IBM VSE/Enterprise Systems Architecture
VSE Central Functions

Linkage Editor Diagnosis Reference

Version 6 Release 1

SC33-6328-00

IBM VSE/Enterprise Systems Architecture
VSE Central Functions

Linkage Editor Diagnosis Reference

Version 6 Release 1

SC33-6328-00

Note!

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

First Edition (April 1995)

This edition applies to Version 2 Release 1 of IBM Virtual Storage Extended/Enterprise Systems Architecture (VSE/ESA), Program Number 5690-VSE, and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters.

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

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

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

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

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

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

Contents

Figures

Notices

References in this publication to IBM* products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

This publication is intended primarily for use by IBM personnel responsible for program service. The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. It is not intended as a description of a programming interface. The use of this information is a customer reponsibility. Service for errors, ommissions, accuracy, or completeness will not be provided.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase NY 10577, U.S.A.

Programming Interface Information

This publication is intended to help the customer to do diagnosis of VSE/ESA. This publication documents information that is Diagnosis, Modification, or Tuning Information provided by VSE/ESA.

Warning: Do not use this Diagnosis, Modification, or Tuning Information as a programming interface.

Trademarks and Service Marks

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

ESA/390
IBM
System/370
System/390
VSE/ESA

Preface

This Diagnosis Reference documents the code of the linkage editor component of VSE/ESA for the task of program service.

The manual consists of the following chapters:

- INTRODUCTION to the linkage editor, showing the context of the component in the system, the input and output of the program, and the different ways of calling it.
- DESIGN INFORMATION, describing the function, I/O flow, storage layout, and division into CSECTs and modules of the program. A description of the linkage editor phase gives all relevant information from the prologue of the listing. A detailed description shows the sequence of operations for each CSECT with the essential labels. A flow chart shows the basic flow of control between the CSECTs of the program.
- DATA AREAS, describing the formats of library records used by the program, linkage editor tables, and the librarian areas used by the linkage editor.
- DIAGNOSTICS which contains interface information and cross references.

An index at the end of the manual will help to find details fast.

The text refers to the following IBM manuals:

- *VSE/ESA Guide to System Functions*, SC33-6611
- *VSE/ESA System Control Statements*, SC33-6613
- *VSE/ESA Diagnosis Tools*, SC33-6614
- *VSE/AF Diagnosis Reference: Librarian*, SC33-6330

For titles and abstracts of other related publications, see *VSE/ESA Library Guide*, GC33-6619.

Summary of Changes

VSE/ESA Version 1 Release 3 has the following technical improvements that are documented in this manual:

- 31-Bit Virtual Addressing:

  Support of the AMODE and RMODE attributes in the ESD input data.

  Support of the new MODE control statement.

  Support of the new AMODE and RMODE parameters in the PARM field of the EXEC LNKEDT statement.

VSE/ESA Version 2 Release 1 has the following additional technical improvements that are documented in this manual:

- Support of pseudo registers as provided by PL/I compilers resp. external dummy sections as provided by the High Level Assembler.
- Support of named common control sections.
- Introduction of free positional ACTION statements.
- Introduction of ACTION option 'ERRLMT' (error limit).
- Indication of offsets for 2/3-byte address constants, if warning message I2165 was issued.

Introduction

System Context

All programs to be executed under VSE/ESA must be prepared first by a language translator and then by the linkage editor. According to these stages of preparation, programs may be stored in any sublibrary as one of the predefined member types.

```
Source modules have a one-character type (A-Z,0-9,#,$,@).
  (Each compiler defines its valid source module types.)
Object modules have type OBJ.
Phases have type PHASE.
```

The linkage editor can run in any partition.

Input

Input consists of the linkage editor statements and object modules to be linked. Linkage editor statements are ACTION, ENTRY, MODE, INCLUDE, and PHASE. For the specification of these statements see *VSE/ESA System Control Statements*, SC33-6613. The sequence of sublibraries to be searched for the input object modules is defined by a LIBDEF OBJ,SEARCH job statement. This sequence is the OBJ-search chain and a sublibrary in this chain is called a SEARCH sublibrary.

Output

Output consists of the phase or phases produced and a linkage editor map giving address information about each phase and CSECT. The phase produced is stored either temporarily or permanently before it can be executed. For // OPTION LINK the phase is stored temporarily in the virtual I/O area, an extension of the page data set. For // OPTION CATAL the phase is stored permanently in the CATALOG sublibrary with an entry in the sublibrary member index. The CATALOG sublibrary is defined via the LIBDEF PHASE,CATALOG job statement. The member index is updated via library management. The linkage editor map is printed on SYSLST after the linkage editor control statements and input listing.

Invocation

Job control calls the linkage editor when it reads a // EXEC LNKEDT statement. This statement can occur in two different combinations:

1. To link and catalog: An object module is link-edited and the resulting phase is *permanently stored* or “cataloged” in a sublibrary. If the phase is re-enterable it can be declared SVA-eligible in the PHASE statement. The job statements to “link and catalog” are:

   ```
   // OPTION CATAL
   // EXEC LNKEDT
   ```

2. To link, load, and execute: An object module is link-edited and the resulting phase is *temporarily stored* in the virtual I/O area and *immediately executed*. The phase to be link-edited and immediately executed must not be part of an overlay structure. The job statements to link, load, and execute are:

   ```
   // OPTION LINK
   // EXEC LNKEDT
   // EXEC
   ```

MSHP Module Control

Phases cataloged under control of MSHP must be maintained via MSHP. Such phases are flagged in the directory entry as being MSHP controlled. This flag normally prevents the replacement of the phase by the linkage editor when not running under control of MSHP.

To allow replacement of such a phase out of control of MSHP, the MSHP bypass function is provided. If this bypass is specified, the linkage editor replaces MSHP-flagged phases and sets another flag to indicate that the phase has been changed without MSHP control.

The MSHP bypass is specified as PARM parameter as follows:

```
// EXEC LNKEDT,PARM='MSHP'
```

Controlled Operator Cancel

Cancelling a librarian or linkage editor job in a release prior to VSE/Advanced Functions Version 2 Release 1 was always critical since the library used was in danger to be destroyed. For Version 2 Release 1 (and all later versions) of VSE/Advanced Functions the librarian or linkage editor continues processing up to a point where a consistent state of the library is reached whenever a librarian command or a linkage editor job is cancelled normally, i.e. not with CANCEL FORCE.

CANCEL FORCE always terminates immediately with risk of library damage.

The normal cancel function is controlled by the flags IJBARCNA and IJBCNCPD in field JCSW8 of the partition COMREG. These flags are set/reset/checked by the linkage editor module INLPLEIT.

Linkage Editor Call Interface

This section describes the invocation of the Linkage Editor from a program within a partition and the input/output interfaces provided.

The Linkage Editor may be invoked from a (problem) program by loading the linkage editor (phase $LNKEDT) into partition storage and branching to its entry point. The Linkage Editor will take the rest of the partition as working space. Therefore, $LNKEDT must be loaded in the upper part of the partition (recommended size reserved for $LNKEDT and working space: 512-1024KB; see *VSE/ESA Guide to System Functions*, SC33-6611, for linkage editor storage requirements.).

The input stream for the linkage editor must have been set up in SYSLNK. (For this purpose Job Control may be used in the same way as when the Linkage Editor is called without this Call Interface. Note that Job Control writes as last statement an ENTRY statement without operands behind the input stream onto SYSLNK. The ENTRY statement indicates the end of the input stream to the Linkage Editor.)
A program may receive from the Linkage Editor the SYSLOG and SYSLST output in its own areas. It may get control back from the Linkage Editor whenever information is put into a caller-supplied area.

Interface Description:

Invocation of the Linkage Editor:

```
Load phase $LNKEDT (entry point in R15).
Load address of 18-FW register save area in R13.
Load address of parameter list in R1.
BALR R14,R15.
```

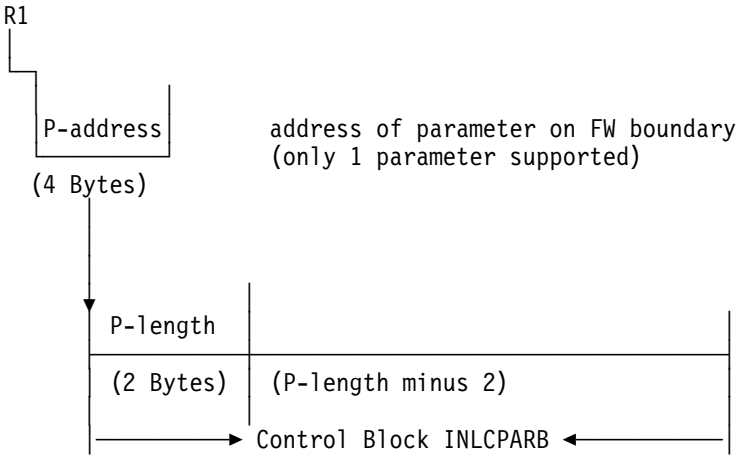
The Linkage Editor returns to the R14-address and passes back a return code in R15.

