SORTRCK,*
INCLUDE   ILHSRCK
PHASE     SORTRCC,*
INCLUDE   ILHSRCC
PHASE     SORTRCD,*
INCLUDE   ILHSRCD
PHASE     SORTRCF,*
INCLUDE   ILHSRCF
END
```

```
CATALR    ILHSPH04

PHASE     SORT,+0
INCLUDE   ILHSPPI
INCLUDE   ILHSRTMG
PHASE     SORTRCL,*
INCLUDE   ILHSRCL
INCLUDE   ILHSRCA
INCLUDE   ILHSRCM
PHASE     SORTRCB,*   2314 Sort Program
INCLUDE   ILHSRCB     Phase 0
PHASE     SORTRCJ,*
INCLUDE   ILHSRCJ
INCLUDE   ILHSRCI
PHASE     SORTRCK,*
INCLUDE   ILHSRCK
PHASE     SORTRCC,*
INCLUDE   ILHSRCC
PHASE     SORTRCD,*
INCLUDE   ILHSRCD
PHASE     SORTRCF,*
INCLUDE   ILHSRCF
END

CATALR    ILHSPHN

PHASE     SORT,*      Also required for Phase 0 of all
INCLUDE   ILHSPPI     sort/merge programs in systems
INCLUDE   ILHSRTMG    without multiprogramming support

CATALR    ILHSPH1

PHASE     SORTRCN,*
INCLUDE   ILHSRCN
PHASE     SORTRSD,*
INCLUDE   ILHSRSD
PHASE     SORTRSE,*
INCLUDE   ILHSRSI
INCLUDE   ILHSRSE
INCLUDE   ILHSRMC
PHASE     SORTASA,*   Entire Sort Program
INCLUDE   ILHSASA     Phase 1
PHASE     SORTROA,*
INCLUDE   ILHSROA
PHASE     SORTRDA,*
INCLUDE   ILHSRDA
PHASE     SORTRDB,*
INCLUDE   ILHSRDB
PHASE     SORTRDC,*
INCLUDE   ILHSRDC
PHASE     SORTRAA,*
INCLUDE   ILHSRAA
PHASE     SORTRAB,*
INCLUDE   ILHSRAB
PHASE     SORTRBA,*
INCLUDE   ILHSRBA
PHASE     SORTRBB,*
INCLUDE   ILHSRBB
PHASE     SORTRBC,*
INCLUDE   ILHSRBC
PHASE     SORTRBD,*
INCLUDE   ILHSRBD
PHASE     SORTRGA,*
INCLUDE   ILHSRGA
PHASE     SORTRGB,*
```

```
INCLUDE    ILHSRGB
PHASE      SORTRPA,*
INCLUDE    ILHSRPA
PHASE      SORTRPB,*
INCLUDE    ILHSRPB
END

CATALR     ILHSPH1T

PHASE      SORTRCN,*
INCLUDE    ILHSRCN
PHASE      SORTRSD,*
INCLUDE    ILHSRSD
PHASE      SORTRSE,*
INCLUDE    ILHSRSI
INCLUDE    ILHSRSE
INCLUDE    ILHSRMC
PHASE      SORTASA,*
INCLUDE    ILHSASA
PHASE      SORTROA,*
INCLUDE    ILHSROA     2400 Sort Program
PHASE      SORTRDA,*   Phase 1
INCLUDE    ILHSRDA
PHASE      SORTRDB,*
INCLUDE    ILHSRDB
PHASE      SORTRAA,*
INCLUDE    ILHSRAA
PHASE      SORTRBA,*
INCLUDE    ILHSRBA
PHASE      SORTRBB,*
INCLUDE    ILHSRBB
PHASE      SORTRBD,*
INCLUDE    ILHSRBD
PHASE      SORTRGA,*
INCLUDE    ILHSRGA
PHASE      SORTRGB,*
INCLUDE    ILHSRGB
PHASE      SORTRPA,*
INCLUDE    ILHSRPA
END

CATALR     ILHSPH11

PHASE      SORTRCN,*
INCLUDE    ILHSRCN
PHASE      SORTRSD,*
INCLUDE    ILHSRSD
PHASE      SORTRSE,*
INCLUDE    ILHSRSI
INCLUDE    ILHSRSE
INCLUDE    ILHSRMC
PHASE      SORTASA,*
INCLUDE    ILHSASA
PHASE      SORTROA,*   2311 Sort Program
INCLUDE    ILHSROA     Phase 1
PHASE      SORTRDA,*
INCLUDE    ILHSRDA
PHASE      SORTRDB,*
INCLUDE    ILHSRDB
PHASE      SORTRDC,*
INCLUDE    ILHSRDC
PHASE      SORTRAB,*
INCLUDE    ILHSRAB
PHASE      SORTRBA,*
INCLUDE    ILHSRBA
```

```
CATALR    ILHPH11
PHASE     SORTRBC,*
INCLUDE   ILHSRBC
PHASE     SORTRBD,*
INCLUDE   ILHSRBD
PHASE     SORTRGA,*
INCLUDE   ILHSRGA
PHASE     SORTRGB,*
INCLUDE   ILHSRGB
PHASE     SORTRPB,*
INCLUDE   ILHSRPB
END

CATALR    ILHSPH14

PHASE     SORTRCN,*
INCLUDE   ILHSRCN
PHASE     SORTRSD,*
INCLUDE   ILHSRSD
PHASE     SORTRSE,*
INCLUDE   ILHSRSI
INCLUDE   ILHSRSE
INCLUDE   ILHSRMC
PHASE     SORTASA,*
INCLUDE   ILHSASA
PHASE     SORTROA,*
INCLUDE   ILHSROA      2314 Sort Program
PHASE     SORTRDA,*    Phase 1
INCLUDE   ILHSRDA
PHASE     SORTRDB,*
INCLUDE   ILHSRDB
PHASE     SORTRDC,*
INCLUDE   ILHSRDC
PHASE     SORTRAB,*
INCLUDE   ILHSRAB
PHASE     SORTRBA,*
INCLUDE   ILHSRBA
PHASE     SORTRBC,*
INCLUDE   ILHSRBC
PHASE     SORTRBD,*
INCLUDE   ILHSRBD
PHASE     SORTRGA,*
INCLUDE   ILHSRGA
PHASE     SORTRGB,*
INCLUDE   ILHSRGB
PHASE     SORTRPB,*
INCLUDE   ILHSRPB
END

CATALR    ILHSPH2

PHASE     SORTRSG,*
INCLUDE   ILHSRSG
PHASE     SORTRSH,*
INCLUDE   ILHSRSH
PHASE     SORTRSJ,*
INCLUDE   ILHSRSM
INCLUDE   ILHSRSJ
INCLUDE   ILHSRMC
PHASE     SORTRAC,*
INCLUDE   ILHSRAC
INCLUDE   ILHSRGD
INCLUDE   ILHSRPC      Entire Sort Program
INCLUDE   ILHSAAC      Phase 2
INCLUDE   ILHSASG
```

```
INCLUDE    ILHSAPC
INCLUDE    ILHSAGD
PHASE      SORTRAD,*
INCLUDE    ILHSRAD
INCLUDE    ILHSRGE
INCLUDE    ILHSRPD
INCLUDE    ILHSAAD
INCLUDE    ILHSASG
INCLUDE    ILHSAPD
INCLUDE    ILHSAGE
PHASE      SORTROB,*
INCLUDE    ILHSROB
INCLUDE    ILHSRBF
INCLUDE    ILHSASF
INCLUDE    ILHSAOB
INCLUDE    ILHSABF
END
```

<u>CATALR</u>   <u>ILHSPH2T</u>

```
PHASE      SORTRSG,*
INCLUDE    ILHSRSG
PHASE      SORTRSH,*
INCLUDE    ILHSRSH
PHASE      SORTRSJ,*
INCLUDE    ILHSRSM
INCLUDE    IIHSRSJ      2400 Sort Program
INCLUDE    ILHSRMC      Phase 2
PHASE      SORTRAC,*
INCLUDE    ILHSRAC
INCLUDE    ILHSRGD
INCLUDE    ILHSRPC
INCLUDE    ILHSAAC
INCLUDE    ILHSASG
INCLUDE    ILHSAPC
INCLUDE    ILHSAGD
PHASE      SORTROB,*
INCLUDE    ILHSROB
INCLUDE    ILHSRBF
INCLUDE    ILHSASF
INCLUDE    ILHSAOB
INCLUDE    ILHSABF
END
```

<u>CATALR</u>   <u>ILHSPH21</u>

```
PHASE      SORTRSG,*
INCLUDE    ILHSRSG
PHASE      SORTRSH,*
INCLUDE    ILHSRSH
PHASE      SORTRSJ,*
INCLUDE    ILHSRSM
INCLUDE    ILHSRSJ
INCLUDE    ILHSRMC
PHASE      SORTRAD,*    2311 Sort Program
INCLUDE    ILHSRAD      Phase 2
INCLUDE    ILHSRGE
INCLUDE    ILHSRPD
INCLUDE    ILHSAAD
INCLUDE    ILHSASG
INCLUDE    ILHSAPD
INCLUDE    ILHSAGE
PHASE      SORTROB,*
INCLUDE    ILHSROB
```

```
CATALR    ILHSPH21
INCLUDE   ILHSRBF
INCLUDE   ILHSASF
INCLUDE   ILHSAOB
INCLUDE   ILHSABF
END

CATALR    ILHSPH24

PHASE     SORTRSG,*
INCLUDE   ILHSRSG
PHASE     SORTRSH,*
INCLUDE   ILHSRSH
PHASE     SORTRSJ,*
INCLUDE   ILHSRSM
INCLUDE   ILHSRSJ
INCLUDE   ILHSRMC     2314 Sort Program
PHASE     SORTRAD,*   Phase 2
INCLUDE   ILHSRAD
INCLUDE   ILHSRGE
INCLUDE   ILHSRPD
INCLUDE   ILHSAAD
INCLUDE   ILHSASG
INCLUDE   ILHSAPD
INCLUDE   ILHSAGE
PHASE     SORTROB,*
INCLUDE   ILHSROB
INCLUDE   ILHSRBF
INCLUDE   ILHSASF
INCLUDE   ILHSAOB
INCLUDE   ILHSABF
END

CATALR    ILHSPH3

PHASE     SORTRSM,*
INCLUDE   ILHSRSM
INCLUDE   ILHSRSE
INCLUDE   ILHSRMC
PHASE     SORTRSN,*
INCLUDE   ILHSRSN
INCLUDE   ILHSRMC
PHASE     SORTROC,*
INCLUDE   ILHSROC
INCLUDE   ILHSRBG     Entire Sort/Merge Program
INCLUDE   ILHSRPE     Phase 3
INCLUDE   ILHSAOC
INCLUDE   ILHSABG
INCLUDE   ILHSASK
INCLUDE   ILHSASL
PHASE     SORTRGH,*
INCLUDE   ILHSRGH
INCLUDE   ILHSAGH
INCLUDE   ILHSAPH
PHASE     SORTRGF,*
INCLUDE   ILHSRGF
INCLUDE   ILHSAGF
INCLUDE   ILHSAPF
PHASE     SORTRGG,*
INCLUDE   ILHSRGG
INCLUDE   ILHSAGG
INCLUDE   ILHSAPG
END
```

```
CATALR    ILHSPH3S

PHASE     SORTRSM,*
INCLUDE   ILHSRSM
INCLUDE   ILHSRSE
INCLUDE   ILHSRMC
PHASE     SORTRSN,*
INCLUDE   ILHSRSN
INCLUDE   ILHSRMC
PHASE     SORTROC,*
INCLUDE   ILHSROC    Entire Sort Program
INCLUDE   ILHSRBG    Phase 3
INCLUDE   ILHSRPE
INCLUDE   ILHSAOC
INCLUDE   ILHSABG
INCLUDE   ILHSASK
INCLUDE   ILHSASL
PHASE     SORTRGF,*
INCLUDE   ILHSRGF
INCLUDE   ILHSAGF
INCLUDE   ILHSAPF
PHASE     SORTRGG,*
INCLUDE   ILHSRGG
INCLUDE   ILHSAGG
INCLUDE   ILHSAPG
END

CATALR    ILHSPH3M

PHASE     SORTRCN,*
INCLUDE   ILHSRCN
PHASE     SORTRSM,*
INCLUDE   ILHSRSM
INCLUDE   ILHSRSE
INCLUDE   ILHSRMC
PHASE     SORTRSN,*
INCLUDE   ILHSRSN
INCLUDE   ILHSRMC
PHASE     SORTROC,*
INCLUDE   ILHSROC    Merge Program
INCLUDE   ILHSRBG    Phase 3
INCLUDE   ILHSRPE
INCLUDE   ILHSACC
INCLUDE   ILHSABG
INCLUDE   ILHSASK
INCLUDE   ILHSASL
PHASE     SORTRGH,*
INCLUDE   ILHSRGH
INCLUDE   ILHSAGH
INCLUDE   ILHSAPH
END

CATALR    ILHSPH3T

PHASE     SORTRSM,*
INCLUDE   ILHSRSM
INCLUDE   ILHSRSE
INCLUDE   ILHSRMC
PHASE     SORTRSN,*
INCLUDE   ILHSRSN
INCLUDE   ILHSRMC
PHASE     SORTROC,*
INCLUDE   ILHSROC    2400 Sort Program
INCLUDE   ILHSRBG    Phase 3
INCLUDE   ILHSRPE
```

```
INCLUDE    ILHSAOC
INCLUDE    ILHSABG
INCLUDE    ILHSASK
INCLUDE    ILHSASL
PHASE      SORTRGF,*
INCLUDE    ILHSRGF
INCLUDE    ILHSAGF
INCLUDE    ILHSAPF
END

CATALR     ILHSPH31

PHASE      SORTRSM,*
INCLUDE    ILHSRSM
INCLUDE    ILHSRSE
INCLUDE    ILHSRMC
PHASE      SORTRSN,*
INCLUDE    ILHSRSN
INCLUDE    ILHSRMC     2311 Sort Program
PHASE      SORTROC,*   Phase 3
INCLUDE    ILHSROC
INCLUDE    ILHSRBG
INCLUDE    ILHSRPE
INCLUDE    ILHSAOC
INCLUDE    ILHSABG
INCLUDE    ILHSASK
INCLUDE    ILHSASL
PHASE      SORTRGF,*
INCLUDE    ILHSRGF
INCLUDE    ILHSAGF
INCLUDE    ILHSAPF
PHASE      SORTRGG,*
INCLUDE    ILHSRGG
INCLUDE    ILHSAGG
INCLUDE    ILHSAPG
END