Return Codes:

```
  0 :  Successful.
  2 :  Warning message issued, but phases are cataloged.
  4 :  Warning or error message issued, but phases are
       cataloged.
  8 :  Some phases are not cataloged because they already
       exist in the target sublibrary.
       (This return code is only given for the
        CALL interface if REPLACE=NO is requested via
        bit PARMNORP in control block INLCPARB.)
 16 :  Severe error occured, phases are not cataloged.
```

Passing a Parameter:

```
R1 <>  R15 :  R1 ──► address of parameter list
```



The parameter-list has the same layout as the parameter-list used for the Librarian Call Interface, see *VSE/AF Diagnosis Reference: Librarian* SC33-6330, data area INLCPARB.

Note: If called by Job Control (// EXEC LNKEDT,PARM='...,...') the PARM contents is transferred in a similar way: R1 points to a field containing a 2-byte length field, followed by the PARM contents, i.e. the data between the apostrophes.

```
Control Block Layout (use INCLUDE-book INLCPARB):
       DC    AL2(64)        Length of control block (P-length)
       DC    XL2'00'        Reserved
       DC    XL4            Identification of caller
* identifications: MSHP - XL4'01',
                   DSNX - XL4'03'
       DC    AL4            (not used by Linkage Editor)
       DC    AL4            (not used by Linkage Editor)
       DC    AL4            Address of SYSLOG exit routine
       DC    AL4            Address of SYSLOG output area (and
                                  input area for A-messages)
       DC    AL4            Address of SYSLST exit routine
       DC    AL4            Address of SYSLST output area
       DC    AL4            (not used by Linkage Editor)
       DC    AL4            (not used by Linkage Editor)
       DC    XL1            Current line count (SYSLST)
       DC    XL1            (not used by Linkage Editor)
       DC    XL1            Flagbyte
                            (X'80' - not used by Linkage Editor)
                            (X'40' - not used by Linkage Editor)
                            (X'20' - MSHP bypass function requested
                                     via Call Interface)
       DC    XL1            Flagbyte
                            (X'20' - OPTON CATAL requested via
                                     Call Interface)
                            (X'10' - no replacement of phases)
       DC    AL4            User area (not used by Linkage Editor)
* pointers for formatted output functions (not used by Linkage Editor)
       DC    AL4            Address of formatted output exit routine
       DC    AL4            Address of formatted output area
       DC    CL2            (not used by Linkage Editor)
       DC    XL1            Flagbyte
                            AMODE/RMODE PARM field specification
                            via Call Interface:
                            (X'80' - RMODE parameter specified via
                                     Call Interface)
                            (X'40' - AMODE parameter specified via
                                     Call Interface)
                            (X'04' - 1:  RMODE=ANY, 0:  RMODE=24)
                            (X'03' - 11:       AMODE=ANY,
                                     10:       AMODE=31,
                                     01 or 00: AMODE=24)
       DC    AL1            (not used by Linkage Editor)
       DC    CL4            Reserved
```

The user area is not read or modified by the Linkage Editor. It may be used by the routines invoking the Linkage Editor to save information (for example, information which is used in an exit routine).

If the Linkage Editor is called by MSHP (identification '01'), the flag 'Call Interface Active' (X'80' in LEITCISW) is not set. The MSHP call is teated in a special way. The only normal Call Interface usage is by DSNX (identification '03').

Any of the exit routine addresses may be zero, which indicates that this exit must not be taken by the Linkage Editor,

that is, the Linkage Editor

- writes on SYSLOG/SYSLST if the corresponding addresses in the control block are zero.

If an exit routine address is not zero, the corresponding area address must also be unequal zero. Before entering an exit routine, the Linkage Editor loads the address of the input control block into Register 1

```
Calling routine:       CALL xyzproc(arg);
Invoked routine:       xyzproc:PROC(arg);  or  xyzproc:ENTRY(arg);
```

Either both addresses (exit and area) must be zero or both addresses must be unequal zero (otherwise: system error).

The SYSLOG/SYSLST output areas will contain one logical record when the corresponding exit is taken. If the logical record does not fit into the output area provided by the invoking routine, the record is cut without notice.
If the logical record is smaller than the output area, the rest of the area is padded with blanks for SYSLOG/SYSLST.

Layout of Output Areas (SYSLOG,SYSLST):

```
DC   XL2                Length of output area (8<length<256 bytes)
DC   XL6                Reserved
DC   CL(length minus 8) Output record (one)
```

The output area contains one output record at a time. The maximum length of the output area is 255 bytes and, consequently, the maximum length of the output record is 247 bytes.
For SYSLST: The first byte contains the Printer Control Character.

Restrictions: When invoking the Linkage Editor from a program the following restrictions apply:

- For each call of the linkage editor the linkage editor must be loaded again as the linkage editor is not serially reusable.
- The 'Delayed Cancel' feature must not be activated by the calling program since it will be activated by the Linkage Editor (that is, flag IJBARCNA in field JCSW8 of the partition communication region must not be set).
- System Files assigned to DASD must be closed before calling $LNKEDT, if the corresponding EXIT will not be taken.

Design Information

Function

The linkage editor adjusts the addresses in the CSECTs of one or several object modules so that these CSECTs can be executed together as one phase or as several phases, e.g. in an overlay structure.

Input

Input to the linkage editor are the linkage editor statements and the object modules produced by language translators (assembler or compiler).

Each object module consists of dictionaries (ESD and RLD) and text for one or more control sections.

Six record types can be produced by the language translators out of the source input to form a module. They appear in the following order:

Rec. Type	Definition
ESD	External symbol dictionary
SYM	Ignored by the linkage editor
TXT	Text
RLD	Relocation list dictionary
REP	Replacement to text by programmer
END	End of module

The dictionaries contain the information necessary for the linkage editor to resolve references between different modules and to relocate the program. The text consists of executable instructions and data fields.

Figure 1 shows an overview of the linkage editor input.

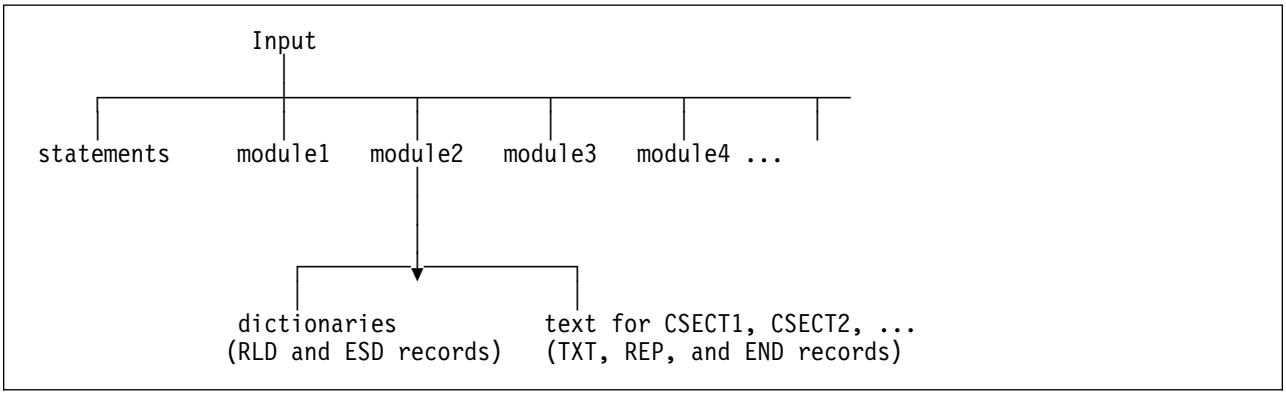


Figure 1. Overview of Input Units

The linkage editor can also re-link phases if they have been retransformed into modules by the librarian command PUNCH. In this way, an already link-edited phase can be recataloged to another library and sublibrary. The librarian command PUNCH causes the contents of a member of type PHASE to be punched as TXT and RLD records.

Since the greatest task of the linkage editor is the adjustment of external symbols, ESD records have the most variety.

ESD Records: The following types of ESD records exist.

- Section Definition (SD): Consists of CSECT name, assembled origin, addressing mode (AMODE) and residency mode (RMODE) attributes, and length of a named CSECT. Generated by a START or CSECT statement in the source module.
- Private Code (PC): Consists of assembled origin, addressing mode (AMODE) and residency mode (RMODE) attributes, and length of an unnamed CSECT. Generated by a START or CSECT statement.
- Label Definition or Label Reference (LD/LR): Contains the name, the assembled address and the associated CSECT ESID of a label that may be referred to by another module. The LD entry is termed LR (label reference) when the entry is matched to an ER entry. Generated by an ENTRY statement.
- Pseudo Register (PR): Contains name, length, and alignment information of a pseudo register (other term: external dummy section). Generated by a DXD or DSECT statement, that is referred to by a Q-type address constant. A pseudo register (or external dummy section) is similar to common storage (COM), but its storage space is allocated only during execution (e.g. by GETVIS).
- External Reference (ER): Contains the name of a reference in another module. Generated by the assembler instruction EXTRN or a V-type address constant in the source module.
- Weak External Reference (WX): Same content as ER, except that WX suppresses Autolink. Generated by the assembler instruction WXTRN.
- Common (CM): Contains the length of “common storage” needed by a particular phase at execution time. Common storage is an area to be reserved at the beginning of the partition for shared use between phases. Its most frequent use is for communication in an overlay structure. Generated in the source module by the assembler instruction COM or directly from the compilers.