CATALR     ILHSPH34

PHASE      SORTRSM,*
INCLUDE    ILHSRSM
INCLUDE    ILHSRSE
INCLUDE    ILHSRMC
PHASE      SORTRSM,*
INCLUDE    ILHSPSN
INCLUDE    ILHSRMC
PHASE      SORTROC,*
INCLUDE    ILHSROC
INCLUDE    ILHSRBG
INCLUDE    ILHSRPE
INCLUDE    ILHSAOC     2314 Sort Program
INCLUDE    ILHSABG     Phase 3
INCLUDE    ILHSASK
INCLUDE    ILHSASL
PHASE      SORTRGF,*
INCLUDE    ILHSRGF
INCLUDE    ILHSAGF
INCLUDE    ILHSAPF
PHASE      SORTRGG,*
INCLUDE    ILHSRGG
INCLUDE    ILHSAGG
INCLUDE    ILHSAPG
END
```

Phase Names and Relocatable Module Names

Phase Name	CSECT (Module Name	Required	Program Phase
SCRT	ILHSPPI	A	0-3
	ILHSRTMG	A	0
SORTRCA	ILHSRCL	A	0
	ILHSRCA	A	0
	ILHSRCM	A	0
SORTRCB	ILHSRCB	A	0
	ILHSRCI	A	0
	ILHSRCJ	A	0
SORTRCK	ILHSRCK	A	0
SORTRCC	ILHSRCC	A	0
SORTRCD	ILHSRCD	A	0
SORTRCE	ILHSRCE	T	0
SORTRCF	ILHSRCF	D	0
SORTRCH	ILHSRCH	M	0
SORTRCN	ILHSRCN	A	1,2,3
SORTRCD	ILHSRSD	S	1
SORTRSE	ILHSRSI	S	1
	ILHSRSE	S	1
	ILHSRMC	S	1
SORTASA	ILHSASA	S	1
SORTROA	ILHSROA	S	1
	ILHSAOA	S	1
SORTRDA	ILHSRDA	S	1
	ILHSADA	S	1
SORTRDB	ILHSRDB	S	1
	ILHSADB	S	1
SORTRDC	ILHSRDC	D	1
	ILHSADC	D	1
SORTRAA	ILHSRAA	T	1
	ILHSAAA	T	1
SORTRAB	ILHSRAB	D	1
	ILHSAAB	D	1
SORTRBA	ILHSRBA	S	1
	ILHSABA	S	1
SORTRBB	ILHSRBB	T	1
	ILHSABB	T	1
SORTRBC	ILHSRBC	D	1
	ILHSABC	D	1
SORTRBD	ILHSRBD	S	1
	ILHSABD	S	1
SORTRGA	ILHSRGA	S	1
	ILHSAGA	S	1
SORTRGB	ILHSRGB	S	1
	ILHSAGB	S	1
SORTRPA	ILHSRPA	T	1
	ILHSAPA	T	1
SORTRPB	ILHSRPB	D	1
	ILHSAPB	D	1
SORTRSG	ILHSRSG	S	2
SORTRSH	ILHSRSH	S	2
SORTRSJ	ILHSRSM	S	2
	ILHSRSJ	S	2
	ILHSRMC	S	2
SORTRAC	ILHSRAC	T	2
	ILHSRGD	T	2
	ILHSRPC	T	2
	ILHSAAC	T	2
	ILHSASG	S	2
	ILHSAPC	T	2

	ILHSAGD	T	2
SORTRAD	ILHSRAD	D	2
	ILHSRGE	D	2
	ILHSRPD	D	2
	ILHSAAD	D	2
	ILHSASG	S	2
	ILHSAPD	D	2
	ILHSAGE	D	2
SORTROB	ILHSROB	S	2
	ILHSRBF	S	2
	ILHSASF	S	2
	ILHSAOB	S	2
	ILHSABF	S	2
SORTRSM	ILHSRSM	M	3
SORTRSN	ILHSRSN	A	3
	ILHSRMC	A	3
SORTROC	ILHSROC	A	3
	ILHSRBG	A	3
	ILHSRPE	A	3
	ILHSAOC	A	3
	ILHSABG	A	3
	ILHSASK	A	3
	ILHSASL	A	3
SORTRGH	ILHSRGH	M	3
	ILHSAGH	M	3
	ILHSAPH	M	3
SORTRGF	ILHSRGF	S	3
	ILHSAGF	S	3
	ILHSAPF	S	3
SORTRGG	ILHSRGG	D	3
	ILHSAGG	D	3
	ILHSAPG	D	3

KEY

A - All programs
D - Direct access (2311,2314) Sort Programs
M - Merge Program
S - Entire Sort Program
T - 2400 Sort Program

Appendix F. BPS Device Type Specifications (DD)

The following device type codes can be entered in the Basic Programming Support ASSGN cards:

C1 1052 Printer-Keyboard

D1 2311 Disk Drive

D3 2314 Disk Drive

L1 1403 or 1404 Printer

L2 1443 or 1445 Printer

P1 2540 Card Read-Punch (punching only)

P2 1442 Card Read-Punch (punching only)

P3 2520 Card Read-Punch (punching only)

R0 2671 Paper Tape Reader

R1 2540 Card Read-Punch (reading only)

R2 2540 Using Punch-Read-Feed feature

R3 1442 Card Read-Punch (reading or reading and punching for combined files)

R4 2501 Card Reader

R5 2520 Card Read-Punch (reading or reading and punching for combined files)

RR 1285 Optical Reader

ST STR devices attached to 2701

T1 2400 7-track Tape

T2 2400 9-track Tape

Appendix G. Storage Requirements

Internal Storage Requirements

This section contains the data required for estimating three values:

1. The size of the supervisor required for a generated system,

2. The amount of main storage required at object-time for the supervisor and IOCS macro instructions, and

3. The internal storage requirements for relocatable subroutines required by COBOL, FORTRAN, Basic FORTRAN, and PL/I.

The supervisor size can be used to determine the size of the available problem-program area. Note that the background problem-program area must be at least 10,240 bytes when the Disk Operating System is used. However, if the COBOL compiler or the Assembler with both disk and tape work files is a part of the system, the minimum background area must be 14,336 bytes. If Assembler F is part of the system, the minimum background area must be 45,056 bytes. The supervisor varies in size from system to system according to the options chosen by the user and to the machine configuration.

The storage estimates shown in this publication are within 15% variance of actual requirements.

Supervisor

Figure 50 gives the main storage requirements for the base supervisors and the elements that can be included in a tailored supervisor. The base supervisor requirement for the batched job system is 5,902 bytes. Additional storage requirements must be added to the base requirement for each additional supervisor element desired that is not within the base requirement. For example, for OC=YES (FOPT) under batched job system, add 176 bytes to the base storage requirement (5,902 bytes). The base requirement for the MPS=YES supervisor is 6,870 bytes. For basic Telecommunication under MPS=YES, i.e. TP = BTAM, add 280 bytes to the base requirement for the MPS supervisor, thus an MPS supervisor that includes basic Telecommunication requires 7,150 bytes (see Figure 50). (Note that QTAM requires an MPS supervisor.) Thus, by the time the supervisor is tailored to the installation requirements, it usually is larger than the base requirements.

Note the relationship between the actual number of bytes in the 6K and 8K supervisors shipped by IBM and the number of bytes specified in the SEND macro for the respective supervisor.

Size of Supervisor Shipped by IBM (in bytes)	Address Specified In SEND Macro (in bytes)
6,144	6,144
6,592	8,192

The SEND macro specifies the beginning of the problem program area to facilitate system generation. Depending on the combination of supervisory functions chosen, a supervisor greater than 6,144 bytes may be generated.

Note: All supervisor generation options are described in the section Macro Instructions For Generating a Supervisor.

Supervisor Element	Generation Operand	MPS=		
		NO	YES	BJF
Required Routines (Basic Size)-SUPVR SVC Interruption Handling System Loader (FETCH and LOAD) I/O Units Control Tables (LUBS, PUBS, and JIBS) General Entry and Exit Routines Communication Region Transient Area End of Job Step Physical IOCS (including Selector Channel Support) Storage Protection (with MP)	SYSTEM=DISK	5902	6870	7166
Optional Routines				
Magnetic Character Options	MICR=1412 or 1419 (for 1259 see Note 1)	1440	1524	1524
	MICR=1419D	1284	1384	1384
Teleprocessing Options				
BTAM	TP=BTAM	376	280	280
QTAM	TP=QTAMn(Note 2)	--	872+[A]	872+[A]
Multitasking	AP=YES (Includes WAITM=YES)	--	2256	2256
1401/1440/1460 Emulator	EU=YES			
	IBM Model 2030	200	192	184
	IBM Model 2040	136	120	112

Figure 50. Supervisor Main Storage Requirements (Bytes) (Part 1 of 6)

Supervisor Element				Generation Operand	MPS= NO	YES	BJF
Configuration Options-CONFG							
Model 30				MODEL=30	0	0	0
Model 40				MODEL=40			
Model 50				MODEL=50			0
Storage Protection				SP=YES	112	0	0
Decimal Feature				DEC=YES	0	0	0
Floating Point Feature				FP=YES	0	112	112
Timer Feature				TIMER=YES (with AP=YES, add 50 bytes)	64	60	64
				TIMER=YES (with any MICR support)	160	156	156
Functional Supervisor Options-FOPT	User Option to Handle						
	Operator Comm.	Prog Ck	Int Timer				
	X			OC=YES	176	160	152
		X		PC=YES	128	200	200
			X	IT=BG,F1,F2	420	416	392
	X		X	Note: If AP=YES when	424	408	408
		X	X	PC=YES, add 150	440	432	432
	X	X		bytes.	264	264	264
	X	X	X	If OC=YES when any MICR support is included, add 50 bytes.	464	472	472
CE Serviceability Programs				CE=YES (Note 3) CE=n (Note 9)	864 n+224	864 n+224	850 n+210
Seek Separation				SKSEP=YES or n (Note 4)	288+4(n) + [a]+[b]	288+4(n) + [a]+[b]	288+4(n) + [a]+[b]
Physical Transient Overlap				PTO=YES (Note 5)	--	416	424
Console Buffering				CBF=n n may be 1 to 9 buffers	554+106n	602+106n	602+106n
				(If CHANQ is not elected--Note 6)	546+106n	602+106n	602+106n

Figure 50. Supervisor Main Storage Requirements (Bytes) (Part 2 of 6)

Supervisor Element	Generation Operand	MPS=		
		NO	YES	BJF
Functional Supervisor Options - FOPT (continued)				
Multiple Wait	WAITM=YES (Note 7)	40	40	40
Abnormal Termination	AB=YES (without AP=YES)	472	456	456
	AB=YES (with AP=YES)	--	600	600
Track Hold	TRKHLD=n (without AP=YES) (Note 8)	--	660 +12n	660 +12n
	TRKHLD=n (with AP=YES) (Note 8)	--	750 +12n	750 +12n
Tape Error Statistics	TEB=n	42+8n	34+8n	34+8n
Command Chaining	CCHAIN=YES	24	24	24
Disk System Input and Output Files	SYSFIL=(2311 or 2314 [, n_1, n_2])	424	440	440
DASD File Protection (with 2321)	DASDFP= n_1, n_2, dev	528+24x (n_2-n_1)	448+24x (n_2-n_1)	440+24x (n_2-n_1)
(with Disk)		456+24x (n_2-n_1)	368+24x (n_2-n_1)	384+24x (n_2-n_1)
DASDFP (with 2321) <u>and</u> Disk SYSFIL		912+24x (n_2-n_1)	846+24x (n_2-n_1)	968+24x (n_2-n_1)
DASDFP (with 2311) <u>and</u> Disk SYSFIL		848+24x (n_2-n_1)	740+24x (n_2-n_1)	908+24x (n_2-n_1)

Figure 50. Supervisor Main Storage Requirements (Bytes) (Part 3 of 6)

Supervisor Element	Generation Operand	MPS		
		NO	YES	BJF
Job Control Options-STDJC (Job Control options affect only the contents of the communications region, not its size.)				
Physical IOCS Support-PIOCS				
Selector Channel Support	SELCH=YES	16	16	16
Burst Mode on Multiplexor Channel	BMPX=YES	48	48	48
Channel Switching Tape Control				
2404 or 2804	CHANSW=RWTAU	40	32	32
2816 only	CHANSW=TSWTCH	40	32	32
Tape Support				
7-track and 9-track, or 7-track only	TAPE=7	64	48	48
9-track only	TAPE=9	56	16	16
Allocate	ALLOC	0	0	0
Input/Output Tables-IOTAB				
Number of I/O devices on system	IODEV=n	8(n-10)	8(n-10)	8(n-10)
Number of programmer logical units	BGPGR=n	2(n-10)	2(n-10)	2(n-10)
	F2PGR=n		2(n-5)	
	F1PGR=n		2(n-5)	
Number of Channel Queue Entries	CHANQ=n (Note 6)	6(n-6)	6(n-6)	6(n-6)
Number of Job Information Blocks	JIB=n	4(n-5)	4(n-5)	4(n-5)

Figure 50. Supervisor Main Storage Requirements (Bytes) (Part 4 of 6)

Note 1. A 1259 is addressed as a 1419 single address adapter machine i.e., MICR=1419.

Note 2. MPS=YES or BJF is required for TP=QTAMn. TP=QTAMn includes BTAM supervisor support.

Telecommunications requires a minimum of two channels: one for the telecommunications; the other for the system resident device. An IBM 2701 Line Adapter Unit attached to an IBM System/360, Model 25, must be placed on the multiplexor channel.

If AP=YES when TP=QTAMn, then the quantity A must be added to the basic storage requirement for TP=QTAMn. A=44+(n-2)12, where n is the value elected for TP=QTAMn.

Note 3. If PTO=YES when CE=YES or CE=n, add 8 bytes to the CE storage requirement. The number of bytes indicated is the current storage requirement. In the future, these storage requirements may increase.

Note 4. When SKSEP=YES, n (in the formula) equals the number of DASD devices specified at system generation time. When SKSEP=n, n (in the formula) is the number of DASD devices supported as specified, but cannot be less than the number specified at system generation time. In either case, a (in the formula) is the 8 bytes required if DASDFP and/or SYSFIL options are selected, and b (in the formula) is the 8 bytes required if Teleprocessing (TP) option is selected.

Note 5. PTO=YES requires that MPS=YES or MPS=BJF for Physical Transient Overlap (PTO) support to be generated. For a description of Physical Transient Overlap (PTO), refer to IBM Systems/360 DOS Timing Estimates listed on the cover of this manual.

Note 6. The selection of the CBF option results in extra channel queue usage. Consider this when requesting the number of CHANQ entries. Thus, specification of the CBF option and election of the CHANQ default, which is 6 channel queue entries, results in the number of buffers specified being added to the CHANQ default. However, when both the CBF and CHANQ options are specified, the number of CHANQ entries desired should be increased by the number of buffers specified. Otherwise, the number of entries generated in the channel queue will be less than desired.

Figure 50. Supervisor Main Storage Requirements (Bytes) (Part 5 of 6)

Note 7. WAITM=YES is assumed when AP=YES.

Note 8. Where n equals the maximum number of tracks (1-255) to be held at any given time by the entire system. The default is 10 if n is an invalid parameter (non-numeric or outside the range, 1-255).

Note 9. If PTO=YES when CE=n, add 8 bytes to the CE storage requirement. In addition, if any of the following options are elected along with CE, subtract 8 bytes from the CE storage requirement under MPS=NO, or MPS=YES and subtract 24 bytes from the CE storage requirement under MPS=BJF.

- Multitasking (AP=YES)
- Track Hold (TRKHLD=n)
- Abnormal Termination (AB=YES)

The number of bytes indicated is the current storage requirement, where n is a minimum of 600 bytes. In the future, these storage requirements may increase.

Figure 50. Supervisor Main Storage Requirements (Bytes) (Part 6 of 6)

Supervisor size increases are not necessarily linear. For example, compare the size requirements of the separate entries OC=YES and PC=YES with the combined entry OC=YES,PC=YES. Combinations of elements may result in an actual supervisor size that is smaller than the calculated total derived for the same supervisor.

The I/O unit control tables in the IBM-supplied supervisor contain entries for up to 10 physical units and the first 10 programmer logical units. (See the IBM System/360 DOS System Control and System Service Programs publication listed on the front cover of this publication for a discussion of these tables.) The I/O table provides six channel queue positions and five job information blocks (JIB). As a minimum, 12 I/O devices and program check interrupts can be included within a 6,144 byte supervisor.

Because storage is protected in increments of 2K and other considerations, a DOS supervisor generated with any of the following options (with the exception of multitasking which forces the storage protection boundary to 10,240 bytes) requires a minimum of 8,192 bytes:

- Multiprogramming
- Telecommunications
- DASD File Protection
- Disk System Input/Output
- 1259/1412/1419 Magnetic Character Readers

If foreground areas are not used, the batch-job supervisor would probably be more useful because the multiprogramming supervisor requires additional space and time to perform its functions.

Computing the Size of a Supervisor

As an example, assume a supervisor is generated using the macros shown in Figure 51. The size of this supervisor is determined as illustrated.

GENERATION OPERAND		MAIN STORAGE REQUIREMENT (bytes)
SUPVR		
SYSTEM=DISK		6870
MPS=YES		
CONFG		
MODEL=40		0
SP=YES		0
DEC=YES		0
FP=YES		112
TIMER=YES		60
STDJC		0
FOPT		
IT=BG		472
PC=YES		
OC=YES		
CCHAIN=YES		24
DASDFP=(1,2,2321)	[846 + 24(2-1)] =	870
SYSFIL=2311		
TEB=4	34 + 8(4) =	66
PIOCS		
CHANSW=RWTAU		32
TAPE=7		48
BMPX=YES		48
IOTAB		
JIB=10	4(n-5) =	20
CHANQ=10	6(n-6) =	24
F1PGR=8	2(n-5) =	6
F2PGR=8	2(n-5) =	6
IODEV=10	8(n-10) =	0
BGPGR=10	2(n-10) =	0 (Note 1)
Total Number of Bytes		8658

Note 1: If entry for BGPGR is below minimum requirement, n=10 is assumed. The following MNOTE is obtained for an entry of less than 10.

BGPGR SPECIFICATION BELOW MINIMUM - "10" ASSUMED

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISKS AND AT LEAST ONE TAPE

```
// EXEC ASSEMBLY
         SUPVR SYSTEM=DISK,MPS=YES
         CONFG MODEL=40,SP=YES,DEC=YES,FP=YES,TIMER=YES
         STDJC LISTX=YES,LINES=46
         FOPT  IT=BG,PC=YES,OC=YES,CCHAIN=YES,DASDFP=(1,2,2321),         X
               SYSFIL=(2311),TEB=4
         PIOCS CHANSW=RWTAU,TAPE=7,BMPX=YES
         ALLOC F1=6K,F2=6K
         IOTAB JIB=10,CHANQ=10,F1PGR=8,F2PGR=8,IODEV=10,BGPGR=10
         DVCGEN CHUN=X'00C',DVCTYP=2540R
         DVCGEN CHUN=X'00D',DVCTYP=2540P
         DVCGEN CHUN=X'00E',DVCTYP=1403
         DVCGEN CHUN=X'01F',DVCTYP=1050A
         DVCGEN CHUN=X'190',DVCTYP=2311
         DVCGEN CHUN=X'191',DVCTYP=2311
         DVCGEN CHUN=X'192',DVCTYP=2321
         DVCGEN CHUN=X'180',DVCTYP=2400T7,CHANSW=YES
         DVCGEN CHUN=X'181',DVCTYP=2400T9,CHANSW=YES
```

IBM System/360 Assembler Coding Form

PROGRAM: EXAMPLE ONE: TWO DISKS AND AT LEAST ONE TAPE