Output

The output consists of the phases produced and a linkage editor map giving address information about each phase. See the description of the linkage editor map in the Chapter “Diagnostics.”

Operation

The program takes the CSECTs out of several modules and combines them, in a different selection or sequence if so specified, into executable phases.

Figure 2 on page 9 shows how phases can be formed.

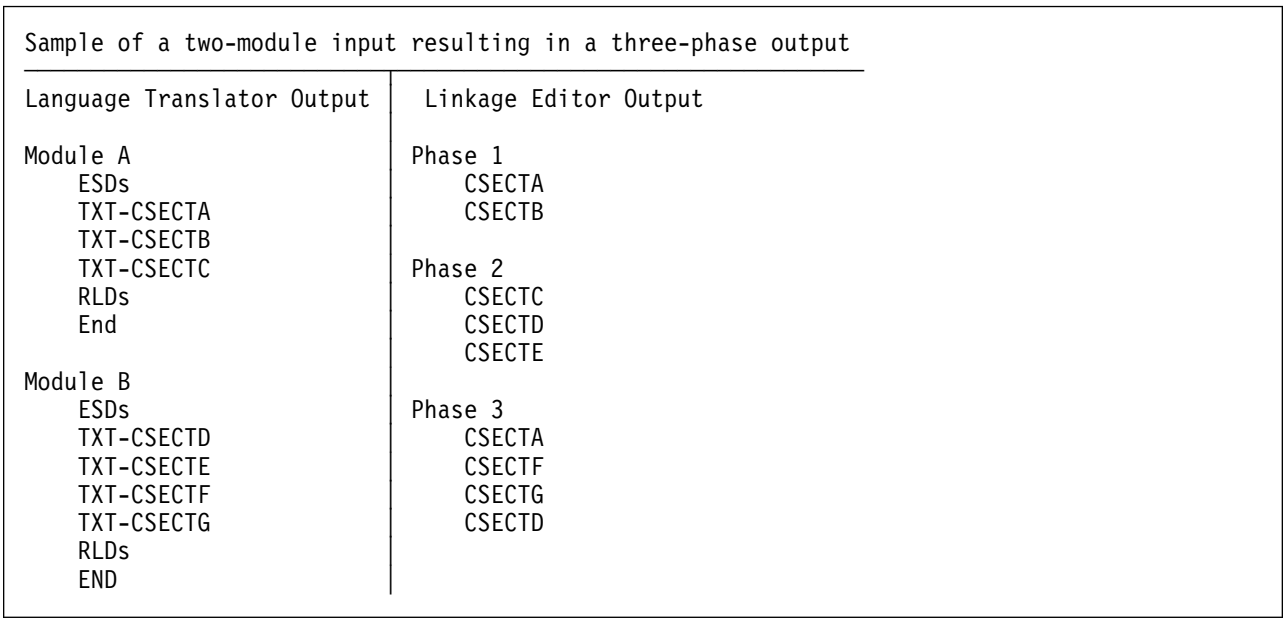


Figure 2. Example of a Module-Phase Relationship

When the linkage editor reads a module, it stores the ESD information in its control dictionary (CD), writes the information from the TXT and REP records into the sublibrary, and the RLD items on an internal buffer called RLD buffer. If the RLD buffer is full, its content is written on workfile IJSYS01.

The CD contains the information to find each CSECT and to resolve any reference between CSECTs.

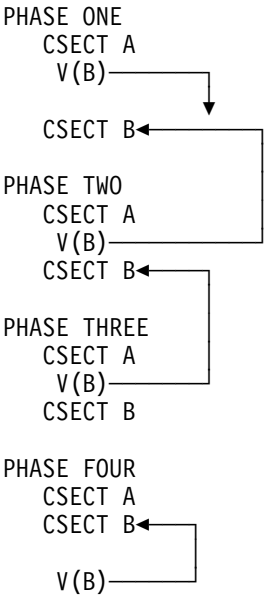
The language translator gives each ESD record a number called ESID number. The linkage editor gives it a CD number unique in the link job, because the same ESID number might occur several times coming from the different modules.

In detail, the linkage editor does the following:

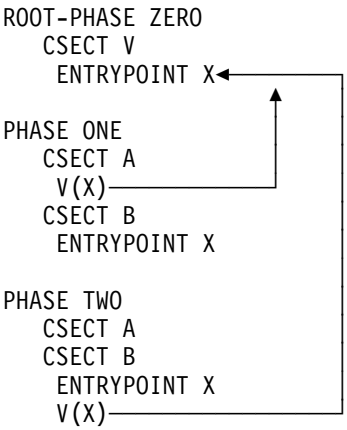
1. Relocates the origin of each CSECT in the phase:
   The relocation factor for each CSECT is determined and saved in the CD.
2. Assigns to each phase an area of storage, determines its transfer address, determines its addressing mode (AMODE) and its residency mode (RMODE), and combines the module TXT records into phase blocks using librarian services.
3. Resolves address cross-references across CSECTs, modules or phases, e.g in overlay structures.
4. Adjusts the contents of the address constants (ADCONs) in the phase and inserts them in the text. The contents of unresolved ADCONs is set to zero. For relocatable phases ADCONs are identified (location, length, sign of relocation factor) in the list of RLD items generated at the end of the cataloged phase. Unresolved ADCONs appear as zero RLD items.

As we have seen above, the linkage editor allows the inclusion of the same control section (CSECT) within each of several phases. If a CSECT appears in a ROOT phase, it does not appear in any other phase. (This does not apply to CSECTs that begin with the letters 'IBM'.) A duplicate CSECT within the same phase will be ignored.

The following examples show how *external references* are resolved, depending on whether or not a ROOT phase exists. The first example shows how external references are resolved when there is no ROOT phase:

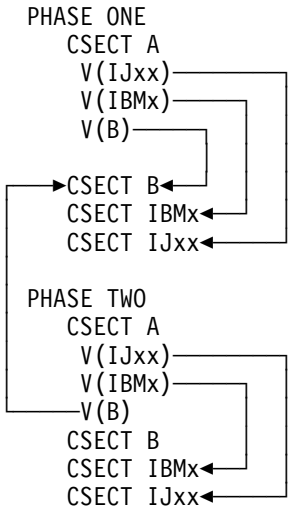


The second example shows the resolution of external references *with* a ROOT phase:



Privileged external references (names beginning with the letters IJ or IBM) are always resolved within the current phase or the ROOT phase. If this is not possible, the resolution will be attempted at the end of the phase via the AUTOLINK function (if NOAUTO is specified, the IJ or IBM prefix is not privileged). The other previously defined phases are not examined for possible resolution. If an external reference does not match the name of a module in the sublibraries to be searched, it will be an unresolved external reference.

The following example shows the resolution of privileged external reference:

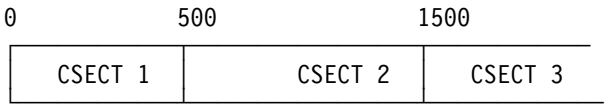


Autolink

Autolink is a feature that automatically includes modules with names of unresolved external references at the end of a phase. The modules are searched in the OBJ-search chain. They must have an ENTRY of the same name, which is then linked with the unresolved reference.

How the Linkage Editor Calculates Addresses

The assembled origin is the address of a CSECT or ENTRY within its object module. This assembled CSECT origin can be declared by assembler statements. Default is 0 for the first, 0 + length of the first for the second CSECT, and so on. (The HLASM sets each CSECT address to 0.)



The phase origin is the address to which the phase is linked. This address can be specified in the PHASE statement. The default is the beginning address of the linkage editor partition plus the length of the save area.

The expression “phase origin” is used also for the beginning of each CSECT after it was linked into a phase (NXPHRG).

The difference between the assembled origin and the load address of a CSECT or ENTRY is called the relocation factor.

If the phase is relocatable and is loaded somewhere else later, the address where it is loaded is called the load origin.

The load process has to add the difference between load origin and phase origin to each address constant in the text. To do this the addresses of the ADCONs, contained in the RLD item list at the end of the phase, have to be updated accordingly.

An example address distribution for a phase linked from two object modules, with Common Sections, is given in Figure 3 on page 12.