```
         ASSGN SYSRDR,X'00C'
         ASSGN SYSIPT,X'00C'
         ASSGN SYSPCH,X'00D'
         ASSGN SYSLST,X'00E'
         ASSGN SYSLOG,X'01F'
         ASSGN SYSLNK,X'191'
         ASSGN SYS001,X'191'
         ASSGN SYS002,X'191'
         ASSGN SYS003,X'191'
         ASSGN SYS004,X'191'
         ASSGN SYS007,X'192'
         SEND  10240
 END
/*
*    CHECK ASSEMBLY LISTING FOR ERRORS.  IF CORRECT
* REMOVE ASSEMBLED SUPERVISOR FROM SYSPCH.  INSERT IN READER
* FOLLOWING THE INCLUDE CARD.
// PAUSE TO CONTINUE PRESS EOB
```

Figure 51. Example of a Supervisor

Supervisor Macro Instructions

Figure 52 lists the main storage requirements for the expansion of the supervisor macro instructions. To determine the amount of main storage required for a given macro expansion, add both literal and variable requirements to the basic requirement.

Macro Instruction Name	NUMBER OF BYTES		
	Basic Storage Requirement	Additional Storage Required For Literals	Variable Requirements*
ATTACH	18 - 32	0 - 4	
CALL	2 - 16 *	0 - 4 *	+4 per operand
CANCEL**	4 - 6		
CHKPT	38 - 42 *		
COMRG	6		
DEQ	4 - 8	0 - 4	
DETACH	2 - 6	0 - 4	
DUMP	6	8	
ENQ	4 - 8	0 - 4	
EOJ	2		
EXIT	2		
FETCH	2 - 10 *	8 - 12 *	
GETIME	10 - 94 *	0 - 8 *	
LOAD	2 - 10 *	8 - 12 *	
MVCOM	12 - 16 *	0 - 4 *	
PDUMP	10 - 28	0 - 16 *	
POST	4 - 8	0 - 8	
RCB	8 - 10		
RETURN	2 - 6 *		
SAVE	4		
SETIME	6 - 14 *	0 - 8 *	
STXIT	2 - 14 *	4 - 12 *	
TECB	4		
WAIT (for TECB)	10 - 14 *	0 - 4 *	
WAITM	4 - 72		

* These values vary according to the selected options.

**Also; if user specifies CANCEL ALL, CANCEL increases two bytes.

Figure 52. Supervisor Macro Instructions Storage Requirements

Foreground Save Areas

A part of each foreground area is reserved for saving registers and for processing labels. The basic save-area size is 88 bytes. The following require additional space:

+32 bytes for floating-point registers

+84 bytes for nonsequential disk labels

+80 bytes for standard tape labels.

MPS Utility Macro Instructions

Figure 53 gives the main storage requirements for the MPS utility macro instructions. These macros are designed so that a file-to-file utility program can fit in a 2K foreground program area (4K if the INTCR macro is used). To determine the size of the macro when expanded, add the variable requirements to, or subtract them from, the basic expansion requirement.

IOCS Declarative Macro Instructions

Figures 54-73 in this section give the main storage requirements for the IOCS declarative macro instructions. Where applicable, both the DTF table requirements and the logic module requirements are given. (See the Supervisor and I/O Macros publication listed on the front cover of this publication.)

The assembled tables and modules are included in the object program when it is linkage edited. The total storage requirement can be determined by adding the appropriate basic module requirement and the storage requirements for optional functions (for example, READ=BACK for magnetic tape files) to the table requirements for each file.

The core sizes in the tables may vary slightly according to the starting address of assembly; that is, boundary alignments may increase the core requirements for the various modules by 2-7 bytes.

	INTAPE	OUTAPE
Basic Macro Expansion	776	718
BUFSIZ=		
80	+0	+0
n	+2(n-80)	+2(n-80)
RECSIZ=		
n	226	
name	260	
FILE=		
name	-24	140
(r)	-12	120
LBL=		
name	324	+276
ERROR=		
SKIP	+8	
IGNORE	0	
name	+72	
CHKPT=		
NO	-26	
name	+28	
RETURN=		
NO		-6
YES		+0
BLK=		
n		+150
name		+182

	INDISK	OUTDISK
Basic Macro Expansion	816	1140
BUFSIZ=		
80	+0	+0
n	+2(n-80)	+2(n-80)
FILE=		
name		+0
(r)	+24	+12
LBL=		
name	+36	+36
RECSIZ=		
n	260	
name	268	
ERROR=		
SKIP	+24	
IGNORE	+0	0
name	+100	148
RETURN=		
NO		0
YES		+0
FORMAT=		
name		+68
FULL		+0
n		44
BLK=		
n		150
name		182

Figure 53. MPS Utility Macro Storage Requirements (Part 1 of 3)

	INLOG	OUTLOG
Basic Macro Expansion	168	170
BUFFER=		
name	+0	+0
(r)	+4	0
COUNT=		
n	+ 0	0
(r)	-28	+4
RETURN=		
NO		-2
YES		+0

	INCARD	OUTCARD
Basic Macro Expansion	418	444
STCTL=		
YES		+60
name		+112
DEVICE=		
2540		+186
2520		+48
RETURN=		
NO		+4
YES		+0

	OUTPRT
Basic Macro Expansion	622
BUFSIZ=	
n	+2(n-144)
RETURN=	
NO	+10
YES	+0
FORMS=	
A	+28
B	+144
C	+114
D	+96

Figure 53. MPS Utility Macro Storage Requirements (Part 2 of 3)

INTCR				
	INPUT= MTST,STDUC or MST,STDLC	INPUT= MTST,name1	INPUT= MTST,NOTRAN or MTDI,NOEDIT	INPUT= MTDI,EDIT or MTDI,EDITR
Basic Macro Expansion	1158	1023	570	1720
BUFSIZ= (n) (n,m1)	 +3n +2n+m1	 +3n +2n+m1	 +3n +2n+m1	 +3n +2n+m1
RECFORM= VAR UNDEF	 +0 -16	 +0 -16	 +0 -16	 +0 -16
ERROPT= IGNORE name2	 +0 +106	 +0 +106	 +0 +106	 +0 +134

Figure 53. MPS Utility Macro Storage Requirements (Part 3 of 3)

DTFCD (DEFINE THE FILE: CARD)

Table Requirements

TYPEFLE=INPUT requires 50 bytes.

TYPEFLE=CMBND requires 84 bytes.

TYPEFLE=OUTPUT requires 48 bytes.

CRDERR=RETRY (2540 only) requires an additional 88 bytes.

DEVICE=2520 requires 8 additional bytes.

			TYPEFLE					
			INPUT	OUTPUT			CMBND	
RECFORM=	IOAREA2=	WORKA=	1442 2520 2540 2501	1442	2520	2540	1442	2520 2540
FIXUNB	-	-	90	74	80	40	126	198
FIXUNB	-	YES	106	116	122	84	154	226
FIXUNB	YES	-	122	118	124	82	126	198
FIXUNB	YES	YES	138	132	138	96	154	226
UNDEF	-	-	-	112	110	70	-	-
UNDEF	-	YES	-	124	132	92	-	-
UNDEF	YES	-	-	128	130	90	-	-
UNDEF	YES	YES	-	140	146	104	-	-
VARUNB	-	-	-	126	132	94	-	-
VARUNB	-	YES	-	140	146	108	-	-
VARUNB	YES	-	-	154	160	118	-	-
VARUNB	YES	YES	-	156	162	120	-	-

Figure 54. CDMOD Main Storage Requirements (Part 1 of 2)

<u>Notes</u>:

1. CTLCHR=YES or ASA; depending upon record format, number of I/O areas, and/or work area specifications:

   for YES, a minimum of 8 to a maximum of 36 additional bytes are required.

   for ASA, a minimum of 28 to a maximum of 65 additional bytes are required.

2. RDONLY=YES changes the size of the modules -50 to +50 bytes. In addition, the user's program must provide a 72-byte save area each time the module is reentered.

3. Part 2 of Figure 54 shows the requirements for the other CDMOD options. The values selected must be added to the TYPEFLE value to determine the amount of storage needed.

CRDERR=RETRY	2520	2540
without IOAREA2 or WORKA	+73	+123
with either or both	+69	+119

CONTROL=YES	1442	2540	2520
INPUT	+24	+30	+30
OUTPUT without WORKA	+24	+20	+12
OUTPUT with WORKA	+24	+32	+12
CMBND without WORKA	+24	+20	+12
CMBND with WORKA	+32	+32	+12

Figure 54. CDMOD Main Storage Requirements (Part 2 of 2)

DTFCN (DEFINE THE FILE: CONSOLE)

		TYPEFLE=	
RECFORM=	WORKA=	INPUT/OUTPUT	OUTPUT only
FIXUNB	-	94	60
FIXUNB	YES	150	90
UNDEF	-	156	114
UNDEF	YES	262	168

Note: No module is required for this macro instruction.

Figure 55. DTFCN Main Storage Requirements

DTFDA (DEFINE THE FILE: DIRECT ACCESS DEVICE)

Table Requirements

RECFORM=FIXUNB requires 205-225 bytes, depending upon imperative macros used in the DTF.

VERIFY=YES requires 40-80 bytes, depending upon imperative macros used in the DTF.

AFTER=YES requires 80 additional bytes.

RECFORM=UNDEF requires 265-285 bytes, depending upon imperative macros used in the DTF.

VERIFY=YES requires 40-80 additional bytes, depending upon imperative macros used in the DTF.

AFTER=YES requires 16 additional bytes.

Relative addressing increases the size of the DTF 60-80 bytes plus 8 bytes per extent. (See IBM System/360 Disk Operating System, Supervisor and Input/Output Macros listed on the front cover of this manual.)

DAMOD (DIRECT ACCESS DEVICE MODULE)

			Formatting Module				
RECFORM=	Basic Module	IDLOC	AFTER	AFTER and IDLOC	RELTRK	HOLD	ERREXT
FIXUNB	392	+216	+304	+520	+208	+68	+28
UNDEF	556	+228	+208	+436	+212	+72	+28

Notes:

1. Basic Module includes coding to handle either FIXUNB or UNDEF records and the WRITEKY, READKEY, READID, WRITEID, SRCHM, VERIFY, and CONTROL functions.

2. AFTER includes coding to create the file and to handle the RZERO option.

3. IDLOC includes coding to return the record identifier to the user in a location he specifies.

4. Specification of trailer label processing in the DTF increases the size of each module by 50 ± 20 bytes.

5. RDONLY=YES changes the size of the module -50 to +50 bytes. In addition, the user's program must provide a 72-byte save area each time the module is reentered.

Figure 56. DAMOD Main Storage Requirements

DTFIS (DEFINE THE FILE: INDEXED SEQUENTIAL)

Table Requirements

1. IOROUT=LOAD requires 248 bytes plus 4 bytes per disk extent specified, plus 8 bytes for IOAREA2.

2. IOROUT=ADD requires 530 bytes plus 4 bytes per disk extent specified plus KL (the length of the key).

3. IOROUT=RETRVE requires 276 bytes plus 4 bytes per disk extent specified, when TYPEFLE=SEQNTL.

4. IOROUT=RETRVE requires 292 bytes plus 4 bytes per disk extent specified when TYPEFLE=RANDOM or RANSEQ.

5. IOROUT=ADDRTR requires 548 bytes plus 4 bytes per disk extent specified plus KL (the length of the key).

6. IOROUT=ADDRTR, TYPEFLE=RANDOM, INDAREA=name, and INDSIZE=n require 300 bytes plus 4 bytes per disk extent.

ISMOD (INDEXED SEQUENTIAL MODULE)

	IOROUT =								
	LOAD					ADD			
RECFORM=		ERREXT	IOAREA2	ERREXT IOAREA2			CORDATA	ERREXT	CORDATA ERREXT
FIXUNB						2572	+184	+476	+660
FIXBLK						2786	+272	+498	+770
BOTH	811	+224	+212	+436		3032	+246	+428	+674

• Figure 57. ISMOD Main Storage Requirements (Part 1 of 5)

	IOROUT=RETRVE							
	TYPEFLE=							
	RANDOM		SEQNTL				RANSEQ	
RECFORM=		ERREXT		ERREXT	IOAREA2	ERREXT IOAREA2		ERREXT
FIXUNB								
FIXBLK								
BOTH	1276	+302	1260	+246	+386	+632	2116	+332

• Figure 57. ISMOD Main Storage Requirements (Part 2 of 5)

	IOROUT=ADDRTR								
	TYPEFLE=								
	RANDOM				SEQNTL				
RECFORM=		CORDATA	ERREXT	CORDATA ERREXT		CORDATA	IOAREA2	CORDATA IOAREA2	ERREXT
FIXUNB	3216	+184	+534	+718	3420	+94	+386	+480	+510
FIXBLK	3432	+272	+554	+826	3634	+272	+386	+658	+528
BOTH	3676	+246	+588	+834	3880	+274	+386	+660	+558

Figure 57. ISMOD Main Storage Requirements (Part 3 of 5)

	IOROUT=ADDRTR			
	TYPEFLE=RANSEQ			
RECFORM=		CORDATA	ERREXT	CORDATA ERREXT
FIXUNB	4056	+194	+556	+750
FIXBLK	4272	+272	+584	+856
BOTH	4516	+248	+616	+864

Figure 57. ISMOD Main Storage Requirements (Part 4 of 5)

Note 1:	When RECFORM=BOTH is specified, the module processes FIXUNB and FIXBLK records.
Note 2:	For CORINDX = YES, add 212 bytes.
Note 3:	RDONLY=YES changes the module size by ±50 bytes with the following exceptions. When IOROUT=ADD or IOROUT=ADDRTR, the module changes in size +60 to +100 bytes. In addition, the user's program must provide a 72-byte save area each time the module is reentered, regardless of function.

Figure 57. ISMOD Main Storage Requirements (Part 5 of 5)

DTFMR (DEFINE THE FILE: MAGNETIC CHARACTER READER)

Table Requirements

If ADDRESS=DUAL is specified, the table requires 264 bytes.

If ADDRESS=DUAL is not specified, the table requires 250 bytes.

MRMOD (MAGNETIC CHARACTER READER MODULE)

If ADDRESS=DUAL is specified, the module requires 1,050 bytes.

If ADDRESS=DUAL is not specified, the module requires 946 bytes.

DTFDI (DEFINE THE FILE: DEVICE INDEPENDENT SYSTEM UNITS)

Table Requirements

240 bytes

DIMOD (DEVICE INDEPENDENT SYSTEM UNITS MODULES)

TYPEFLE=	Basic Module	IOAREA2=YES
INPUT	308	+60
OUTPUT	643	+80

Note: RDONLY=YES changes the size of the module -50 to +50 bytes. In addition, the user's program must provide a 72-byte save area each time the module is reentered.

Figure 58. DIMOD Main Storage Requirements

DTFMT (DEFINE THE FILE: MAGNETIC TAPE)

Table Requirements

TYPEFLE=WORK requires 48 bytes per work file.

The table requirements for INPUT and OUTPUT files are:

TYPEFILE=	RECFORM =	Basic Size Without STDLABELS	ERROPT, ERREXT (Without STDLABELS)	Basic Size With STDLABELS	ERROPT, ERREXT (With STDLABELS)
INPUT	FIXUNB or FIXBLK	96	*	112	*
	VARUNB or VARBLK	109	*	128	*
	UNDEF	92	*	108	*
OUTPUT	FIXUNB or FIXBLK	86	+10	104	+4
	VARUNB or VARBLK	98	+10	116	+4
	UNDEF	84	+4	100	+4

* Included in basic Size of Module.

MTMOD (MAGNETIC TAPE MODULE)

MAGNETIC TAPE MODULE					
		INDEPENDENT OPTIONS			
RECFORM=	BASIC MODULE	WORKA= YES	CKPTREC= YES	READ= BACK	ERREXT
FIXUNB/FIXBLK	846	+80	+136	+48	+64
VARUNB/VARBLK	890	+120	+144	+80	48
UNDEF	672	+80	+144	+16	+64

Notes:

1. Only one module is required for processing all files having a common RECFORM. This module can be generated with the options charted above. To determine the size of the module with the options, the number of option bytes specified in the chart must be added to the basic module.

2. RDONLY=YES, changes the size of the module -50 to +50 bytes. In addition, the user's program must provide a 72-byte save area each time the module is reentered.

WORKFILE MODULE			
		NOTEPNT=	
TYPEFIE=WORK	without NOTEPNT	YES	POINTS
without ERROPT	232	450	286
with ERROPT	322	540	380
with ERROPT & ERREXT	436	654	494

Figure 59. MTMOD Main Storage Requirements

DTFOR (DEFINE THE FILE: OPTICAL READER)

Table Requirements

1. RECFORM=FIXUNB requires 136 bytes.

2. RECFORM=FIXBLK varies as a function of blocking.
   Size = 136 + (16 x blocking factor x number of I/O areas)

3. RECFORM=UNDEF requires 136 bytes.

ORMOD (OPTICAL READER MODULE)

DEVICE=	RECFORM=	Basic Module	INDEPENDENT OPTIONS CONTROL= YES	IOAREA2= YES	WORKA= YES	IOAREA2= YES and WORKA=YES
1285 or 1287T	FIXUNB	892	+220	+76	+60	+96
	FIXBLK	1256	+264	+56	+28	+84
	UNDEF	848	+224	+56	+64	+88
	UNDEF BLKFAC=YES	1180	+268	+56	+24	+80
1287D	FIXUNB	1360	+184	---	---	---
	UNDEF	1256	+188	---	---	---

Figure 60. ORMOD Main Storage Requirements

DTFPH (DEFINE THE FILE: PHYSICAL IOCS)

Device	Size
Tape	104
DASD MOUNTED=ALL	40
DASD MOUNTED=SINGLE	84

Note: No module is required for this macro instruction.

Figure 61. DTFPH Main Storage Requirements

DTFPR (DEFINE THE FILE: PRINTER)

Table Requirements

Size = 48 bytes

PRMOD (PRINTER MODULE)

RECFORM=	BASIC MODULE	WORKA=YES	IOAREA2=YES	PRINTOV	CTLCHR= ASA	CTLCHR= YES	CONTROL=YES
FIXUNB	40	+44	+42	+78	+188	+34	+40
UNDEF	70	+22	+20	+76	+170	+16	+40
VARUNB	94	+14	+24	+76	+162	+8	+40

Note: RDONLY=YES changes the size of the module -50 to +50 bytes. In addition, the user's program must provide a 72-byte save area each time the module is reentered.

Figure 62. PRMOD Main Storage Requirements

DTFPT (DEFINE THE FILE: PAPER TAPE READER)

Table Requirements

The possible table specifications and sizes are:

1. No translations, no shifts, and no deletes require 72 bytes.

2. TRANS=name with no shifts and no deletes requires 76 bytes.

3. TRANS=name, SCAN=name, RECFORM=FIXUNB require 110 bytes.

4. TRANS=name, SCAN=name, RECFORM=UNDEF require 94 bytes.

PTMOD (PAPER TAPE READER MODULE)

The module specifications and sizes are:

1. No parameters specified (no translations, no shifts, and no deletes) requires 252 bytes.

2. TRANS=YES with no shifts and no deletes requires 310 bytes.

3. TRANS=YES, SCAN=YES, RECFORM=FIXUNB require 536 bytes.

4. TRANS=YES, SCAN=YES, RECFORM=UNDEF require 436 bytes.

   Note: If module 2 is used, all records require translation.

DTFSD (DEFINE THE FILE: SEQUENTIAL DASD)

RECFORM=	TYPEFLE					
	INPUT		OUTPUT	WORK		CONTROL = YES
	with UPDATE	without UPDATE		with UPDATE	without UPDATE	
FIXBLK or FIXUNB	176	152	160	152	152	+24
VARBLK or VARUNB	192	152	170	---	---	+24
UNDEF	192	152	162	152	152	+24

Figure 63. DTFSD Main Storage Requirements

SDMOD (SEQUENTIAL DASD MODULE)

Module Name	Basic Module	TRUNCS	CONTROL	ERROPT	HOLD	ERROPT ERREXT
SDMODFI	462	+80	+28	+144	*	+228
SDMODFO	546	+136	+28	+64	*	+200
SDMODFU	798	+88	+28	+164	+96	+252
SDMODVI	729		+28	+104	*	+188
SDMODVO	1045		+28	+68	*	+120
SDMODVU	1086		+28	+176	+76	+296
SDMODUI	533		+28	+100	*	+171
SDMODUO	653		+28	+68	*	+116
SDMODUU	941		+28	+148	+40	+248
SDMODW	572		+22	+148	+10	+246

*The HOLD function does not apply to these modules.

Notes:

1. For SDMODW, NOTEPNT=YES requires 206 additional bytes: NOTEPNT=POINTRW requires 144 additional bytes, UPDATE=YES requires 40 additional bytes.
2. RDONLY=YES changes the size of the module -50 to +50 bytes. In addition, the user's program must provide a 72-byte save area each time the module is reentered.

Figure 64. SDMOD Main Storage Requirements

DTFSR (DEFINE THE FILE: SERIAL DEVICE)

When a DTFSR is assembled, it generates both a table and a module. For example, if DTFSR is used for a printer, a table and a module are generated just as though DTFPR and PRMOD were used. To determine the main storage requirements for DTFSR, add the table and the module requirements for the appropriate device type--such as 48 bytes (DTFPR) and n bytes (PRMOD) for a printer.

DTFBT (DEFINE THE FILE: BTAM)

Table Requirements

Size = 64 + N (40 + 8x) + BUFCB + BUFNO (BUFL + f) + y + z
(+ 32N for BSC only)

where:

- N = number of lines in the line group
- x = number of CCWs in the largest channel program available for the device, given in Figure 65
- *BUFCB = 8 if a buffer pool is used
  = 0 if not
- *BUFNO = number of buffers in the pool
- *BUFL = length of each buffer
- f = number of bytes required to extend each buffer to a multiple of 8
- y = size of the model channel program table for the line group, given in Figure 65. If two or more DTFBTs use the same model channel program and are linkage edited together, include the value only once.
- z = size of the table of special characters, given in Figure 60. For BSC only, if two or more DTFBTs use the same transmission code, include the value only once.

*If the buffer pool is shared by 2 or more DTFBTs, include the value only once.

Device	x		y	z
	without start/stop	with Auto Poll		
1030	7	9	84	23
1050NS	7	9	68	16
1050S	11		136	34
1060	6	9	80	15
2260L	1		Not applicable	Not applicable
2260R	7		136	20
2740	4		24	21
2740C	4		68	33
2740D	6		52	18
2740DC	7		100	32
2740DT	8		68	22
2740DTC	8		116	36
2740S	6	9	48	12
2740SC	7	9	84	24
7770/72	4		52	0
115A	4		36	4
83B3	5		40	6
TWX 33/35	7		56	18
WTTA	5		52	36
1130 (leased)	8		148	53
1130 (dial)	8		196	53
1130 (multipoint)	8		164	53
S360 (leased)	8		240	53
S360 (dial) size depends on codes in the FEATURE operand:				
No ID Verif.	8		296	53
RIW	8		300	53
SIW,RXW	8		324	53
SXW,RXW	8		300	53
SIX,RXW	8		324	53
RXW	8		300	53
SIW,RIX	8		320	53
SXW,RIX	8		296	53
SIX,RIX	8		320	53
RIX	8		296	53
SIW	8		320	53
SXW	8		296	53
SIX	8		320	53
SIW,RIW	8		324	53
SXW,RIW	8		300	53
SIX,RIW	8		324	53
2780				
Pt to Pt Switched				
EBCDIC	8		140	53
USASCII	8		140	53
TRANSCODE	8		140	53
Pt to Pt Non Switched				
EBCDIC	8		96	53
USASCII	8		96	53
TRANSCODE	8		96	53
Multipoint				
EBCDIC	8		132	53
USASCII	8		84	53
TRANSCODE	8		84	53

Figure 65. Parameters for DTFBT Table Requirements Formula

DTFBTND (DEFINE THE FILE END: BTAM)

This macro instruction does not require main storage at execution time.

DFTRMLST (DEFINE THE TERMINAL LIST: BTAM)

Table Requirements for OPENLST and WRAPLST

Size = n (m + 1) + 2

where: n = number of list entries
m = number of polling/addressing characters--a function of the device, as shown in Figure 66

Device	m
1030	1
1050NS	2
1050S	2
1060	2
2260L	Not applicable
2260R	2
2740	Not applicable
2740C	Not applicable
2740D	Not applicable
2740DC	Not applicable
2740DT	Not applicable
2740DTC	Not applicable
2740S	1
2740SC	1
7770/72	Not applicable
115A	2
83B3	2
1130	2
2780	3
WTTA	Not applicable

Figure 66. Number of Polling/Addressing Characters

Table Requirements for DIALST

Size = n(m + 1) + 1 + q + p + i

where: n = number of list entries
m = values specified in Figure 66
q = 2 if the list includes entries for polling or addressing
= 0 otherwise
p = number of dial digits
i = 3 if the inlist operand is coded (BSC only)
= 0 otherwise

Table Requirements for IDLST

Size of TWX calling list = d + 5 + 2b
Size of TWX answering list = 5 + b
Size of BSC IDLST only = 3 + d + 2r + s + i

where: d = number of dial digits
b = number of TWX ID characters
r = number of ID characters expected to be received (BSC)
s = number of ID characters to be sent (BSC)
i = 3 if the inlist operand is coded (BSC)
= 0 otherwise

Table Requirements for SSLAST/SSAWLST

Size = n(m + 1) + 6

where: n = number of list entries
m = values specified in Figure 66

Table Requirements for AUTOLST/AUTOWLST

Size = n(m + 2) + 8

where: n = number of list entries
m = values specified in Figure 66

Table Requirements for WTTALST

1. When the WRU feature is present in DTFBT:

   Size = 3 + 2r + s

2. When the IAM feature is present, and WRU is not present in DTFBT:

   Size = 2 + s

where: r = number of ID characters expected to be received
s = number of ID characters to be sent

The size of the BTAM module varies with the options selected, as shown in Figure 67. When all operands are omitted or the standard (default) options are coded, the resulting basic module requires 4770 bytes of main storage.

Operand	Option	Number of bytes added to basic module
ERLOGIC=	N	-2090
	C	+ 320
	NC	-1830
SWITCH=	YES	+ 290
AUDIO=	YES	+ 410 (SWITCH must equal YES)
BUFFER=	YES	+1740
	REQREL	+ 640
TERMTST=	YES	+ 750
L2260=	YES	-1280 (If ERLOGIC=N)
		+1050 (If ERLOGIC=E)
		+1360 (If ERLOGIC=C)
		-1020 (If ERLOGIC=NC)
TRANSL=	YES	+ 140
BSCS=	YES	+9380 (If SWITCH=YES and BUFFER=YES)
		+6640 (If SWITCH=YES and BUFFER=NO)
		+5290 (If SWITCH=NO and BUFFER=NO)
		+8020 (If SWITCH=NO and BUFFER=YES)
SSAPL=	YES	- 970 (If ERLOGIC=N)
		+1560 (If ERLOGIC=E)
		- 700 (If ERLOGIC=NC)
		+1880 (If ERLOGIC=C)
BSCMPT=	YES	+ 880 (BSCS must equal YES)
WTTA	YES	+ 320
BSCTEST	YES	+1690
DECBEXT	YES	- 50 (BSCS must equal YES)

Figure 67. BTAM Main Storage Requirements

BTAM DATA EVENT CONTROL BLOCK

Table Requirements

Size = 40 bytes

When the MF operand of a READ or WRITE macro is coded MF=L, or when the MF operand is omitted, a Data Event Control Block (DECB) is reserved. One DECB should be reserved per line.

QTAM STORAGE REQUIREMENTS

The main storage requirements for QTAM depend to a great extent on the configuration of the user's Teleprocessing installation and the nature of his applications. Storage requirements increase proportionately as the number of communication lines, terminals, and QTAM-provided processing functions increase. These requirements can be estimated from formulas and tables presented in this section.

BASIC QTAM LOGIC MODULES

Figure 68 lists the basic QTAM logic modules and corresponding storage requirements.

Module	Module Name	Storage Requirements (in bytes)
Implementation module	IJLQIP	3749
LPS Control Module	IJLQLP	1010
Line Appendage for PCI and Program Check module (Note 4)	IJLQLA	728
Line Appendage and ERP Module (Note 4)	IJLQEP	4160
Audio Line Appendage module (Note 1)	IJLQAA	2376
Disk Appendage module (Note 4)	IJLQDA	1365
Audio Disk Appendage module (Note 2)	IJLQAD	1604
Physical I/O module (Note 4)	IJLQRW	1521
Terminal Test Recognition module (Note 4)	IJLQTT	1386
Checkpoint module (Note 3)	IJLQCK	1148
Message Writer Initiator module	IJLQMW	1255
IBM 2260 Local Appendage module (Note 5)	IJLQLO	1604
WTTA Appendage Module (Note 6)	IJLQTA	1370

Note 1: Required if the QTAM Audio Support is selected.

Note 2: Required if the IBM 7772 Vocabulary file is used.

Note 3: Required if the Checkpoint/Restart facility is selected in the DTFQT.

Note 4: Not required when the system contains only audio devices.

Note 5: Required if the IBM 2260 Local device is used.

Note 6: Required if World Trade Telegraph support is selected.

Figure 68. Basic QTAM Logic Modules

DTFQT (DEFINE THE FILE: QTAM)

There are seven types of DTF tables which may be generated by a DTFQT macro instruction. The storage estimates for each follow.

Table Requirements for DASD Message Queues File

Size = 315 bytes

Table Requirements for Communication Line Group File

For Nonaudio Line Group
  Size = 48 + (128 + 8x)N

For Audio Line Group
  Size = 64 + P + (161 + L_1 + L_2 + G + x-z)N

where:
- N = number of lines in the line group
- x = a function of the device, given in Figure 69
- P = 22 for IBM 7772 only
- L_1 = length of input buffers
- L_2 = length of address chain buffers
- G = 9 when using time stamping option, otherwise = 0
- z = 17 when information mode is used

Device	x
1030	8
1050NS	8
1050S	10
1060	7
2260 Remote	8
2260 Local	5
274A	4
274B	6
274C	6
274D	7

Device	x
274E	7
274F	4
274G	6
274H	7
115A	8
83B3	7
TWX33/35	6
7770	0
7772	60
WTTA	7

Figure 69. Values for Communication Line Group Table

Table Requirements for Main Storage Process Queue

Size = 84 + 12x

where: x = 0 for nonmixed application, and
1 for mixed application

Table Requirements for Main Storage Destination Queue

Size = 80 bytes

Table Requirements for Checkpoint Records File

Size = 220+L

where: L = length of the Checkpoint record specified by the SOWA keyword operand.

Table Requirements for IBM 7772 Vocabulary File

Size = 40+4p

where: p = number of BUFARU macro instructions

Table Requirement for Audio Output Queue

Size = 76 bytes

QTAM STORAGE REQUIREMENTS FOR CONTROL INFORMATION

The storage estimates for required control information are shown in Figure 70.

Control Blocks and Information	Storage Requirements (in bytes)
Terminal table TERMTBL macro instruction[1]	12
OPTION macro instruction	No storage is reserved for this macro; it defines user areas that are included in the expansion of the TERM macro instruction (U parameter)
TERM macro instruction[1]	9 + I + U + D + 44F where: (I + U + D)≤243
LIST macro instruction[1]	12 + L + 2N + 140[2] where: (3 + L +2N)≤243
PROCESS macro instruction[1]	13 + A for audio process program entry 9 + Y for nonaudio process program entry
Polling list POLL macro instruction[3]	4 + 3N for autopolled terminals except IBM 1030 4 + 2N for nonswitched terminals or autopolled IBM 1030 5 for switched IBM terminals 3 + I for TWX 4 + T for WTTA terminals
Queue Control Block for Process Queues and Destination Queues	32X
Audio Line Table LINETBL macro instruction[1]	4
LINE macro instruction[1]	5 + Z
Audio Word Table WORDTBL macro instruction[1]	4
WORD macro instruction[1]	8 + W

Figure 70. Storage Requirements for QTAM Control Information (Part 1 of 2)

where: N = number of terminals
I = number of bytes in terminal ID
U = number of bytes in optional area
D = number of bytes in device address area; size depends on contents:

For nonswitched terminal--addressing and polling characters (1 byte/character)
For IBM switched terminal--field telling the number of dial digits (1 byte) + dial digits (1 byte/digit) + addressing characters (1 byte/character)
For TWX--field telling the number of dial characters (1 byte) + dial digits (1 byte/digit) + field telling the number of ID characters (1 byte) + ID characters (2 bytes/character)
For WTTA terminals--1 byte + field telling the number of ID characters (1 byte) + ID characters (2 bytes/character)
For IBM 2260 Local--a field of 6 full words consisting of a CCB and other control information