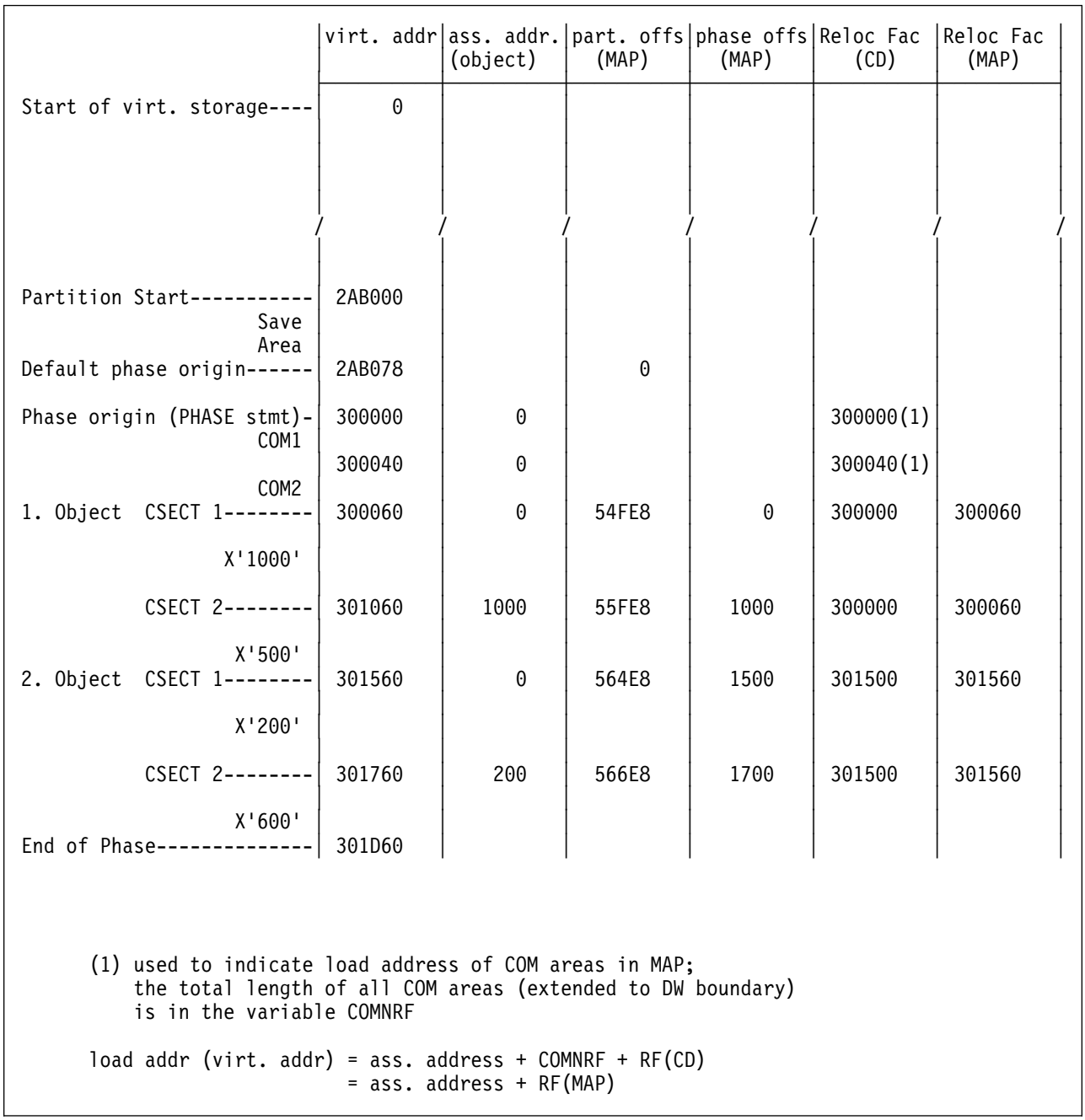


Figure 3. Address Calculation

I/O Flow and Library Access Control

As explained in the introduction to this manual, library access is gained for the linkage editor by job control via the LIBDEF statement. Input modules are accessed by a LIBDEF OBJ,SEARCH statement. The CATALOG sublibrary to which the output phases are written is defined via LIBDEF PHASE,CATALOG statement.

All librarian services required by the linkage editor are controlled by the interface CSECT INLPLEIT. They are called from various CSECTs using the LEIT macro, which makes use of the LEITCALL subroutine to link to INLPLEIT. Buffers are provided for use by the librarian. Communication is done via the LEITPL parameter list.

The temporary output of a phase to the VIO area is done by the virtual I/O (VIO) routines of the supervisor which are called also by CSECT INLPLEIT using the macro VIO with the operands OPEN, MOVE, or EXTND.

Figure 4 shows the I/O flow in the linkage editor program.

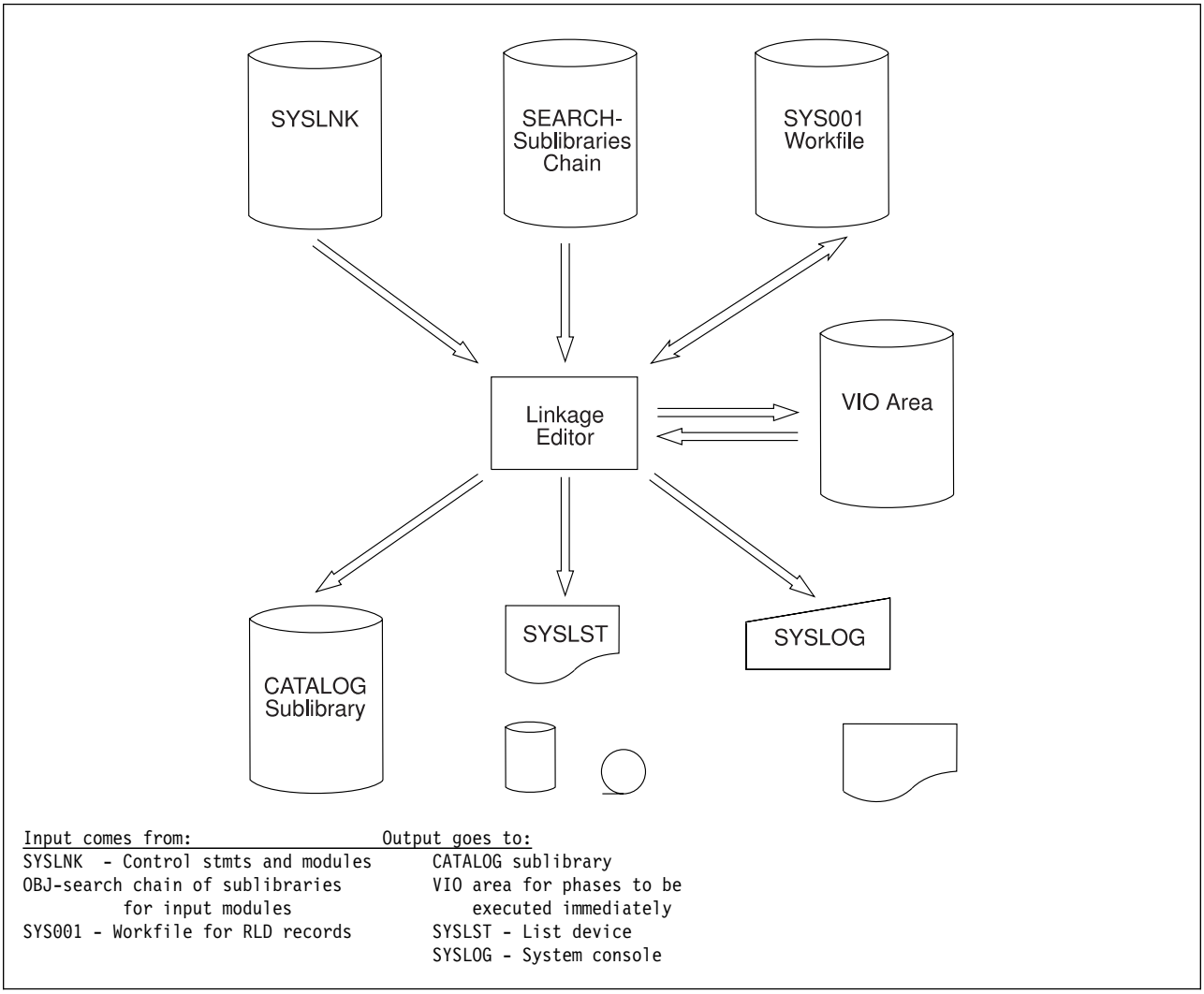
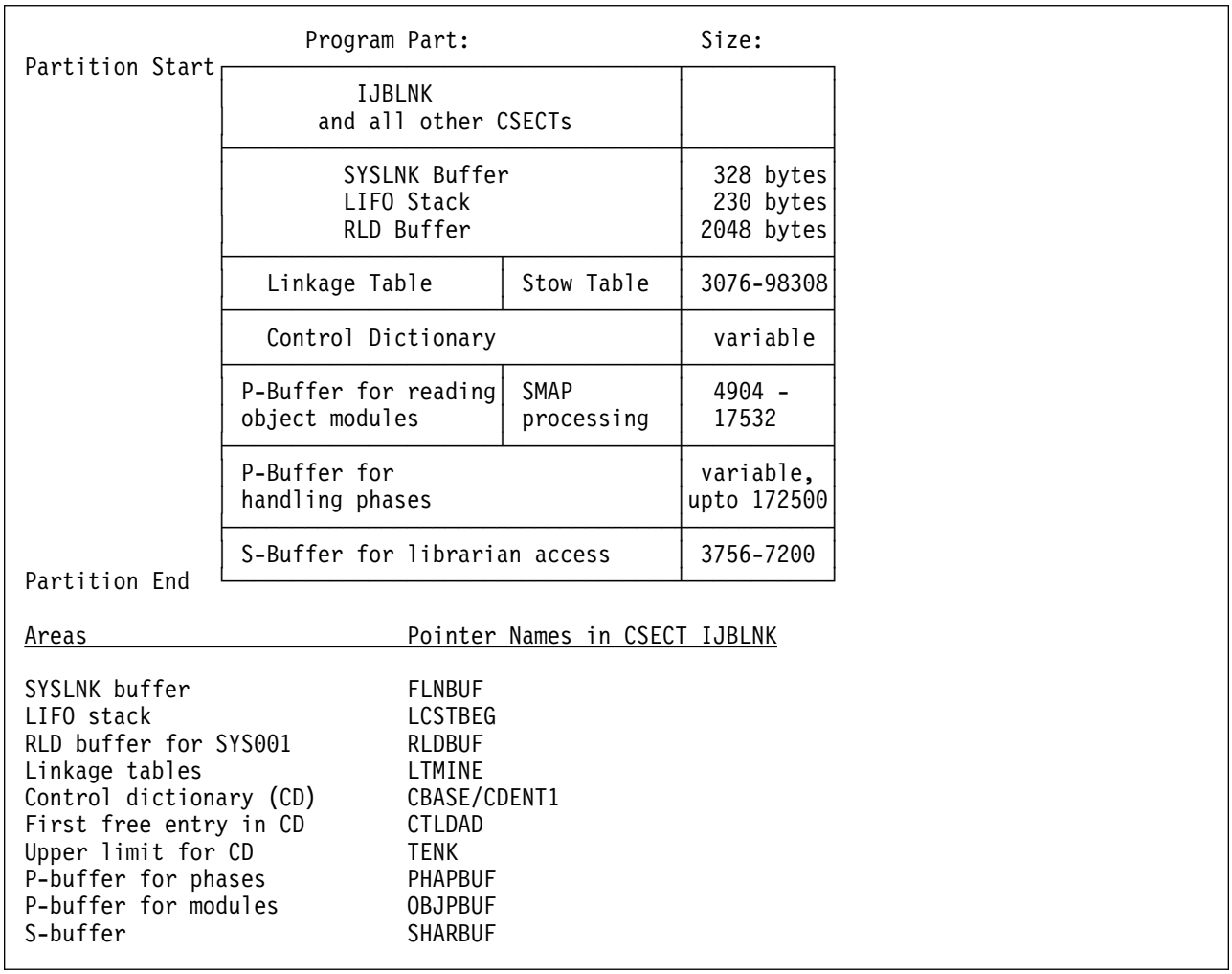