L = number of bytes in name of the distribution list entry in terminal table (1-8)
A = number of bytes in name of the audio process entry in the terminal table. Because the following field in the entry must be aligned to a fullword boundary, this field must be either three, seven or eleven bytes long.
X = number of lines or terminals (depending on queuing techniques) and the number of process queues
Y = number of bytes in name of the process entry in terminal table (1-8)
Z = number of bytes in name of the line entry in line table (1-8)
W = number of bytes of the selected word
F = 1 for IBM 2740 Model 2 terminals with the Buffer Receiver Option; 0 for other terminals
T = number of bytes in the CPU identification (WTTA terminals)

Notes:

1. Add the number of bytes necessary for fullword boundary.

2. This number (140) is the number of bytes in the Distribution List module (IJLQDL). This number is included in the storage requirements only once if the LIST macro is used more than once.

3. Add the number of bytes necessary for halfword boundary.

Figure 70. Storage Requirements for QTAM Control Information (Part 2 of 2)

QTAM REQUIREMENT FOR BUFFERS

For Nonaudio Applications:

Buffer Pool Size = 8 + (X + 16)N + 24M

For Audio Applications with IBM 7772:

Audio Buffer Pool Size = 24 + (X + 88)N

where: N = number of buffers specified
X = size of each buffer
M = number of CCWs QTAM generates for data insertion by the PAUSE macro

QTAM STORAGE REQUIREMENTS FOR DEVICE I/O MODULES

Figure 71 gives the storage requirements for the QTAM device I/O modules. The storage for a particular device I/O module need be included only once and only if the terminal type is present in the system.

Terminal Device Type	Module Name	Storage Requirements (in bytes)
IBM 1030 Data Collection System	IJLQM0	194
IBM 1060 Data Communications System	IJLQM1	168
IBM 2260-2848 Display Complex	IJLQM2	206
AT&T 83B3 Selective Calling Stations	IJLQM3	102
Western Union Plan 115A Outstations	IJLQM4	90
IBM 1050 Data Communications System on a switched network	IJLQM5	207
IBM 1050 Data Communications System	IJLQM6	194
AT&T Model 33/35 TWX Stations	IJLQM8	113
IBM 2260-2848 Display Complex Local	IJLQM9	75
IBM 2740 Communications Terminal:		
Type 274A	IJLQN0	90
Type 275B	IJLQN1	157
Type 274C	IJLQN2	162
Type 274D	IJLQN3	221
Type 274E	IJLQN4	198
Type 274F	IJLQN5	119
Type 284G	IJLQN6	183
Type 275H	IJLQN7	156
World Trade Telegraph Terminals	IJLQN8	300

Figure 71. Storage Requirements for QTAM Device I/O Modules

REQUIREMENTS FOR OTHER QTAM MACRO INSTRUCTIONS

Figure 72 gives the storage requirements for all other QTAM macro instructions.

Much QTAM logic consists of modules introduced by the use of certain QTAM macro instructions in the user's program. These macro instructions expand into in-line coding that establishes the linkage to, and parameters for, the QTAM modules. Often a module so introduced into the system will itself introduce another module, a process termed a second level routine.

Column two of Figure 72 shows the extent of the coding produced by the expansion of the macro instruction in column one. If the coding links to a QTAM module, the module is presented in column three; if that module links to other modules, they are presented in column four.

Storage requirements for a sharable module, or for a second level routine that is linked to more than once in the same partition, are included only once. For example, the macro instructions DIRECT, EOA, and ROUTE all link to the same module, IJLQLK. If two or more of these macro instructions are used in the same message control program, the module is included only once. Similarly, if the same macro instruction is used more than once in the same program, storage is required for only the additional linkage since the module is included just once.

Macro Instruction	In-line linkage or code Note 5	Sharable modules		Second level routine	
		Size	Name	Size	Name
ARUMGTYP	16				
BREAKOFF	8	164	IJLQBO		
CANCELM	8	112	IJLQCM		
CHECKARU	70+ message text				
CHNGP	42	146	IJLQCP	78	IJLQFL
CHNGT	26	250	IJLQCT		
CKREQ	10	100	IJLQCR		
CLOSEMC	6	514	IJLQQT	512	IJLQCL
COPYC	24	414	IJLQDC		
COPYP	34	104	IJLQDP	78	IJLQFL
COPYQ	34	98	IJLQDQ		
COPYT	30	120	IJLQDE		
COUNTER	12				
DATESTMP	8	70	IJLQDT	70	IJLQEX
DIRECT	12	104	IJLQLK		
ENDRCV	10		Note 1		
ENDRCV (WTTA)	16	132	IJLQEB		
ENDREADY	80		Note 1		
ENDSEND	10		Note 1		
EOA	28	100	IJLQEA	104 76 48 60 104 Note 2	IJLQSH IJLQSK IJLQRG IJLQMT IJLQLK
EOB	6	132	IJLQEB		Note 1
EOBLC	6	380	IJLQEC		Note 1

Figure 72. Storage Requirements for Other QTAM Macro Instructions (Part 1 of 5)

Macro Instruction	In-line linkage or code Note 5	Sharable modules Size	Sharable modules Name	Second level routine Size	Second level routine Name
ERRMSG	32+ message text	284	IJLQER Note 1	104	IJLQLK
GET Segment	(See Figure 73.)	464	IJLQGS		
Message		482	IJLQGM		
Record		510	IJLQGR		
GET (AUDIO) Audio message	(See Figure 73.)	370	IJLQGA		
Audio and non-audio messages		767	IJLQGB		
Audio message & nonaudio record		792	IJLQGC		
Audio message & nonaudio segment		747	IJLQGD		
INTERCPT	12	128	IJLQIT		
LOGSEG	32	Note 3			
LOGSEG (ARU)	26	304	IJLQLG		
LPSTART	28		Note 1		
MODE (C)	14	48	MODE (U) module & IJLQMM	104	IJLQSH
MODE (U) INITIATE	10	24	IJLQMI		
PRIORITY	10	36	IJLQMP	104	IJLQSH
CONVERSE	10	336	IJLQMC Note 1		
MOD2260	4				
MSGTYPE (C)	19	60	IJLQMT	104	IJLQSH
MSGTYPE (U)	4				
OPCTL	56	3146	IJLQOC	104 104 512	IJLQLK IJLQSH IJLQCL Note 1

Figure 72. Storage Requirements for Other QTAM Macro Instructions (Part 2 of 5)

Macro Instruction	In-line linkage or code Note 5	Sharable modules Size	Sharable modules Name	Second level routine Size	Second level routine Name
PAUSE	13 + no. of insert chars.	352	IJLQPZ		Note 1
PCLIIMIT	12	120	IJLQPL		
POSTARU	6	Note 1			
POSTRCV	6		Note 1		
POSTSEND	12		Note 1		
PUT Segment	See Figure 73.	508	IJLQPS		
Message		468	IJLQPM		
Record		532	IJLQPR		
PUT (AUDIO)	See Figure 73.	386	IJLQPA		
RCVHDR	8				
RCVSEG Note 4	0				
RELEASEM	12	161	IJLQRM		
REPEAT	34	Note 1			
REROUTE	26	64	IJLQRR Note 1	104	IJLQLK
RETRIEVE DASD address	14	130	IJLQRD		
By sequence number	26	420	IJLQRS	130	IJLQRD
ROUTE	8	48	IJLQRG	104 104	IJLQLK IJLQSH
SENDHDR	16				
SENDSEG Note 4	4				
SEQIN	8	136	IJLQSI	104	IJLQSH

Figure 72. Storage Requirements for Other QTAM Macro Instructions (Part 3 of 5)

Macro Instruction	In-line linkage or code Note 5	Sharable modules		Second level routine	
		Size	Name	Size	Name
SEQOUT	8	68	IJLQSO	70	IJLQEX
SKIP (CT)	8	48	IJLQST	104	IJLQSH
SKIP (S)	8 + no. to be skipped	76	IJLQSK	104	IJLQSH
SOURCE	8	144	IJLQSR	104	IJLQSH
STARTARU	42	269	IJLQSS	78	IJLQFL
STARTLN	12	512	IJLQCL	78	IJLQFL
STOPARU	42	269	IJLQSS	78	IJLQFL
STOPLN	12	512	IJLQCL	78	IJLQFL
TIMESTMP	8	198	IJLQTS	70	IJLQEX
TRANS	10	114+ 266T	IJLQTR		
WRU	0				

Figure 72. Storage Requirements for other QTAM Macro Instructions (Part 4 of 5)

where:

C = character operand specified (conditional)
U = character operand null (unconditional)
S = skip to and include designated character configuration
CT = skip designated count of nonblank characters
T = number of translation tables

Translation tables are: RCV1030, RCV1050, RCV1050F, RCV1060, RCV2260, RCV2740, RCV2740F, RCVARU, RCV83B3, RCV115A, RCVTWX, RCVITA2, RCVZSC3, SND1030, SND1050, SND1060, SND2260, SNDITA2, SNDZSC3, SND2740, SND83B3, SND115A, SNDTWXE, and SNDTWXO

Notes:

1. These delimiters or modules cause linkages to QTAM routines included in Figure 66.

2. If the macro instruction MSGTYPE, ROUTE, or SKIP (S) is used in the program, the storage estimate for IJLQMT, IJLQRG, or IJLQSK, respectively, is not included in the storage estimate for EOA.

3. Because the user defines his own DTFxx and xxMOD macros for his message log file, the size requirements cannot be specified here. Information to determine the storage requirements for the specific logging medium is given under the pertinent DTF and Module, i.e., DTFMT and MTMOD for Tape. Refer to the section IOCS Declarative Macro Instructions.

4. Identifies entry point for RCVSEG and SENDSEG subgroups of LPS.

5. Figure 68 shows the linkage requirement for OPEN, CLOSE, GET, and PUT.

Figure 72. Storage Requirements for Other QTAM Macro Instructions (Part 5 of 5)

Other IOCS Macro Instructions

Figure 71 lists the main storage requirements for the expansion of the remainder of the IOCS macro instructions.

Macro Instruction Name	NUMBER OF BYTES: Basic Storage Requirement	Storage Required For Literals (1)	Variable Requirements
ASMTRTAB	256n		n=number of different operands coded
BTWAIT	16		
CCB	16-24		
CHECK	8	+4	(3)
CHGNTRY	20 (start-stop)		+6 (No operands coded in register notation)
	72 (BSC)		+6 (No operands coded in register notation)
	210 (start-stop Auto Poll)		+72 (No operands coded in register notation)
CHGNTRY	22 (2260 Local)		+4 (No operands coded in register notation)
CHNG (2)	0		
CLOSE	10	+8	+4 per filename parameter, or +8 per register parameter
CLOSER	14 plus 10 if any register parameter is specified	+8	+14 per filename parameter, or +8 per register parameter
CNTRL	10-18	+4	
CONTROL (BTAM)	20		+38 (No operands coded in register notation)
DISEN	8	+4	(3)
DSPLY	24	+4	(3)
ENDFL	8-12	+12	
ESETL	8-12	+4	
EXCP	2-6	0-+4	
FEOV	8-12	+4	
FREE	8-12	0-4	

Figure 73. Other IOCS Macro Instruction Storage Requirements (Part 1 of 4)

	NUMBER OF BYTES		
Macro Instruction Name	Basic Storage Requirement	Storage Required For Literals (1)	Variable Requirements
GET	8	+4 per symbolic name	(3)
LBRET	2		+2
LERB	20		
LITE	8	+4	(3)
LERPRT	14		+8 (No operands coded in register notation)
LOPEN	8		+2 (No operands coded in register notation)
NOTE	12	+4	
OPEN	10	+8	+4 per filename parameter, or +8 per register parameter
OPENR	14 plus 10 if any register parameter is specified	+8	+14 per filename parameter, or +8 per register parameter
POINTR	8	+4 per symbolic name	(3)
POINTS	8-12	+4	
POINTW	8	+4 per symbolic name	(3)
PRTOV	8	+4 per symbolic name	(3)
PUT	8-12	+4 per symbolic name	(3)
RDLNE	12	+4	(3)
READ	8-36	+4 per symbolic name	
READ (BTAM)	20		+38 (No operands coded in register notation)
RELBUF	14		+4 (First operand not coded in register notation)
RELSE	8	+4 per symbolic name	+4

Figure 73. Other IOCS Macro Instruction Storage Requirements (Part 2 of 4)

Macro Instruction Name	NUMBER OF BYTES: Basic Storage Requirement	Storage Required For Literals (1)	Variable Requirements
REQBUF	12		+8 (All possible operands not coded in register notation)
RESCN	24	+4	(3)
RESETPL	12		+4 (No operands coded in register notation)
SEOV	10	+8	+3
SETFL	8-12	+12	
SETL	14-22	+8	
TRNSLATE	26		+16 (No operands coded in register notation)
TRSRCTW (BTAM WTTA)	256		
TRSRCT3 (BTAM WTTA)	256		
TRSSCTW (BTAM WTTA)	256		
TRSSCT3 (BTAM WTTA)	256		
TRUNC	8-12	0-+4	
TWAIT	24		+20 (All possible operands not coded in register notation)

Figure 73. Other IOCS Macro Instruction Storage Requirements (Part 3 of 4)

Macro Instruction Name	NUMBER OF BYTES: Basic Storage Requirement	NUMBER OF BYTES: Storage Required for Literals (1)	NUMBER OF BYTES: Variable Requirements
WAIT	10-+4	0-+4	
WAIT (BTAM)	16		+20 (No operands coded in register notation)
WAITF	8-12	0-+4	+4 per filename parameter or +8 per register parameter
WRITE	8-26	0-+4	
WRITE (BTAM)	20		+38 (No operands coded in register notation)

Notes:

1. The storage requirement for literals shared by more than one macro instruction should be included for only one macro instruction. For example, EXCP and WAIT share a common CCB name. Other macro instructions can share a common filename.

2. CHNG is provided for Basic Programming Support compatibility.

3. +4 per symbolic name that is specified. +2 per register that is specified in ordinary register notation. +0 per register that is specified in special notation.

Figure 73. Other IOCS Macro Instruction Storage Requirements (Part 4 of 4)

Relocatable Subroutines

This section lists the amounts of main storage required for FORTRAN, Basic FORTRAN, PL/I, and COBOL subroutines when they have been linkage edited into an executable object program. For many FORTRAN, Basic FORTRAN, COBOL, and PL/I main programs or subroutines, several modules are required. Basic FORTRAN programs, for example, require the modules IJTACOM, IJTACON, and IJTFIOS.

The number of blocks (physical records) per routine in the relocatable library is also shown in this section. On a 2311, each track contains nine 322-byte blocks; on a 2314, each track contains sixteen 322-byte blocks. This information is useful in estimating linkage editor times. See Linkage Editor Times discussion in IBM System/360 Disk Operating System, Timing Estimates listed on the front cover of this manual.

COBOL SUBROUTINES

For additional information about these subroutines, see the COBOL Programmer's Guide listed in the Preface of this publication.

Module Name and Alternate Entry Point(s)	Bytes of Main Storage	Number Relocatable Library Blocks (Physical Records)
IHD00000	426	6
IHD00001		
IHD00100	428	3
IHD00101		
IHD00200	142	2
IHD00201		
IHD00300	170	2
IHD00301		
IHD00400	804	6
IHD00401		
IHD00500	436	7
IHD00501		
IHD00600	348	3
IHD00601		
IHD00602		
IHD00700	280	6
IHD00701		
IHD00800	68	4
IHD00801		
IHD00802		
IHD00900	56	4
IHD00901		
IHD01000	53	2
IHD01001		
IHD01100	216	3
IHD01101		
IHD01102		