Figure 4. I/O Flow

Program Layout in the Partition

As Figure 5 shows, the program in the partition starts out with the root of the phase, CSECT IJBLNK. Then follow all other CSECTs of which CSECT (and module) INLPQNAM is the last one. It is followed by work buffers and tables, and the CD with a fixed beginning but variable length. The lengths of the P- and S-buffers are calculated from the high end of the partition down to the end of the CD.



Areas	Pointer Names in CSECT IJBLNK
SYSLNK buffer	FLNBUF
LIFO stack	LCSTBEG
RLD buffer for SYS001	RLDBUF
Linkage tables	LTMINE
Control dictionary (CD)	CBASE/CDENT1
First free entry in CD	CTLDAD
Upper limit for CD	TENK
P-buffer for phases	PHAPBUF
P-buffer for modules	OBJPBUF
S-buffer	SHARBUF

Figure 5. Partition Layout

The S-buffer and the P-buffers are formatted by the librarian. The S-buffer is the shared buffer for library management of the CATALOG sublibrary; the P-buffers are used for member I/O.

(For a complete description of the S- and P-buffers see *VSE/AF Diagnosis Reference: Librarian SC33-6330.)*

Figure 6 on page 15 shows how the buffer sizes depend on the partition size.

	Partition Size			
	128 K	256 K	512 K	1024 K
GETVIS Area	49 152	49 152	49 152	49 152
LE code	43 600	43 600	43 600	43 600
SYSLNK Buffer	328	328	328	328
LIFO Stack	230	230	230	230
RLD Buffer	2 048	2 048	2 048	2 048
Linkage Table	3 076	6 148	24 532	98 308
P-Buffer (Obj)	4 904	9 496	17 532	17 532
S-Buffer	3 756	4 904	7 200	7 200
P-Buffer (Phase)	11 300	72 000	172 500	172 500
Control Dictionary	11 300	72 000	204 900	653 000

Figure 6. Approximate Buffer Sizes in Relation to Partition Size

Function-to-CSECT Overview

The program consists of one phase $LNKEDT. This phase consists of three modules, INLPLEIT, INLPQNAM and IJBLE1.

INLPLEIT consists of one CSECT of the same name.

INLPQNAM is a module of the VSE/ESA Librarian. This is a common module which builds qualified names for messages. See *VSE/AF Diagnosis Reference: Librarian* SC33-6330. It is called by module INLPLEIT.

IJBLE1 consists of a number of CSECTS whose names all begin with IJB.

The functions of the CSECTs, in the order of the program structure, are the following:

IJBLNK Contains global constants. Contains common subroutines for linkage editor input handling, Autolink processing, error handling, message/map display and print, and some other tasks.

IJBDTF Contains DTF's for SYSLNK and SYS001.

IJBFIN Contains subroutines for record control (input, sequence), finding a module in a sublibrary, control of phase directory entry information, library interface call.

IJBSRV Contains a subroutine for ACTION option 'ERRLMT'.

IJBMOD Contains common subroutines for AMODE/RMODE processing.

IJBESD Processes ESD records.

IJBOTH Processes TXT, REP, RLD, and END records.

IJBSCN Processes ACTION, INCLUDE, PHASE, MODE, and ENTRY statements.

IJBCTL Pre- and post-processes PHASE and ENTRY statements.

IJBMAP Prints linkage editor map.

IJBRLD Relocates ADCONs in each phase and generates the corresponding RLD records to be used by FETCH/LOAD or by the linkage editor if the phase is re-linked.

IJBCAT Updates the member index of the sublibrary.

IJBINL Serves as entry point to the program. Initializes linkage editor processing. Processes the parameters specified in the PARM field of the EXEC LNKEDT statement.

INLPLEIT Calls librarian routines or VIO routines of the supervisor.

Phase Description for Phase $LNKEDT

Entry Point: IJBINL

Function: Transforms modules of a program into executable phases.***:*** (For details see “Sequence of Operation” below.)

Called By: Job Control or via Call Interface by MSHP or DSNX

Phases Called: $IJBLBR (Librarian)

Data Areas Used:

- I/O areas
- RLD buffer
- Linkage tables
- Control dictionary
- LEITPL (Communication area to librarian)
- Buffers for librarian services
- Stow table (librarian)
- LIFO stack
- Librarian interface control blocks used by INLPLEIT

Messages Caused: See Message-to-CSECT Cross Reference

Messages Issued: By CSECT IJBLNK and for librarian (i.e. "L"-) messages by INLPLEIT.

Input: Statements and object modules

Output: Phases and linkage editor map

Exit Normal: Caller

Exit Error: Cancel

Register Use:

```
R10 Base Register for IJBLNK, IJBCAT
R11 Base Register for IJBFIN, IJBSRV, IJBMOD
R10/11/12 Base Registers for INLPLEIT
R13 Base register for all other CSECTs
```

Sequence of Operation: The sequence of operation is listed in the following by CSECT, in the order of their execution. The most important labels are given on the left side and the operation which they mark on the right side. (See also control flow in Figure 13 on page 33.)

CSECT IJBINL

Called By: Job Control or - via Call Interface - by MSHP or DSNX

```
IJBINL  : - Calculates relocation factor.
OPCATAL : - Opens SYSLST and SYSLNK.
CALCWA  : - Calculates storage addresses for:
            -- I/O areas
            -- Linkage table
            -- Control dictionary
            -- Buffers for librarian services
            -- Stow table
          - Gets partition start address and checks size.
CONCLIB : - Calls INLPLEIT to get access to CATALOG sublibrary.
PRMPROC : - Processes the AMODE,RMODE and MSHP parameters from the
            the PARM field of the EXEC LNKEDT statement.
INLN000 : - Reads from SYSLNK.
ALIGN4X : - Determine first possible phase origin
```

CSECT INLPLEIT

Called via LEITCALL by: IJBINL, LNK, OTH, SCN, CTL, FIN, RLD, CAT

```
CSECT INLPLEIT uses or modifies the following librarian and the following supervisor macros:
 INLMFIND
 INLMGETR
 INLMLAMB
 INLMLDIS
 INLMLRPL
 INLMMCON
 INLMMDIS
 INLMNOTE
 INLMPOIN
 INLMPUTR
 INLMSCON
 INLMSTOW
 LBRACCCB
 LBRACCES

 VIO EXTND
 VIO MOVE
 VIO OPEN

and the following librarian and supervisor control blocks:
 INLCDENT
 INLCLAMB
 INLCLARG
 INLCLPT (field LPTVIOTB)
 INLCLRPL
 INLCMACB
 INLCSACB
 LBRACCDS

 MAPVIORB
```