Module Name and Alternate Entry Point(s)	Bytes of Main Storage	Number Relocatable Library Blocks (Physical Records)
IHD01200	36	4
IHD01201		
IHD01300	56	4
IHD01301		
IHD01400	52	4
IHD01401		
IHD01500	330	3
IHD01501		
IHD01502		
IHD01504		
IHD01600	84	2
IHD01601		
IHD01602		
IHD01700	217	3
IHD01701		
IHD01800	57	2
IHD01801		
IHD01802		
IHD01900	812	8
IHD01901		
IHD01902		
IHD01904		
IHD01908		
IHD02000	358	3
IHD02001		
IHD02002		
IHD02100	178	2
IHD02101		
IHD02200	20	2
IHD02201		
IHD02300	188	2
IHD02301		
IHD02302		
IHD02304		
IHD02400	182	2
IHD02401		
IHD02500	196	2
IHD02501		
IHD02600	64	2
IHD02601		
IHD02700	420	4
IHD02701		
IHD02702		

IHD02800	372	4
IHD02801		
IHD02802		
IHD02900	316	4
IHD02901		
IHD03000	580	5
IHD03001		
IHD03002		
IHD03100	880	5
IHD03102		
IHD03104		
IHD03108		
IHD03200	607	4
IHD03201		
IHD03300	228	3
IHD03301		
IHD03400	630	4
IHD03401		
IHD03500	796	5
IHD03501		
IHD03502		
IHD03600	338	3
IHD03700	691	5
IHD03701		
IHD03702		
IHD03800	1162	7
IHD03801		
IHD03802		
IHD03803		
IHD03804		
IHD03900	781	4
IHD03901		
IHD04000	622	4
IHD04001		
IHD04100	1888	7
IHD04101		

FORTRAN SUBROUTINES

For additional information about these subroutines, see the FORTRAN IV Programmer's Guide listed in the Preface of this manual.

Module Name	Bytes of Main Storage	Number Relocatable Library Blocks (Physical Records)
ILFACOM	1480	8
ILFADCON	4312	20
ILFCLABS	144	3
ILFCLAS	208	3
ILFCLEXP	208	4
ILFCLLOG	224	4
ILFCLSCN	408	5
ILFCLSQT	168	3
ILFCSABS	136	3
ILFCSAS	208	3
ILFCSEXP	208	4
ILFCSLOG	216	4
ILFCSSCN	344	4
ILFCSSQT	168	3
ILFDEBUG	1872	10
ILFDIOCS	584	4
ILFFCDXI	288	3
ILFFCXPI	272	3
ILFFDUMP	472	4
ILFFDVCH	72	3
ILFFDXPD	176	3
ILFFDXPI	168	3
ILFFEXIT	32	2

ILFFINT	1488	9
ILFFIOCS	3032	16
ILFFIXPI	184	3
ILFFMAXD	120	2
ILFFMAXI	224	3
ILFFMAXR	216	3
ILFFOVER	80	2
ILFFRXPI	168	3
ILFFRXPR	168	3
ILFFSLIT	192	3
ILFGHTAB	256	3
ILFIBCOM	3928	19
ILFIBERR	208	3
ILFLASCN	400	4
ILFLATN2	528	5
ILFLERF	800	6
ILFLEXP	480	4
ILFLGAMA	728	5
ILFLLOG	392	4
ILFLSCN	416	4
ILFLSCNH	304	3
ILFLSQRT	160	3
ILFLTANH	312	3
ILFLTNCT	416	3
ILFNAMEL	2224	11
ILFSASCN	312	3
ILFSATN2	384	4
ILFSERF	432	3
ILFSEXP	304	4
ILFSGAMA	488	5
ILFSLOG	288	3
ILFSSCN	304	3
ILFSSCNH	248	3
ILFSSQRT	184	3
ILFSTANH	234	3
ILFSTNCT	320	4
ILFTRBK	640	6
ILFUNTAB	256	3

BASIC FORTRAN SUBROUTINES

For additional information about these subroutines, see the Basic FORTRAN IV Programmer's Guide listed in the Preface of this publication.

Module Name	Bytes of Main Storage	Number Relocatable Library Blocks (Physical Records)
IJTAAFR	160	2
IJTACOM	2416	13
IJTACON	2448	12
IJTADIR	1112	8
IJTADXD	200	3

IJTADXI	144	3
IJTAIXI	160	3
IJTAPST	144	3
IJTARBE	928	6
IJTARXI	136	3
IJTARXR	208	3
IJTDVCK	64	2
IJTEXPN	280	3
IJTFDMP	648	6
IJTFIOS	3568	17
IJTFXIT	24	3
IJTHEXC	144	2
IJTIFIX	112	2
IJTLEXP	480	4
IJTLLOG	368	3
IJTLSCN	392	4
IJTLSQT	168	3
IJTLTAN	312	3
IJTLTNH	336	3
IJTMAXD	104	2
IJTMODI	56	2
IJTMODR	104	2
IJTOVRF	80	3
IJTSINT	72	2
IJTSLIT	184	3
IJTSLOG	264	3
IJTSMX0	192	3
IJTSMX1	192	3
IJTSSCN	280	3
IJTSSQT	192	3
IJTSTAN	192	2
IJTSTNH	264	3

PL/I SUBROUTINES

For additional information about these subroutines see the PL/I Programmer's Guide listed in the Preface of this publication.

Module Name	Bytes of Main Storage	Number Relocatable Library Blocks (Physical Records)
IJKEXHC	1200	10
IJKQALM	456	4
IJKQASM	232	4
IJKQBLA	280	3
IJKQBSA	208	3
IJKQCLA	288	3
IJKQCSA	208	3
IJKQDLA	288	3
IJKQDSA	212	3
IJKQLLA	384	4
IJKQLSA	272	3

IJKQNLD	608	4
IJKQNSD	480	4
IJKQQLM	160	4
IJKQQSM	176	3
IJKQRLB	776	5
IJKQRSB	408	3
IJKQSLD	416	4
IJKQSSD	304	3
IJKQTLB	360	3
IJKQTSB	280	3
IJKRBBM	424	4
IJKRBIM	292	4
IJKRBKA	292	3
IJKREBM	92	3
IJKRELM	152	3
IJKREPM	140	3
IJKRESM	144	3
IJKRGIM	108	2
IJKRGKM	84	3
IJKRMBX	278	3
IJKRMLX	172	3
IJKRMPX	386	4
IJKRMSX	132	3
IJKRSBM	200	3
IJKRSLM	192	3
IJKRSPM	265	3
IJKRSSM	184	3
IJKRUBM	148	3
IJKRWBM	356	4
IJKRWLM	244	3
IJKRWPM	577	5
IJKRWSM	236	3
IJKRXLM	168	3
IJKRXSA	152	3
IJKSDMP	56	2
IJKSDTM	58	2
IJKSTMM	104	2
IJKSYSA	201	3
IJKSYSI	152	3
IJKSZBA	1772	8
IJKSZCA	1636	8
IJKSZLM	60	3
IJKTCBM	550	4
IJKTCUM	252	3
IJKTDPD	184	3
IJKTDIM	540	4
IJKTFDM	480	4
IJKTFMM	196	3
IJKTGDI	414	4
IJKTLCM	876	5
IJKTLIM	1158	7
IJKTLOM	1076	7
IJKTLTB	7	2
IJKTPSM	72	3

IJKTRGM	398	4
IJKTRON	1372	8
IJKTSIM	652	5
IJKTSTM	674	5
IJKTXCF	120	2
IJKTXRM	356	3
IJKTXRN	378	3
IJKVBCM	60	3
IJKVBTM	132	3
IJKVCBM	238	3
IJKVCEM	984	6
IJKVCFM	680	5
IJKVCPM	214	3
IJKVCTM	392	4
IJKVECM	404	4
IJKVFCM	492	4
IJKVGIM	236	3
IJKVIGM	148	3
IJKVIIM	180	3
IJKVNPM	324	4
IJKVPCM	68	3
IJKVPNM	316	4
IJKVPRM	1196	7
IJKVRPM	796	5
IJKVTBM	228	4
IJKVTCM	284	4
IJKXTBM	128	2
IJXSYSA	305	3
IJXSYSI	224	3

IOCS Modules for the Compilers

These modules are described in Appendix B of this publication.

COBOL Module	Module Name	Bytes of Main Storage	Number Relocatable Library Blocks (Physical Records)
CDMOD (Card Modules)	IJCFZIZ0	96	2
	IJCFZIZ1	100	2
	IJCFZIZ2	96	2
	IJCFZIZ3	96	2
	IJCFAOZ0	92	2
	IJCFAOZ1	128	2
	IJCFAOZ2	132	2
	IJCFAOZ4	248	3
	IJCFAOZ5	224	3
	IJCFZOZ1	74	2
	IJCFZOZ2	80	2
	IJCFZOZ4	208	3
	IJCFZOZ5	192	3
	IJCFZII0	128	2
	IJCFZII1	132	2
	IJCFZII2	128	2
	IJCFZII3	128	2

	IJCFAOI0	116	2
	IJCFAOI1	152	2
	IJCFAOI2	156	2
	IJCFAOI4	264	3
	IJCFAOI5	248	3
	IJCFZOI1	116	2
	IJCFZOI2	124	2
	IJCFZOI4	232	3
	IJCFZOI5	216	3

Note:

The last character in each module name identifies the card device:

0 = 2540 (input only)
1 = 1442
2 = 2520
3 = 2501
4 = 2540 when CRDERR=RETRY is specified.

PRMOD (Printer Module)	IJDFAPIZ	280	3
	IJDFAPZZ	268	2
	IJDFZPIZ	152	2
	IJDFZPZZ	118	2
MTMOD (Magnetic Tape Modules)	IJFFBCZZ	894	6
	IJFUBCZZ	848	5
	IJFVBCWZ	1274	6
	IJFVBCZZ	1138	6
SDMOD (Sequential DASD Modules)	IJGFIEZZ	614	4
	IJGFIZZZ	470	3
	IJGFOZZZ	554	3
	IJGFUZZZ	806	4
	IJGUIEZZ	641	4
	IJGUIZZZ	541	3
	IJGUOZZZ	671	4
	IJGUUZZZ	953	5
	IJGVIEZZ	741	4
	IJGVIZZZ	637	4
	IJGVOZZZ	1077	5
	IJGVUZZZ	1094	6
DAMOD (Direct Access Module)	IJIBAIZZ	1064	5
	IJIBAZZZ	864	4
	IJIBZIZZ	848	4
	IJIBZZZZ	648	4
	IJIFAIZZ	988	5
	IJIFAZZZ	794	4
	IJIFZIZZ	648	4
	IJIFZZZZ	492	3
ISMOD (Indexed Sequential DASD Module)	IJHAABZZ	4524	19
	IJHAARCP	4334	19
	IJHAARCZ	3966	18
	IJHAARZP	4062	17
	IJHAARZZ	3684	16
	IJHAASZZ	3888	16
	IJHAIZZZ	3040	13
	IJHBABZZ	4280	18
	IJHBARCP	4090	18
	IJHBARCZ	3718	16

IJHBARZP	3820	17
IJHBARZZ	3440	15
IJHBASZZ	3642	15
IJHBIZZZ	2794	12
IJHUABZZ	4064	17
IJHUARCP	3822	17
IJHUARCZ	3506	15
IJHUARZP	3540	15
IJHUARZZ	3224	14
IJHUASZZ	3428	15
IJHUIZZZ	2584	11
IJHZLZZZ	811	4
IJHZRBZZ	2124	10
IJHZRRCZ	1460	8
IJHZRRZZ	1290	7
IJHZRSZZ	1268	7

PL/I Module	Module Name	Bytes of Main Storage	Number Relocatable Library Blocks (Physical Records)
CDMOD (Card Modules)	IJCFAOI1	152	2
	IJCFAOI2	156	2
	IJCFAOI4	264	3
	IJCFAOZ1	128	2
	IJCFAOZ2	132	2
	IJCFAOZ4	248	3
	IJCFYOI1	132	2
	IJCFYOI2	136	2
	IJCFYOI4	248	3
	IJCFYOZ1	108	2
	IJCFYOZ2	112	2
	IJCFYOZ4	224	3
	IJCFZIZ0	96	2
	IJCFZIZ1	100	2
	IJCFZIZ2	96	2
	IJCFZIZ3	96	2
	IJCFZII0	128	2
	IJCFZII1	132	2
	IJCFZII2	128	2
	IJCFZII3	128	2
	IJCFZOZ1	74	2
	IJCFZOZ2	80	2
	IJCFZOZ4	208	3
	IJCFZOI1	116	2
	IJCFZOI2	124	2
	IJCFZOI4	232	3

Note: The last character in each module name identifies the card device:

0 = 2540 (input only)
1 = 1442
2 = 2520
3 = 2501
4 = 2540 when CRDERR=RETRY is specified.

PRMOD	IJDFAZIZ	220	2
(Printer	IJDFAZZZ	196	2
Modules)	IJDFYZIZ	96	2
	IJDFYZZZ	72	2
	IJDFZPIZ	152	2
	IJDFZPZZ	118	2
MTMOD	IJFFBZZZ	894	5
(Magnetic	IJFFZZZZ	846	5
Tape Modules)	IJFUBZZZ	680	4
	IJFUZZZZ	672	4
	IJFVZZZZ	890	5
SDMOD	IJGFIETZ	742	4
(Sequential	IJGFIEZZ	614	4
DASD Modules)	IJGFOEZZ	618	4
	IJGFUEZZ	970	5
	IJGUIEZZ	641	4
	IJGUOEZZ	729	4
	IJGUUEZZ	1101	6
	IJGVIEZZ	741	4
	IJGVOEZZ	1145	6
	IJGVUEZZ	1270	6
DAMOD	IJIFAZZZ	794	4
(Direct	IJIFZZZZ	492	3
Access			
Modules)			
ISMOD	IJHAARCP	4334	19
(Indexed	IJHAARCZ	3966	18
Sequential	IJHAARZP	4062	17
DASD Modules)	IJHAARZZ	3684	16
	IJHBARCP	4090	18
	IJHBARCZ	3718	16
	IJHBARZP	3812	17
	IJHBARZZ	3440	15
	IJHUARCP	3822	17
	IJHUARCZ	3506	15
	IJHUARZP	3540	15
	IJHUARZZ	3224	14
	IJHZLZZZ	811	4
	IJHZRRCZ	1460	8
	IJHZRRZZ	1290	7
	IJHZRSZZ	1268	7
DIMOD	IJJFCBID	747	4
(Device	IJJFCBZD	668	4
Independent	IJJFCIID	368	3
Modules)	IJJFCIZD	308	3

RPG Modules	Module Name	Bytes of Main Storage	Number Relocatable Library Blocks (Physical Records)
CDMOD	IJCFCCZ0	218	2
(Card	IJCFCCZ1	132	2
Modules)	IJCFCCZ2	218	2

	IJCFCIZ0	158	2
	IJCFCIZ1	124	2
	IJCFCIZ2	142	2
	IJCFYOI0	96	2
	IJCFYOI1	132	2
	IJCFYOI2	136	2
	IJCFYOZ0	72	2
	IJCFYOZ1	108	2
	IJCFYOZ2	112	2
	IJCFZII0	128	2
	IJCFZII1	132	2
	IJCFZII2	128	2
	IJCFZII3	128	2
	IJCFZIZ3	96	2

Note: The last character in the module name indicates the card device:

0 = 2540
1 = 1442
2 = 2520
3 = 2501

PRMOD (Printer Module)	IJDFYPZZ	138	2
MTMOD (Magnetic Tape Modules)	IJFFZCZZ	1046	6
	IJFVZCWZ	1200	6
DAMOD (Direct Access Module)	IJIBZZZZ	648	4
ISMOD (Indexed Sequential Modules)	IJHZRBZZ	2124	10
	IJHZRRZZ	1290	7
	IJHZRSZZ	1268	7
SDMOD (Sequential DASD Modules)	IJGFIZZZ	470	3
	IJGFUZZZ	806	4
	IJGFOZZZ	554	4
	IJGVIZZZ	637	4
	IJGVUZZZ	1094	6
	IJGVOZZZ	1077	5

Other Modules Required for the System	Module Name	Bytes of Main Storage	Number Relocatable Library Blocks (Physical Records)
Tape Work File Modules	IJFWEZZZ	322	3
	IJFWZNZZ	450	3
	IJFWZZZZ	232	2
Disk Work File Modules	IJGWEZZZ	728	4
	IJGWZNZZ	800	4
	IJGWZRZZ	738	4

Nondevice Dependent IOCS Modules Used by IBM Processor Programs	IJJCP0	363	3
	IJJCP1	285	3
	IJJCP2	108	2
	IJJCP3	38	2
	IJJCPD0	521	3
	IJJCPD1	451	3
	IJJCPD2	252	2
	IJJCPD3	176	2
	IJJCPDV	483	3
	IJJCPDV2	184	2
	IJJCPV	317	3
	IJJCPV2	48	2
	IJJCPV1	245	2
	IJJCP1N	213	2
	IJJCP0N	291	3
	IJJCPDV1	427	3
	IJJCPD1N	395	3
	IJJCPD0N	465	3

BTAM Modules	Module Name	Bytes of Main Storage	Number Relocatable Library Blocks (Physical Records)
	IJL0AY	78	2
BTAM Channel Program Modules (For start-stop, the modules include a Table of Special Characters)	IJL0EZ	88	2
	IJL00Y	52	2
	IJL01Z	107	2
	IJL01J	155	2
	IJL02Z	95	2
	IJL02J	143	2
	IJL03Z	156	2
	IJL04Z	46	2
	IJL05Z	40	2
	IJL07Z	84	2
	IJL07J	100	2
	IJL07Y	170	2
	IJL08Z	45	2
	IJL08X	60	2
	IJL08H	65	2
	IJL08P	108	2
	IJL089	113	2
	IJL08M	152	2
	IJL08R	101	2
	IJL08Q	132	2
	IJL08U	90	2
	IJL08Y	70	2
	IJL09Y	74	2
	IJL10Y	52	2
	IJL@06Z	16	1
BTAM WTTA Subroutine	IJLWTZ	1031	6

BTAM Modules	Module Name	Bytes of Main Storage	Number Relocatable Library Blocks (Physical Records)
BTAM Channel Program Modules for BSC	IJL0BZ	240	3
	IJL0BY	296	3
	IJL0BA	300	3
	IJL0BC	324	3
	IJL0BE	300	3
	IJL0BG	324	3
	IJL0BI	300	3
	IJL0BK	320	3
	IJL0BM	296	3
	IJL0BO	320	3
	IJL0BQ	296	3
	IJL0BS	320	3
	IJL0BU	296	3
	IJL0BW	320	3
	IJL0B4	324	3
	IJL0B6	300	3
	IJL0B8	324	3
	IJL0CY	196	2
	IJL0CZ	148	2
	IJL0DY	140	2
	IJL0DZ	96	2
	IJL1BZ	220	2
	IJL1CZ	164	2
	IJL1DZ	132	2
	IJL2DZ	84	2
BSC Tables of Special Characters	IJLEBD	53	2
	IJLASC	53	2
	IJLTCD	53	2

	Module Name	Bytes of Main Storage	Number Relocatable Library Blocks (Physical Records)
BTAM	IJLRASA	256	2
Translation	IJLRCTW	256	2
Tables	IJLRCT1	256	2
	IJLRCT2	256	2
	IJLRCT3	256	2
	IJLRC30	256	2
	IJLRC40	256	2
	IJLRC50	256	2
	IJLRC60	256	2
	IJLRC80	256	2
	IJLRF40	256	2
	IJLRF50	256	2
	IJLRSCI	256	2
	IJLSASA	256	2
	IJLSCTW	256	2
	IJLSCT1	256	2
	IJLSCT2	256	2
	IJLSCT3	256	2
	IJLSD30	256	2
	IJLSD40	256	2
	IJLSD50	256	2
	IJLSD60	256	2
	IJLDS80	256	2
	IJLSSCI	256	2

External Storage Requirements

This section discusses the residence requirement for each IBM-supplied component in a system pack. The overall organization of the system pack is as follows:

Name	Start Location, if Present
IPL Program	Track 0 of Cylinder 0.
System Volume Label	Track 0 of Cylinder 0.
System Directory	Track 1 of Cylinder 0.
Librarian Work Area	Tracks 2, 3, and 4 of Cylinder 0.
Transient Directory	Track 5 of Cylinder 0.
Open Routine Directory	Track 6 of Cylinder 0.
Library Routine Directory	Track 7 of Cylinder 0.
Foreground Program Directory	Track 8 of Cylinder 0.
Problem Program Phase Directory	Track 9 of Cylinder 0.
Core Image Directory	Track 0 of Cylinder 1, on a 2311; track 10 of Cylinder 0 on a 2314.

Core Image Library	Beginning of the first available track following the core image directory.
Relocatable Directory, Optional	Track 0 of the first available cylinder following the core image library.
Relocatable Library, Optional	Beginning of the first available track following the relocatable directory.
Source Statement Directory, Optional	Track 0 of the first available cylinder following the previous library.
Source Statement Library Optional	Beginning of the first available track following the source statement directory.
Label Cylinder	First full cylinder after the last system library.

Track	Provides Storage For
0	Background User Labels
1	Background PARSTD Labels
2	Foreground 2 User Labels
3	Foreground 2 PARSTD Labels
4	Foreground 1 User Labels
5	Foreground 1 PARSTD Labels
6 - 9	Standard Labels (2311)
or	
6 - 19	Standard Labels (2314)

Volume Table of Contents (VTOC)	Location assigned by the user.

Directory Entries: Each track in the core image directory can contain entries for 144 phases on the 2311 and 270 phases on the 2314. Each track in the relocatable directory can contain entries for 180 modules on the 2311 and 340 modules on the 2314. Each track in the source statement directory can contain entries for 160 books on the 2311, and 270 books on the 2314. However, the first track in both the relocatable and the source statement directories can contain five fewer entries.