This is how it operates:

```
OPENC  :  - Builds CATALOG sublibrary control blocks using the macros:
            INLCDENT to generate structure declarations for the
                     control blocks INLCDENT (stow table entry)
                                    INLCLARG (any search argum.)
            LBRACCCB to initialize the control block LBRACCDS
            INLMLAMB to initialize the control block INLCLAMB
            INLMLRPL to initialize the control block INLCLRPL
            LBRACCES to connect to the specified CATALOG sublib.
            INLMSCON to initialize the control block INLCSACB
          - Sets flag IJBARCNA in partition COMREG to establish the
            delayed cancel function in order to keep a consolidated
            library structure for a normal cancel request.
          - Opens virtual I/O control block if option was LINK
            using the macro VIO OPEN which initializes the control
            block MAPVIORB. The MAPVIORB pointer is inserted
            into the table pointed to by field LPTVIOTB in control
            block INLCLPT. This table is in the system GETVIS area.
OPENJ  :  - Builds library control block INLCLRPL using the
            macro INLMLRPL for requests to the OBJ-search chain.
CONNECT:  - Uses routine LEITFIND to establish if a phase of the
            same name as the output phase is in some sublibrary.
            If MSHP is active, the phase is replaced in the same sublibrary.
            If MSHP is not active and the phase is not under MSHP control,
            the phase is replaced in the specified CATALOG sublibrary.
          - Uses the macro INLMMCON to build control block INLCMACB.
WRITE  :  - Writes text information into CATALOG sublibrary
            using macro INLMPUTR.
          - Writes text information into VIO area, if option LINK,
            using macro VIO MOVE. IF necessary, extends the
            virtual storage allocation using macro VIO EXTND.
UPDATE :  - Updates ADCONs in already processed phases
            using macros INLMGETR and INLMPUTR.
          - Updates ADCONs in phase already in VIO area, if option
            LINK, using macro VIO MOVE.
READ   :  - Reads OBJ records from object modules using macro INLMGETR.
FINDM  :  - Finds member from OBJ-search chain using macro INLMFIND.
NOTEM  :  - Notes position in object module using macro INLMNOTE.
POINTM :  - Re-positions in OBJ member using macro INLMPOIN.
STOWML :  - Catalogs phases specified in the stow table
            using macro INLMSTOW.
LEITFIND: - Establishes target sublibrary by finding the phase to
            be replaced, if the linkage editor runs under MSHP.
          - If phase is MSHP controlled, allows replace only
            under MSHP.
            The routine uses the macros INLMFIND, INLMMDIS,
            INLMLDIS, LBRACCES, and INLMSCON.
CHECKCIT: - Uses INLMLAMB to build the control block INLCLAMB
          - Checks if linkage editor is used via Call Interface
            and does the setup for this interface.
LEITWTP : - Call Interface exit for SYSLST.
LEITWTO : - Call Interface exit for SYSLOG.
```

CSECT IJBLNK

Called By: all CSECTs

```
CDENT1  : - Points to control dictionary (CD) in partition
LTMINE  : - Points to Linkage Table in partition
SHARBUF : - Points to buffer for CATALOG sublibrary access
PHAPBUF : - Points to buffer for CATALOG sublibrary access
OBJPBUF : - Points to buffer for OBJ-search chain access
LEITPL  : - Communication area between INLPLEIT and other CSECTs
LCSTBEG : - Header of LIFO stack for nested INCLUDEs
CPHENT  : - Current phase entry
CESDENT : - Current ESD entry
MODE    : - AMODE/RMODE values from MODE control statement for
            current phase
PARMMODE: - AMODE/RMODE values from PARM field of EXEC statement
ESDMODE : - AMODE/RMODE values assigned from ESD data for
            current phase
PHASMODE: - AMODE/RMODE values finally assigned to a phase

ALNKPR  : - Searches the control table for unresolved ER's and
            initializes for Autolink.
RDNEXT  : - Reads "cards" from SYSLNK or SEARCH sublibrary.
RDEXEC  : - Passes control to:
            ---------------> IJBSCN (Control statement)
            ---------------> IJBESD (ESD records)
            ---------------> IJBOTH (TXT,REP,RLD,END records)
```

Subroutines in IJBLNK

As the first CSECT (IJBLNK) of the linkage editor program, IJBLNK contains most of the subroutines used by other linkage editor CSECTs. After processing any of these subroutines, control is returned to the calling routine if not indicated otherwise.

The following list shows name, main entry points, and function of each routine.

```
Subr.   Entry   Function
```

```
LTESID: Input to this routine is an ESID number supplied by the
        language translators.

        LTESID  If CD number is:
                zero: The ESID number was not yet processed.
                      Returns to the address in the link
                      register.
                negative: The ESD record is bypassed. Returns to
                      link register + 4. Addresses of the
                      linkage table entry and the control
                      dictionary number are supplied.
                positive: Returns to register + 8. Relocation
                      factor for SD/PC, control-dictionary
                      -number and -address are supplied.
```

```
Subr.   Entry   Function
──────────────────────────
SRCHCD  SRCHCD  Searches (backward) the CD for a matching label.
        SRPCOD  Continues the search after a matching label has been found.
CNVHEX  CNVHEX  Converts EBCDIC input into hexadecimal output.
PRINT   PRINT   Prints messages and link map on SYSLST.
LOGMSG  LOGMSG  Prints error message on SYSLOG.
PRTLST  PRTLST  Prepares for printing the linkage editor
                diagnostics of input.
SPACE1  SPACE1  Spaces one line on SYSLST. (Not active.)
XTPHNO  XTPHNO  Extracts the phase number from CD entries for
                SD, PC, LD, or LR records.
        XTPHGT  Entry XTPHGT is used if the entry is known to be an SD or PC.
ABTERR  ABTERR  Gives control to ---------------------> IJBRLD
                for handling of an abnormal termination error.
CDSIZE  CDSIZE  Checks for CD overflow.
ALNKPR  ALNKPR  Initializes for the scanning of the sublibrary
                member index for Autolink. Extracts
                unresolved ERs from the CD in collating sequence
                and includes the corresponding modules.
RDNEXT  RDNEXT  Reads the input stream.
        RDEXEC  Determines type of statement and appropriate CSECT.
        EXLOAD  Branches to CSECT determined.
CANCL   CANCL   Cancel routine.
ERROR   ERROR   Sets up to print non-termination error messages.
        WARNING Sets up to print warning messages. (return code 2)
        INFORMTN Sets up to print informational messages. (return code 0)
                If the calling routine sets the RETRN bit in ERRSW,
                returns to caller. If RETRN is off, returns to
                RDNEXT or ALNKPR if error during Autolink.
NOTCTL  NOTCTL  Converts input statements (X'02' in first byte) to
                print format.
```

CSECT IJBSRV

Called By: IJBSCN, (RLD)

Subroutines in IJBSRV

CSECT IJBSRV contains following subroutines:

```
Subr.     Entry     Function
──────────────────────────────
HNDLCNT   HNDLCNT   Process the parameter of the ACTION option 'ERRLMT'.
PRCD      PRCD      Print the CD in readable form within the linkage editor map (for test purpose only)
```

CSECT IJBMOD

Called By: IJBINL, SCN, CTL, MAP, CAT

Subroutines in IJBMOD

CSECT IJBMOD contains common subroutines for AMODE/RMODE processing used by other linkage editor CSECTs. After processing any of these subroutines, control is returned to the calling routine.

The following list shows name, main entry points, and function of each routine.

Subr.	Entry	Function
CHKMODE	CHKMODE	Set default AMODE or RMODE value and check for conflicting AMODE/RMODE specification on the MODE statement or on the PARM field of the EXEC LNKEDT statement.

DETESDMD	DETESDMD	Determine AMODE/RMODE values for the entry point of a phase from the ESD SD entry of the CSECT containing the entry point and from the cumulative AMODE/RMODE of the phase. (accumulated on a most restrictive basis from all CSECTs contributing to the phase.)
		
		Processing rules for AMODE/RMODE data from the ESD data:
		- If the external symbol of the entry point is marked with any of the allowed AMODE values and an RMODE of 24, the entry point of the phase is assigned the same AMODE attribute as its associated external symbol.
		- The AMODE 24/RMODE ANY combination is invalid. (This combination is already handled in CSECT IJBESD and reset to 24/24.)
		- If the external symbol of the entry point is marked with AMODE 31/RMODE ANY, the entry point of the phase is assigned an AMODE 31 and the RMODE will be that of the phase, which is the RMODE accumulated on the "most restrictive" basis.
		- If the external symbol of the entry point is marked with AMODE ANY/RMODE ANY, the entry point of the phase is assigned an AMODE and RMODE according to the following hierarchy:
		  - If the phase contains one or more CSECTs marked AMODE 24, the linkage editor assigns an AMODE of 24 to the entry point of the phase.
		  - If the phase has an RMODE of 24 and it contains no CSECTs marked AMODE 24, the linkage editor assigns an AMODE of ANY to the entry point of the phase.
		  - If the RMODE of the phase is ANY, the linkage editor assigns an AMODE of 31 to the entry point of the phase.

GETMODE	GETMODE	Retrieve AMODE/RMODE values assigned to a phase from the PHASE C/D entry. (Returns in ESDMODE the AMODE/RMODE values assigned from the ESD data to the phase and in PHASMODE the AMODE/RMODE values finally assigned to the phase.)