Library Block Lengths: In the core image library on a 2311, each track contains two 1728-byte blocks; on a 2314, each track contains four 1688-byte blocks. In the relocatable library on a 2311, each track contains nine 322-byte blocks; on a 2314, each track contains sixteen 322-byte blocks. In the source statement library on a 2311, each track contains sixteen 160-byte blocks; on a 2314, each track contains twenty-seven 160-byte blocks.

Core Image Library

Component	Number of Library Blocks (Physical Records)	
	2311	2314
Supervisor Nucleus (6K)	4	-
Supervisor Nucleus (8K)	5	5
All Other Transient Functions[1]	202	200
Job Control	20	20
Linkage Editor	14	15
IPL	4	4
CSERV	5	5
DSERV	5	5
RSERV	4	4
SSERV	4	4
MAINT	32	32
CORGZ	17	17
Assembler	87	87
Assembler F	69	69
Autotest	58	-
COBOL	171	173
COBOL Debug	4	4
FORTRAN	42	42
Basic FORTRAN	20	20
PL/I	264	268
RPG	98	101

Tape Sort	36	36
Disk Sort	74	76
Tape and Disk Sort	69	70
CE Serviceability Programs	14	14
Utilities:		
Card-to-Disk	10	10
Card-to-Printer and/or Punch	12	12
Card-to-Tape	10	10
Clear Data Cell	4	4
Clear Disk	4	4
Data Cell to Data Cell	11	11
Data Cell to Disk	11	11
Data Cell to Printer	12	12
Data Cell to Tape	10	10
Disk-to-Card	11	11
Disk to Data Cell	11	11
Disk-to-Disk	11	11
Disk-to-Printer	12	12
Disk-to-Tape	10	10
Tape-Compare	5	5
Tape-to-Card	11	11
Tape to Data Cell	11	11
Tape-to-Disk	11	11
Tape-to-Printer	11	12
Tape-to-Tape	10	10
VTOC Display	3	3
Copy Disk-to-Disk	7	7
Copy Disk or Data Cell-to-Tape	7	7
Copy Disk-to-Card	7	7
Restore Tape-to-Disk or Data Cell	4	4
Restore Card-to-Disk	4	4
Initialize Disk	14	14
Alternate Track Assignment Disk	25	26
Initialize Data Cell	9	10
Alternate Track Assignment Data Cell	20	20
OLTEP[2]	16	16
Vocabulary File Utility	19	19
BTAM Transient Routines	20	20
QTAM Transient Routines	22	22

Notes:

1. All transient functions not specifically listed are grouped together.
2. If the OLTEP Tape-Gap timing program is run in a multiprogramming environment, the performance of any foreground program may be affected.

Relocatable Library

Component	Number of Library Blocks (Physical Records)
System Control	790
Job Control	
Linkage Editor	
CSERV	
DSERV	
RSERV	
SSERV	
IPL	
CORGZ	
MAINT	
Standard System Dump	
Translating System Dump	
Assembler	604
Assembler F	475
Autotest	358
COBOL	1342
COBOL Subroutines	161
COBOL Debug	23
FORTRAN	300
FORTRAN Subroutines	279
Basic FORTRAN	140
Basic FORTRAN Subroutines	148
PL/I	2252
PL/I Subroutines	342
RPG	782
Tape Sort	204
Tape and Disk Sort	974
Disk Sort	518
CE Serviceability Programs	51
Utilities (3 Groups)	
Group 1	666
Group 2	432
Group 3	230
OLTEP[1]	144
Vocabulary File Utility	139
BTAM Modules	179
QTAM Modules	394
Compiler IOCS Modules	
COBOL	546
PL/I	384
RPG	102
Non-Device Dependent	46

[1] If the OLTEP Tape-Gap timing test program is run in a multiprogramming environment, the performance of any foreground program may be affected.

Source Statement Library

File Definition Macros	Number of Library Blocks
A.CDMOD	197
A.DAMOD	188
A.DIMOD	140
A.DTFBG	2
A.DTFCD	108
A.DTFCN	73
A.DTFDA	198
A.DTFDI	61
A.DTFEN	24
A.DTFIS	7
A.DTFMR	72
A.DTFMT	165
A.DTFOR	91
A.DTFPH	51
A.DTFPR	56
A.DTFPT	111
A.DTFSD	197
A.DTFSR	62
A.ISMOD	4
A.MRMOD	102
A.MTMOD	620
A.ORMOD	421
A.PRMOD	95
A.PTMOD	206
A.SDMOD	120
A.SDMODFI	193
A.SDMODFO	196
A.SDMODFU	301
A.SDMODVI	115
A.SDMODVO	144
A.SDMODVU	197
A.SDMODUI	103
A.SDMODUO	114
A.SDMODUU	174
A.SDMODW	275
A.DTFIS1	62
A.DTFIS2	66
A.DTFIS3	53
A.ISMOD0	94
A.ISMOD1	193
A.ISMOD2	155
A.ISMOD3	116
A.ISMOD4	113
A.ISMOD5	114
A.ISMOD6	195
A.ISMOD7	130
A.ISMOD8	111
A.ISMOD9	165
Total	6,750

Supervisor Communications Macros	Number of Library Blocks
A.ATTACH	11
A.CALL	10
A.CANCEL	4
A.CHKPT	23
A.CHNG	2
A.COMRG	2
A.DEQ	5
A.DETACH	4
A.DUMP	2
A.ENQ	4
A.EOJ	2
A.EXIT	5
A.FETCH	6
A.GETIME	9
A.LOAD	6
A.MVCOM	7
A.PDUMP	7
A.POST	6
A.RCB	2
A.RETURN	11
A.SAVE	10
A.SETIME	8
A.STXIT	9
A.TECB	2
A.WAITM	8
Total	165

BTAM Macros	Number of Library Blocks
A.ASMTRTAB	3
A.BTMOD	2313
A.DFTRMLST	82
A.DTFBT	186
A.DTFBTND	2
A.IJLBTMDS	60
A.TRNSLATE	48
A.CONTROL	3
A.CHGNTRY	107
A.BTWAIT	37
A.LERB	30
A.BTRWC	149
A.LERPRT	35
A.LOPEN	9
A.RELBUF	19
A.REQBUF	29
A.RESETPL	11
A.TWAIT	27
A.ONLTST	50
A.TRSRCTW	20
A.TRSRCT3	20
A.TRSSCTW	24
A.TRSSCT3	24
Total	3,387

QTAM Macros	Number of Library Blocks
A.IJLQBABD	27
A.IJLQBFRD	29
A.IJLQBRBD	11
A.IJLQCKPD	8
A.IJLQDTFD	152
A.IJLQIP1D	4
A.IJLQLABD	53
A.IJLQLCBD	67
A.IJLQMCBD	12
A.IJLQQCBD	26
A.IJLQSTBD	4
A.IJLQSVCD	13
A.IJLQTBLD	9
A.IJLQVECD	12
IJLQCTLD	9
A.ARUMGTYP	11
A.BREAKOFF	4
A.BUFARU	18
A.BUFFER	17
A.CANCELM	5
A.CHECKARU	26
A.CHNGP	12
A.CHNGT	7
A.CKREQ	4
A.CLOSEMC	4
A.COPYC	10
A.COPYP	12
A.COPYQ	7
A.COPYT	7
A.COUNTER	5
A.CTLTBL	7
A.DATESTMP	4
A.DIRECT	5
A.DTFQT	403
A.ENDRCV	7
A.ENDREADY	10
A.ENDSEND	7
A.EOA	6
A.EOB	2
A.EOBLC	2
A.ERRMSG	15
A.INTERCPT	7
A.LINE	8
A.LINETBL	7
A.LIST	9
A.LOGSEG	9
A.LPSTART	37
A.MODE	17
A.MSGTYPE	12
A.OPCTL	21
A.OPTION	6
A.PAUSE	13
A.POLL	14
A.POLLIMIT	6
A.POSTARU	6
A.POSTRCV	6
A.POSTSEND	11
A.PROCESS	20
A.RCVHDR	5
A.RCVITA2	19
A.RCVSEG	5

QTAM Macros	Number of Library Blocks
A.RCVZSC3	19
A.RELEASEM	5
A.REPEAT	11
A.REROUTE	11
A.RETRIEVE	18
A.ROUTE	4
A.SENDHDR	7
A.SENDSEG	7
A.SEQIN	6
A.SEQOUT	6
A.SKIP	8
A.SNDITA2	24
A.SNDZSC3	24
A.SOURCE	4
A.STARTARU	18
A.STARTLN	15
A.STOPARU	15
A.STOPLN	15
A.STCBD	4
A.TERM	48
A.TERMTBL	44
A.TERMBLD	3
A.TIMESTMP	6
A.TRANS	17
A.WORD	9
A.WORDTBL	5
A.IJLQDEQU	65
A.IJLQDSCT	2
A.IJLQTSVC	104
A.LCBD	18
A.PREFIXD	13
A.QCBD	10
A.WRU	5
Total	1,859

IOCS Imperative Macros	Number of Library Blocks
A.CCB	17
A.CHECK	9
A.CLOSE	9
A.CLOSER	10
A.CNTRL	27
A.DISEN	6
A.DSPLY	8
A.ENDFL	6
A.ERET	3
A.ESETL	6
A.EXCP	4
A.FEOV	6
A.FREE	5
A.GET	8
A.LITE	8
A.LBRET	5
A.NOTE	6
A.OPEN	9

IOCS Imperative Macros	Number of Library Blocks
A.OPENR	13
A.POINTR	9
A.POINTS	6
A.POINTW	9
A.PRTOV	10
A.PUT	12
A.RDLNE	6
A.READ	30
A.RELSE	6
A.RESCN	12
A.SEOV	4
A.SETFL	5
A.SETL	9
A.TRUNC	6
A.WAIT	5
A.WAITF	17
A.WRITE	33
Total	335

COBOL and PL/I DASD Macros	Number of Library Blocks
A.LOADA	28
A.LODIS*	30
A.RANDA	28
A.RUADA	44
A.RRUIS*	32
A.RUAIS*	40
A.SEQDA	30
A.SRUIS*	38
Total	270

*Used by PL/I

PL/I Macro	Number of Library Blocks
A.IJKZL	6

MPS Utility Macros	Number of Library Blocks
A.INCARD	37
A.INDISK	124
A.INLOG	39
A.INTAPE	146
A.INTCR	296
A.OUTCARD	91
A.OUTDISK	170
A.OUTLOG	39
A.OUTPRT	110
A.OUTAPE	128
Total	1,180

Supervisor Generation Macros	Number of Library Blocks
A.ALLOC	22
A.ASSGN	14
A.COMMN	46
A.CONFG	17
A.DVCGEN	42
A.FOPT	211
A.IJLQBFRD	29
A.IJLQDSCT	2
A.IJLQIP1D	4
A.IJLQQCBD	26
A.IJLQSTBD	4
A.IJLQTSVC	104
A.IJLQVECD	12
A.IOTAB	146
A.LUBGEN	19
A.PIOCS	24
A.SEND	191
A.SGDFCH	96
A.SGDSK	49
A.SGSVC	260
A.SGTCHS	332
A.SGTCON	126
A.SGTHAP	295
A.SGUNCK	91
A.SMICR	209
A.STDJC	22
A.SUPVR	44
Total	2,438

Maintenance Macro	Number of Library Blocks
A.DOSCHLV	59
Z.ILFMERGE	11
Z.DELETECL	17
Z.DELETERL	41
Z.DELETESL	38
Z.LINKEDIT	114
Total	280

IBM Model 30 Emulator Macros	Number of Library Blocks
A.EU3CG	106
A.EU3DK	727
A.EU3EJ	61
A.EU3ER	30
A.EU3FT	37
A.EU3MS	66
A.EU3OS	669
A.EU3PH	239
A.EU3PT	171
A.EU3RD	235
A.EU3TP	306
A.EU30	859
Total	3,562

Sample Problems	Number of Library Blocks
Z.AS1	6
Z.AS2	6
Z.AS3	6
Z.AS4	6
Z.AS5	6
Z.AS6	6
Z.AT1	6
Z.CB1	25
Z.EU3SPRGM	88
Z.EU4SPRGM	88
Z.FO1	5
Z.ILFSAMPL	5
Z.MCR1	18
Z.MCR2	30
Z.ORDC	45
Z.ORJT	21
Z.PL1	11
Z.RG1	13
Z.SM1	2
Z.SM2	2
Z.SM4	2
Z.SM5	2
Z.SM6	2
Z.UTDCPR1	1
Z.UTDKPR1	1
Z.UTDKPR2	1
Z.UTDKPR3	1
Z.UTTPPR1	1
Z.UTTPPR2	1
Z.VFU1	17
Z.ILFSAMPL	5
Total	338

IBM Model 40 Emulator Macros	Number of Library Blocks
A.DIAG	19
A.EU4CG	60
A.EU4DK	689
A.EU4EJ	85
A.EU4ER	43
A.EU4FT	17
A.EU4IN	253
A.EU4MS	36
A.EU4OS	656
A.EU4PH	248
A.EU4PT	183
A.EU4RD	241
A.EU4TP	317
A.EU40	1047
Total	3,894

Workfile Requirements

This section contains information for determining the workfile requirements for the Linkage Editor, Assembler F, Assembler, COBOL, FORTRAN, Basic FORTRAN, PL/I, and RPG.

WORKFILE REQUIREMENTS FOR LINKAGE EDITOR AND ASSEMBLER (IBM 2311 AND IBM 2314)

This section contains information for determining the workfile requirements for the Linkage Editor and Assembler when the workfile is on an IBM 2311 or 2314.

Two workfiles are used by the Linkage Editor: SYSLNK for input, and SYS001 for a workfile.

The best overall performance for linkage editing results from using two disks and one tape: SYSRES and SYSLNK each assigned to a separate disk drive, and SYS001 assigned to tape. When possible, SYSLNK should be assigned to the faster of the two disks because more I/O is performed on SYSLNK than on SYSRES. Because linkage editing time is relatively small compared to assembler or compiler times, optimum assignments for assembling and linkage editing (or compiling and linkage editing) should be based on assembler or compiler conditions, not linkage edit time. Thus, when making optimum workfile assignments, the major consideration should be compiler time. A savings in linkage editing time is generally at the expense of compiler performance.

When built by a language processor, SYSLNK contains 25 card images per track. When an object deck is used as input to the Linkage Editor, job control formats SYSLNK. In this case, SYSLNK contains 9-36 card images per track; ESD, TXT, and RLD cards are packed 4 per record while all other input cards are not packed (1 per record).

In a compile and linkage edit situation, any allocation made for SYS001 for the compilation is more than sufficient as a workfile allocation for the Linkage Editor. However, when the programmer must allocate SYS001, he can use this information. The Linkage Editor workfile (SYS001) contains 10 records per track. The total number of records (R) required for linkage editing a program is equal to the following:

$$R = 1 + \left\lceil \frac{x_1}{4} \right\rceil + \left\lceil \frac{x_2}{4} \right\rceil + \; . \; . \; . \; + \left\lceil \frac{x_n}{4} \right\rceil$$

where $\lceil\,\rceil$ = rounded high

$x_1, x_2, \ldots, x_n$ = number of RLD cards in each module to be processed by the Linkage Editor

ASSEMBLER WORKFILES

The Assembler workfile requirements can be determined by adding the appropriate track values from Figure 75 or Figure 76 to the appropriate track values determined from Figure 74. Note that Figure 74 is expressed in terms of number of bytes. The approximate number of tracks can be calculated by dividing the number of bytes by 3000 for a 2311 file or by 6000 for a 2314 file. These numbers represent the approximate number of text bytes per track for a 2311 and a 2314, respectively. In Figures 75 and 76 SYSLNK requirements are expressed in terms of tracks per macro instruction; the other three areas are expressed in tracks per macro definition.

In determining the total number of tracks required for Assembler workfiles, the SYSLNK value must be included each time the macro instruction is used. The SYS001, SYS002, and SYS003 values need be included only once per macro definition, regardless of the number of times the macro instruction is used.

For example, if three tape files are defined, three DTFMT macros are used. The track values for SYS001, SYS002, SYS003 are included once because the Assembler uses macro definition only once; but the SYSLNK value is multiplied by three because the macro instruction is expanded three times.

		Number of Bytes			
		SYSLNK Bytes per Statement	SYS001 Bytes per Statement	SYS002 Bytes per Statement	SYS003 Bytes per Statement
IJQD16TW	1 for 1 Statements	15	150	150	36
IJQD16DW					
IJQD32 IJYASM		15	130	130	36

Figure 74. Non-Macro Assembly Workfile Requirements for 2311 and 2314

	Number of Tracks							
	SYSLNK Tracks per Macro Inst.		SYS001 Tracks per Macro Def.		SYS002 Tracks per Macro Def.		SYS003 Tracks per Macro Def.	
	2311	2314	2311	2314	2311	2314	2311	2314
Short Macros like IOCS Imperatives	0.02	0.13	1	1	2	1	1	1
CDMOD	0.15	0.09	11	6	3	7	17	8
DAMOD	0.38	0.24	10	5	13	7	9	4
DIMOD	0.55	0.35	8	4	11	6	11	5
DTFCD	0.05	0.03	6	3	7	4	9	4
DTFCN	0.08	0.05	4	2	5	3	6	3
DTFDA	0.08	0.05	9	5	10	6	13	6
DTFDI	0.36	0.23	4	2	4	2	5	2
*DTFEN	0	0	127	67	143	75	162	84
DTFIS	0.12	0.08	9	5	11	6	13	6
DTFMR	0.34	0.17	4	2	5	3	5	3
DTFMT	0.05	0.03	9	5	10	5	12	6
DTFOR	0.08	0.05	4	2	4	2	5	2
DTFPH	0.08	0.05	3	2	4	2	5	2
DTFPR	0.05	0.03	3	2	4	2	5	2
DTFPT	0.05	0.03	7	4	8	4	9	4
DTFSD	0.08	0.05	10	5	11	6	13	7
DTFSR	0.08	0.05	79	41	90	47	105	54
ISMOD	0.74	0.47	53	28	64	33	58	30

Figure 75. Assembler Macro Instruction Workfile Requirements for the IBM 2311 and IBM 2314 (Part 1 of 2)

	Number of Tracks							
	SYSLNK Tracks per Macro Inst.		SYS001 Tracks per Macrd Def.		SYS002 Tracks per Macro Def.		SYS003 Tracks per Macro Def.	
	2311	2314	2311	2314	2311	2314	2311	2314
MRMOD	0.67	0.34	10	5	12	6	8	4
MTMOD	0.23	0.15	22	12	23	12	27	14
ORMOD	0.20	0.13	11	6	16	8	15	7
PRMOD	0.12	0.08	5	3	6	4	8	4
PTMOD	0.15	0.09	13	7	13	7	14	7
SDMODFI	0.80	0.50	12	6	14	8	16	8
SDMODFO	0.80	0.50	12	7	15	8	16	8
SDMODFU	0.80	0.50	15	8	18	10	19	9
SDMODVI	0.80	0.50	11	6	12	6	14	7
SDMODVO	0.80	0.50	13	7	16	9	16	8
SDMODVU	0.80	0.50	13	7	17	9	16	8
SDMODUI	0.80	0.50	10	5	11	6	13	7
SDMODUO	0.80	0.50	11	6	12	6	14	7
SDMODUU	0.80	0.50	13	7	15	8	16	8
SDMODW	0.80	0.50	16	9	17	9	22	11

* The measurements for DTFEN, unlike the other macro instruction measurements, represent minimum workfile requirements. The measurements were made assembling DTFEN by itself, which resulted in minimum expansion. The use of DTFEN should be avoided wherever possible because of its large workfile requirements.

Figure 75. Assembler Macro Instruction Workfile Requirements for the IBM 2311 and IBM 2314 (Part 2 of 2)

	Number of Tracks						
	SYSLNK Tracks per Macro Inst.	SYS001 Tracks per Macro Def.		SYS002 Tracks per Macro Def.		SYS003 Tracks per Macro Def.	
	2311 and 2314	2311	2314	2311	2314	2311	2314
Short Macros like IOCS Imperatives	0.02	1	1	1	1	2	1
CDMOD	0.15	2	2	2	1	10	5
DAMOD	0.38	5	3	5	3	6	4
DIMOD	0.55	5	3	5	3	7	4
DTFCD	0.05	2	1	2	1	5	3
DTFCN	0.08	2	1	2	1	4	3
DTFDA	0.08	3	2	3	2	7	4
DTFDI	0.36	2	1	2	1	4	3
*DTFEN	0	2	1	2	1	121	62
DTFIS	0.12	3	2	3	2	7	4
DTFMR	0.34	2	1	2	1	4	3
DTFMT	0.05	2	1	3	2	7	4
DTFOR	0.08	2	1	1	1	3	2
DTFPH	0.08	1	1	1	1	3	2
DTFPR	0.05	2	1	1	1	3	2
DTFPT	0.05	2	1	2	1	6	3
DTFSD	0.08	2	1	3	2	8	4
DTFSR	0.08	2	1	3	2	70	38
ISMOD	0.74	18	17	15	15	26	25
MRMOD	0.67	5	3	4	2	7	4

Figure 76. Assembler F Macro Instruction Workfile Requirements for the IBM 2311 and IBM 2314 (Part 1 of 2)

	Number of Tracks						
	SYSLNK Tracks per Macro Inst.	SYS001 Tracks per Macro Def.		SYS002 Tracks per Macro Def.		SYS003 Tracks per Macro Def.	
	2311 and 2314	2311	2314	2311	2314	2311	2314
MTMOD	0.23	6	3	5	3	20	11
ORMOD	0.20	3	4	3	4	7	6
PRMOD	0.12	3	2	2	1	5	3
PTMOD	0.15	4	2	3	2	12	7
READ, WRITE	0.02	1	1	1	1	2	1
SDMODFI	0.80	4	3	4	2	12	6
SDMODFO	0.80	5	3	4	2	12	7
SDMODFU	0.80	6	3	5	3	15	8
SDMODVI	0.80	5	3	5	3	10	5
SDMODVO	0.80	7	4	6	3	12	6
SDMODVU	0.80	7	4	6	4	12	7
SDMODUI	0.80	5	3	4	2	9	5
SDMODUO	0.80	5	3	4	3	10	5
SDMODUU	0.80	7	4	6	3	12	6
SDMODW	0.80	6	4	6	3	13	7

* The measurements for DTFEN, unlike the other macro instruction measurements, represent minimum workfile requirements. The measurements were made assembling DTFEN by itself, whichis resulted in minimum expansion. The use of DTFEN should be avoided wherever possible because of its large workfile requirements.

Figure 76. Assembler F Macro Instruction Workfile Requirements for the IBM 2311 and IBM 2314 (Part 2 of 2)

IBM 2311 Workfile Requirements

This section contains information for determining the workfile requirements for Autotest, COBOL, FORTRAN, Basic FORTRAN PL/I and RPG when the workfile is on an IBM 2311.

AUTOTEST WORK AREA (IBM 2311)

To autotest a program with no symbols and with no test requests, the minimum work area (IJSYSAT) for machine sizes of 32K or higher requires 11 tracks. For machine sizes less than 32K, 6 tracks are required. A complete description of this work area, including formulas to calculate the required number of tracks for a specific program, is included in the Autotest publication.

COBOL WORKFILES (IBM 2311)

Although the amount of COBOL work space depends mainly on the size of the object program, these general guidelines can be given. For a COBOL program of about 400 source statements, 10 tracks should be assigned to SYSLNK, SYS001, SYS002, and SYS003. For a COBOL program of about 800 source statements, 20 tracks should be assigned to SYSLNK, SYS001, SYS002, and SYS003.

FORTRAN WORKFILES (IBM 2311)

For a FORTRAN program, 6 tracks should be assigned to SYSLNK and 4 tracks to SYS001 and SYS002 for every 100 source statements.

For a Basic FORTRAN program, 4 tracks should be assigned to SYSLNK and to SYS001 for every 100 source statements.

PL/I WORKFILES (IBM 2311)

For each 100 PL/I source statements, 3 tracks should be reserved for SYSLNK; 5 tracks for SYS001; and 7 tracks each for SYS002 and SYS003.

This estimate is based on the following assumptions:

1. There are about 30 variable names per 100 statements in the external procedure.

2. The number of PL/I syntactical elements per source statement (except DECLARE statements and format lists) is about 10. For example, the statement
   ```
   A=B+C;
   ```
   has 6 syntactical elements, and the statement
   ```
   READ FILE (F) INTO (AREA7);
   ```
   has 10.

RPG WORKFILES (IBM 2311)

For an RPG program of about 150 source statements, 8 tracks should be assigned to SYSLNK and to SYS003, and 3 tracks to SYS001 and to SYS002.

For an RPG program of about 500 source statements, 22 tracks should be assigned to SYSLNK; 10 tracks to SYS001 and to SYS002; and 26 to SYS003.

For an RPG program of about 1,000 source statements, 44 tracks should be assigned to SYSLNK; 20 tracks to SYS001 and to SYS002; and 52 to SYS003.

IBM 2314 Workfile Requirements

This section contains information for determining the workfile requirements for COBOL, FORTRAN, Basic FORTRAN and RPG when the workfile is on an IBM 2314.

COBOL WORKFILES (IBM 2314)

Although the amount of COBOL work space depends mainly on the size of the object program, these general guidelines can be given. For a COBOL program of about 400 source statements, 5 tracks should be assigned to SYSLNK, SYS001, SYS002, and SYS003. For a COBOL program of about 800 source statements, 10 tracks should be assigned to SYSLNK, SYS001, SYS002, and SYS003.

FORTRAN WORKFILES (IBM 2314)

For a FORTRAN program, 3 tracks should be assigned to SYSLNK and 2 tracks each to SYS001 and SYS002 for every 100 source statements.

For a Basic FORTRAN program, 2 tracks should be assigned to SYSLNK and to SYS001 for every 100 source statements.

PL/I WORKFILES (IBM 2314)

For each 200 PL/I statements, 3 tracks should be reserved for SYSLNK; 5 tracks for SYS001; and 7 tracks each for SYS002 and SYS003. This estimate is based upon the following assumptions:

1. There are about 30 variable names per 100 statements in the external procedure.

2. The number of PL/I syntactical elements per source statement (except DECLARE statements and format lists) is about 10. For example, the statement
   A=B+C
   has 6 syntactical elements; the statement
   READ FILE (F) INTO (AREA7)
   has 10 syntactical elements.

RPG WORKFILES (IBM 2314)

For an RPG program of about 150 source statements, 5 tracks should be assigned to SYSLNK and to SYS003, and 2 tracks to SYS001 and to SYS002.

For an RPG program of about 500 source statements, 11 tracks should be assigned to SYSLNK, 5 tracks to SYS001 and to SYS002, and 13 to SYS003.

For an RPG program of about 1,000 source statements, 22 tracks should be assigned to SYSLNK, 10 tracks to SYS001 and to SYS002, and 26 to SYS003.

OPTIMUM ASSIGNMENT OF WORKFILES

Figure 77 lists the optimum assignments of the symbolic units used as workfiles when assembling and linkage editing or compiling and linkage editing user programs. While SYSRES and SYSLNK must be assigned to disk units, SYSnnn can be assigned to either tape or disk units. Where split cylinders are recommended, Figure 77 gives the division of the tracks in each cylinder between the symbolic units.

Key to figures:

- Workfile assignments are given by an integer and a letter n,m where:

  n = the disk unit.

  m = the disk cylinder(s) group assigned for the symbolic unit.

- Cylinder track numbers are given by integer(s) t where:

  t = the number of disk tracks per cylinder to be assigned for the symbolic unit.

- Tape speeds are given by an integer s where:

  s = the relative speed of the tape unit (the number 1 is the fastest speed).

- Tape channels are given by an integer c where:

  c = the channel that the tape should be on.

- If two symbolic units have the same disk and cylinder(s) assignment indication (n,m), they share a split cylinder.

- Where applicable, the assignments that give the best overall performance are indicated.

Figure 77 (Part 1 of 3), shows that for a two disk system, disk workfile assignments for Assembler (14K variant) are:

Assign SYSRES	Disk drive number 1 Cylinder group A
Assign SYSLNK	Disk drive number 1 Cylinder group C
Assign SYS001	Disk drive number 2 Cylinder group A
Assign SYS002	Disk drive number 2 Cylinder group A

Assign SYS003	Disk drive number 1 Cylinder group B

Interpreting the information collectively reveals that SYSRES, SYSLNK and SYS003 are assigned separate cylinders on disk 1, and SYS001 and SYS002 share the same cylinder group (A) on disk 2. Thus, Cylinder group (A) is split such that SYS001 and SYS002 each occupy 5 tracks of every cylinder in the group. To define the extents for workfiles, refer to the IBM publication System Control and System Service Programs, listed on the front cover of this manual.

Workfile Assignments For Two Disk Drive Systems							
Symbolic Unit Requirement		Assembler[1] (14K Variant)	FORTRAN	Basic FORTRAN	COBOL	PL/I[3]	RPG[2]
SYSRES	Disk Drive Number (n)	1	1	1	--	1	1
	Cylinder Group (m)	A	A	B	--	A	A
	Tracks Per Cylinder (t)	0-9	0-9	0-9	--	0-9	0-9
SYSLNK	Disk Drive Number (n)	1	2	2	2	2	2
	Cylinder Group (m)	C	A	A	B	B	A
	Tracks Per Cylinder (t)	0-9	0-3	0-4	0-9	0-9	0-6
SYS001	Disk Drive Number (n)	2	2	2	2	2	2
	Cylinder Group (m)	A	A	A	A	A	A
	Tracks Per Cylinder (t)	0-4	4-6	5-9	0-3	0-3	7,8,9
SYS002	Disk Drive Number (n)	2	2	Not Req'd	2	2	2
	Cylinder Group (m)	A	A	--	A	A	B
	Tracks Per Cylinder (t)	5-9	7-9	--	4,5,6	4,5,6	0-9
SYS003	Disk Drive Number (n)	1	Not Req'd	Not Req'd	2	2	2
	Cylinder Group (m)	B	--	--	A	A	C
	Tracks Per Cylinder (t)	0-9	--	--	7,8,9	7,8,9	0-9

Figure 77. Optimum Workfile Assignments for Assembling and Linkage Editing or Compiling and Linkage Editing (Part 1 of 3)

Workfile Assignments for One Disk Drive							
Symbolic Unit Requirement		Assembler (14K Variant)	FORTRAN	Basic FORTRAN	COBOL	PL/I	RPG
SYSRES	Disk Drive Number (n)	1	1	1	--	1	1
	Cylinder Group (m)	A	A	A	--	A	A
	Tracks Per Cylinder (t)	0-9	0-9	0-9	--	0-9	0-9
SYSLNK	Disk Drive Number (n)	1	1	1	1	1	1
	Cylinder Group (m)	D	B	B	B	B	B
	Tracks Per Cylinder (t)	0-9	0-9	0-4	0-9	0-9	0-6
SYS001	Disk Drive Number (n)	1	1	1	1	1	1
	Cylinder Group (m)	B	A	B	A	C	B
	Tracks Per Cylinder (t)	0-4	0-3	5-9	0-3	0-3	7,8,9
SYS002	Disk Drive Number (n)	1	1	Not Req'd	1	1	1
	Cylinder Group (m)	B	A	--	A	C	C
	Tracks Per Cylinder (t)	5-9	4-9	--	4,5,6	4,5,6	0-9
SYS003	Disk Drive Number (n)	1	Not Req'd	Not Req'd	1	1	1
	Cylinder Group (m)	C	--	--	A	C	D
	Tracks Per Cylinder (t)	0-9	--	--	7,8,9	7,8,9	0-9

Figure 77. Optimum Workfile Assignments for Assembling and Linkage Editing or Compiling and Linkage Editing (Part 2 of 3)

Workfile Assignments for One Disk Drive with Three Tape Units							
Symbolic Unit Requirement[5]		Assembler[1] (14K Variant)	FORTRAN	Basic[2] FORTRAN	COBOL	PL/I	RPG
SYS001	Tape[4] Speed (s)	3	1	Any	2	2	1
	Channel Require-ment (c)	2	1	Any	2	2	2
SYS002	Tape[4] Speed (s)	2	2	Not Req'd	1	1	2
	Channel Require-ment (c)	1	1	--	1	1	1
SYS003	Tape[4] Speed (s)	1	Not Req'd	Not Req'd	1	1	2
	Channel Require-ment (c)	2	--	--	2	2	2

Note 1: The configuration that yields the best overall performance depends upon the variant of the assembler used (10K or 14K) and the storage available. The assignments shown assume that the 14K variant of the Assembler is used in a 64K environment. For cases with over approximately 500 lines of output, the two disk drive configuration yields the best overall performance. For cases with under approximately 500 lines of output, the three tape drive configuration yields the best overall performance.

Note 2: Indicates best overall performance.

Note 3: Two disk drive chart- For PL/I only users, compilation times will be faster if the following assignments are made for SYS001, SYS002 and SYS003 in place of those given.

Assign SYS001 to drive 2, group A, tracks 0-4
Assign SYS003 to drive 2, group A, tracks 5-9
Assign SYS002 to drive 1, group C, tracks 0-9

The assignments for SYSRES and SYSLNK are the same.

Note 4: Relative tape speeds are indicated by 1, 2 or 3. 1 designates the fastest tape unit, 2 the next fastest, etc.

Note 5: SYSRES and SYSLNK must always be assigned to disk in a one disk drive, three tape drive configuration.

Figure 77. Optimum Workfile Assignments for Assembling and Linkage Editing or Compiling and Linkage Editing (Part 3 of 3)

Index

Reader's Comments Form

IBM System/360
Disk Operating System
System Generation and Maintenance

C24-5033-7

Your comments, accompanied by answers to the following questions, help us produce better publications for your use. If your answer to a question is "No" or requires qualification, please explain in the space provided below. Please give specific page and line references with your comments when appropriate. All comments will be handled on a non-confidential basis. Copies of this and other IBM publications can be obtained through IBM branch offices.

	Yes	*No*
• Does the publication meet your needs?	☐	☐
• Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

• What is your occupation? ______________________________

• How do you use this publication:

As an introduction to the subject?	☐	As an instructor in a class?	☐
For advanced knowledge of the subject?	☐	As a student in a class?	☐
For information about operating procedures?	☐	As a reference manual?	☐

Your comments:

Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

C24-5033-7

Staple

Your comments, please . . .

This publication is one of a series that serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, help us produce better publications for your use. Each reply is carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in using your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

Fold | Fold

FIRST CLASS
PERMIT NO. 170
ENDICOTT, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation
P. O. Box 6
Endicott, N. Y. 13760

Attention: Programming Publications, Dept. 157

Fold | Fold

IBM

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

Additional Comments:

C24-5033-7

IBM

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]