CSECT IJBOTH

Called By: IJBLNK

```
IJBOTH : - Initializes IJBOTH.
         - Branches to the right processor for the statement at hand:
           ---------->TXTPRC
           ---------->REPROC
           ---------->RLDPRC
           ---------->ENDPRC
TXTPRC : TXT processor
         - Puts text into CATALOG sublibrary format.
         - Calls INLPLEIT to write text. -------> RDNEXT (IJBLNK)
REPPRC : REP processor
         - Modifies REP to text.
         - Processes in TXTPRC. ----------------> RDNEXT (IJBLNK)
RLDPRC : RLD processor
         - Converts R and P pointer information to CD
           number information for RLD pass 2.
         - Stores RLD records in buffer (writes to IJSYS01
           if necessary.) ----------------------> RDNEXT (IJBLNK)
ENDPRC : END processor
         - Updates input control mechanism (LIFO stack PERIDA).
         - Supplies transfer address.
         - Identifies unassigned LD/LR in control
           dictionary for this module.
         - Clears the linkage table.
         - Accepts CSECT length if necessary. (Some
           language translators supply CSECT length
           in end record.)
         - If not end of a multiple OBJ library member
           goes to ------------------> RDNEXT (IJBLNK)
         - If Autolink necessary
           goes to ------------------> ALNKPR (IJBLNK)
           Else goes to -------------> RDNEXT (IJBLNK)
```

CSECT IJBFIN

Called By: all CSECTs

```
READIN  : - Controls record I/O.
LNKPOINT: - Points to next record.
LNKNOT  : - Notes record.
LCFIND  : - Finds module.
IJBLETR : - Handles messages.
LEITCALL: - Calls INLPLEIT for librarian services.
LCLOSE  : - Disconnects all connected libraries and sublibraries.
```

CSECT IJBESD

Called By: IJBLNK

```
IJBESD : - Controls input for updating LIFO-stack PERIDA.
         - Builds dummy PHASE statements if none supplied.
ESDRET : - Checks validity of type code for ESD record.
         - Exits if all ESD records are processed.
         - If end-of-record or SYM record found
           goes to ------------------> RDNEXT (IJBLNK)
           Else preliminary processing of SD/PC/LD/ER/CM/PR records.
         - Compares label on input ESD with CD.
         - Completes SD/PC processing by ensuring that
           name field on ESD record is blank.
         - Completes processing:
ELBCM  :   - CM
ELBPR  :   - PR
ELBSD  :   - SD/PC
ELBLD  :   - LD
ELBER  :   - ER
         - If no dictionary update ---------> ESDRET
         - Else:
           -- Posts ESD record in CD.
           -- Moves CD number to linkage table.
           -- Checks for CD and linkage table overflow.
           -- Goes to ----------------------> ESDRET
```

In detail, ESD processing takes the following steps:

Pre-Processing

1. For each ESD record produced by a language translator, an input CD entry is built at a fixed location in storage. In some cases, this input CD entry is moved to the CD during processing.
2. The input ESD type field is validated.
   - If it is a weak external, the ESD type field in the input CD entry is set to ER and the NOAUTOL and WXTRN bits in CSWITCH are turned on.
   - If it is invalid, an error condition exists, the whole ESD record is ignored, and the next ESD record is processed.
3. Further preprocessing depends on the ESD type:
   - For LD input: An LD record has a pointer to the linkage table where the CD number is checked to see if the LD record has already been processed.
   - For ER input: If NOAUTO was specified, the NOAUTOL bit in CSWITCH is set on.
   - For SD or PC input: Two conditions must be fulfilled:
     a. The assembled origin must be aligned on a double-word boundary.
     b. The PC must be unnamed.

     The relocation factor is calculated by subtracting the assembled origin from the storage address (NXPHRG).

For a normal INCLUDE, pre-processing is finished at this stage.

For a submodular INCLUDE, the name list of included CSECTs is scanned for a name identical to the name of the input CD entry. If the names match, pre-processing is finished. If not, the ESD type field in

the input CD entry is changed to ER and a switch is set to ensure that the CD number in the linkage table is given a negative value.

Processing

1. The CSECT scans the CD for an entry with the same name as the input ESD.

   This scan starts at the end of the CD and proceeds towards the beginning until either a match occurs or the beginning of the CD is reached. If a match occurs, the CD entry is called a duplicate.

   The scan continues if the duplicate is a phase entry.

2. If no duplicate is found, the input CD entry is added to the end of the CD. If the input ESD is an SD, PC, CM, PR, or ER, an entry is made in the linkage table.

3. If a duplicate is found, the action taken by the ESD processor depends on the relationship between input and duplicate. Figure 7 shows all possible actions and their abbreviations A1 to Err-46.

```
Action   Meaning
 A1:     Ignores input CD entry.
 A2:     Adds input CD entry to the end of CD.
 A3:     Replaces duplicate with the input CD entry.
 A4*:    Adds the linkage table entry pointing to the last entry
         added to the CD.
 A5*:    Adds the linkage table entry pointing to the duplicate.
 A6:     Changes duplicate LD to LR.
 A7:     Continues scan of CD.
 A8:     Saves length of longest CM or PR in CD.
 A9:     Gives CD number in linkage table a negative value.
 A10:    Changes input LD to LR.
 A11:    Sets 'Possible Duplicate Entry' switch.
 A12:    Set LATELINK indicator for possible link during post processing of SD/PC.
 Err-43: Prints error message '2143I' and goes to RDNEXT.
 Err-46: Prints error message '2146I' and goes to RDNEXT.
 Err-48: Prints error message '2148I'.
-----------------------------------------
*If a submodular INCLUDE was used and the name list of included
 SDs does not contain an SD, the CD number in the linkage table is
 given a negative value.
```

Figure 7. ESD Processing Actions

To find which action is taken while processing input CM, PR, ER, SD, or LD records, use Figure 8 on page 26 to Figure 12 on page 27. The upper part of these figures shows the various conditions which exist (Y), do not exist (N), or can be ignored (-), while the lower part indicates the actions taken (X).

Duplicate = SD	Y Y N N N N
= PC	N N Y N N N
= CM	N N N Y N N
= LD/LR	N N N N N Y
= ER	N N N N Y N
= PR	N N Y N N N
SD Length < CM Length	N Y - - - -
A3	- - - - X -
A4	- - - - - -
A5	X X - X X -
A7	- - X - - -
A8	- - - X - -
Err-46	- - - - - X
Err-48	X X - - - -

Figure 8. Decision Table if Input is a CM Record

Duplicate = SD	Y N N N N N
= PC	N Y N N N N
= CM	N N Y N N N
= LD/LR	N N N Y N N
= ER	N N N N Y N
= PR	N N N N N Y
A5	- - - - - X
A7	X X X X X -
A8	- - - - - X

Figure 9. Decision Table if Input is a PR Record

Duplicate = SD, LD, or LR	Y Y Y Y Y Y Y Y Y N N N N Y N
= LD	- - N N N N Y Y Y Y N N N N - N
= CM	N N N N N N N N N N Y N N N N N
= ER	N N N N N N N N N N N Y Y Y N N
= PR	N N N N N N N N N N N N N N N Y
Duplicate unassigned *	N N N N N N N N N N - - - - Y -
Name = 'IJ..' or 'IBM..'	Y Y Y Y Y N Y Y Y N - - - - - - -
Name = 'IBM..'	N Y - - N - - - N - - - - - - - -
NOAUTO for input	N N Y N N - Y N N - - - Y N - -
Duplicate in current phase	N N - Y N - - Y N - - N Y Y - -
Duplicate in ROOT phase	N - - - Y - - - Y - - - - - - -
A2	X X - - - - - - - - - - - - - -
A3	- - - - - - - - - - - X - X - -
A4	X X - - - - - - - - - - - - - -
A5	- - X X X X X X X X X X X X - -
A6	- - - - - - X X X X - - - - - -
A7	- - - - - - - - - - - - - - X X
A12	- - - - - - - - - - - - - - X -

* SD is to be considered assigned

Weak externals are processed like ERs for which NOAUTO is requested.

Figure 10. Decision Table if Input is an ER Record

Duplicate = SD	N N N N N Y Y Y Y N N N N N N N N N N
= CM	Y Y N N N N N N N N N N N N N N N N N
= LD or LR	N N N N N N N N N Y Y Y Y Y Y Y Y Y N
= ER	N N Y Y Y N N N N N N N N N N N N N N
= PR	N N N N N N N N N N N N N N N N N Y
Duplicate unassigned	- - - - - - - - - N N N Y Y Y Y Y Y -
Input and dupl. ESIDs agree	- - - - - - - - - - - - - N N N Y Y Y -
ASSORGs agree	- - - - - - - - - - - - - - - - N N N Y -
Duplicate in current phase	- - - Y N Y N N N - Y N - Y N - Y N - -
Name = 'IBM..'	- - N Y Y - Y N N N Y Y N Y Y N N Y Y - -
Duplicate in ROOT phase	- - - - - - - - N Y - - - - - - - - - -
SD length < CM length	N Y - - - - - - - - - - - - - - - - - -
A2	- - - - X - X X - - - X - - X - - X - -
A3	X X X X - - - - - - - - - - - - - - X -
A4	- - - - X - X X - - - X - - X - - X - -
A5	X X X X - X - - X - - - - - - - - - X -
A7	- - - - - - - - - - - - - - - - - - - X
A9	- - - - - X - - X - - - - - - - - - - -
Err-43	- - - - - - - - - X X - X X - X X - - -
Err-48	- X - - - - - - - - - - - - - - - - - -

Figure 11. Decision Table if Input is an SD Record

Duplicate = SD, LD, or LR	N N N N Y Y Y Y Y Y Y Y Y Y Y Y Y Y N
= CM	Y N N N N N N N N N N N N N N N N N N
= LD or LR	N N N N - - - - - Y Y Y Y Y Y Y N N N N N
= ER	N Y Y Y N N N N N N N N N N N N N N N
= PR	N N N N N N N N N N N N N N N N N N Y
Duplicate unassigned **	- - - - N N N N N N N N N N Y - - - - -
Input unassigned	- - - - - - - - Y Y N N N N N - N N N N -
Input points to duplicate	- - - - - - - - - - - - - - - - Y N N N -
Input and duplicate point to the same entry	- - - - - - - - - Y N N N N N - - - - - -
Names of C/D entries agree	- - - - - - - - - - N N N Y Y - - - - - -
Name of input and duplicate = 'IBM...'	- N Y Y N Y Y - - - N Y Y - - - - N Y Y -
Duplicate in current phase	- - Y N - N Y N Y - - N Y N Y - - - N Y -
Input and duplicate ASSORGs agree	- - - - N N N Y Y Y Y Y Y Y Y - Y Y Y Y -
A1	- - - - - - - - - X X - - - - X - X - - - -
A2	- - - X - X - X - - - X - X - - - - X - -
A3*	- X X - - - - - - - - - - - - - X*- - - - -
A7	- X
A10	- X X X - - - - - - - - - - - - - - - - -
A11	- - - - - - - - - X - - - - - X X - - - - -
Err-43	- - - - X - X - - - X - X - - - - X - X -
Err-46	X -

* Action A3 is performed retaining the ESD type of the duplicate
** SD is to be considered assigned

Figure 12. Decision Table if Input is an LD Record

Post-Processing

1. For ER, LD/LR, CM, or PR input, the next ESD record is selected for processing.
2. For SD or PC input
    a. If it is not a zero-length PC the cumulative AMODE and RMODE values for the current phase are updated. These values are accumulated on a most restrictive basis. (That is, if for example the RMODEs of the CSECTs are not the same the more restrictive value, RMODE 24, is chosen.)
    b. The CD is scanned for unassigned LDs or LRs pointing to the input record.
    c. The CD entries found during the previous scan are updated. This is done by storing in the CD entry the CD number found in the linkage table entry that corresponds to the input item. For unassigned LD's with LATELINK indicator on a (forward) subscan is started to find the ER for which the indicator had been set. If found the linkage is completed by replacing the ER with the LD entry.
    d. The storage address (NXPHRG) is updated by adding the length of the CSECT.

If the length of the CSECT is provided in the END statement, CSECT IJBOTH performs action d.

CSECT IJBSCN

Called By: IJBLNK

```
IJBSCN : - Controls input for updating PERIDA.
         - Finds operation field and checks validity.
         - Branches to the processor for this type of statement:
           ------------> INCCRD
           ------------> ACTISEC
           ------------> MODCRD
           ------------> PHCRD
           ------------> ENTCRD

INCCRD : INCLUDE statement processor
         - Checks validity of operands.
         - If no operand, -----------------> RDNEXT (IJBLNK)
         - Else locates module to be included and autolinks.
         - Goes to ------------------------> RDNEXT (IJBLNK)

ACTISEC: ACTION statement processor
         - Handles initial condition (after IJBINL execution)
           * Processes all consecutive ACTION statements if first statements
           * Sets default MAP/NOMAP option if no ACTION statement
           * Terminates job step, if ENTRY statement found
         - Processes single (not initial) ACTION statements
         - Calls HNDLCNT (IJBSRV) subroutine for ERRLMT option
         - Prints 'ACTION TAKEN' line for each logical ACTION statement
         - Goes to ------------------------> RDNEXT (IJBLNK)
MODCRD : MODE statement processor
         - Checks validity of operands.
         - Goes to ------------------------> RDNEXT (IJBLNK)

PHCRD  : PHASE statement processor
         - Checks validity of operands.
         - If Autolink required -----------> ALNKPR (IJBLNK)
         - Else goes to -------------------> IJBCTL

ENTCRD : ENTRY statement processor
         - Provides exit from ENTRY or PHASE statement processors.
         - Saves transfer address if in ENTRY statement.
         - If Autolink --------------------> ALNKPR (IJBLNK)
         - Else goes to -------------------> IJBCTL
```

CSECT IJBCTL

Called By: IJBSCN

```
IJBCTL : - If first phase -------------------> PHSPRC
         - Else ----------------------------> WRTRFR

WRTRFR : PHASE post-processor
         - Reserves space for relocation information.
         - Calls DETESDMD(IJBMOD) to determine AMODE/RMODE
           values assigned from the ESD data to the phase.
         - Inserts AMODE/RMODE values from the MODE control
           statement or from the PARM field of the EXEC LNKEDT
           statement and the AMODE/RMODE values
           assigned from the ESD data to the phase into
           the PHASE C/D entry.
         - If Autolink ----------------------> ALNKPR (IJBLNK)
         - Else
           -- If PHASE statement ------------> PHSPRC
           -- If ENTRY statement ------------> IJBMAP

PHSPRC : PHASE pre-processor
         - Determines optional operands specified.
         - Builds current phase CD entry.
         - Gets information to processing phase.
         - Determines if relocation possible.
         - Goes to ---------------------------> RDNEXT (IJBLNK)
```

CSECT IJBMAP

Called By: IJBCTL

```
IJBMAP : - Accumulates list of common areas.
         - Accumulates list of pseudo registers.
         - Displaces phase load address by cumulative length
           of commons.
         - Calculates load origin for transfer address.
         - Sorts CSECTs by load address.
         - If option SMAP, sorts CSECT names alphabetically
           and produces list.
         - If option MAP prints map.
         - Exits depending on errors and option CANCEL.
           Goes to -----------------> EOJ or IJBRLD
```

CSECT IJBRLD

Called By: IJBMAP

```
IJBRLD :  - If no more RLDs --------> TSTUNR (IJBRLD)
          Pass2 P-pointer processor
           -- Reads RLDs.
           -- Gets relocation factor for P-pointer.
           -- Calls INLPLEIT to move the correspondent CD entry
              to the current phase entry, if the P-pointer is
              outside the current phase.
           -- Control flow in IJBRLD depends on conditions found.
RLDOR  :  Pass2 R-pointer processor
           -- Gets relocation factor for R-pointer, unless R-pointer is zero (CXD).
           -- Adds the assembled origin (i.e. the address of the SD,
              PC or CM that defines the ER) if R-pointer is an ER.
           -- If a constant must not be processed ---> IJBRLD
RLDCON : Pass2 RLD constant processor
         - Adjusts constant portion of RLD record by
           relocation factor.
         - Loads PR displacement to Q-type address constant
         - Loads PR cumulative length to CXD field
         - Calls INLPLEIT to update ADCONs in phase
           text in CATALOG sublibrary ----------> IJBRLD
TSTUNR   - If RLD PASS 3 ----------------------> ENDVER (IJBRLD)
         - If MAP option -----------------------> TSTCNT
TSTCNT : MAP routine
         - Lists unresolved ADCONs.
         - Lists ADCONS outside of limits of current phase.
         - If phase not relocatable ------------> ENDVER (IJBRLD)
         - Initializes Pass3 RLD processing.
           -- Modifies Pass2 RLD processor to insert relocation
              information for relocatable phases in space
              reserved by WRTRFR (IJBCTL).
           -- Positions at start of RLD information.
         - Adds RLD information to phase text.
WRLST  : Block phase header
         - Calls INLPLEIT to write last buffer to library.
         - Closes SYSLNK and IJSYS01 and --------> IJBCAT
```

CSECT IJBCAT

Called By: IJBRLD

```
IJBCAT : - Starts stow table.
         - Initializes CD search.
SCANCD : - Locates next phase entry in CD.
         - If end of stow table:
           - Calls INLPLEIT to update sublibrary member index or,
             in case of a previous severe error, deletes the
             phases already written into the CATALOG sublibrary.
           - Starts new stow table.
         - Modifies load and transfer address of phase in process.
         - Calls GETMODE(IJBMOD) to retrieve AMODE/RMODE
           values assigned to the phase from the PHASE C/D entry
           and inserts these values into the stow table entry.
         - Adds entry to stow table.
         - If more entries in CD, -------------------------> SCANCD
         - Calls INLPLEIT to disconnect any CATALOG sublibrary
           or SEARCH sublibrary connected.
         - Sets return code into MSHPRET:
             0 if successful link
             2 if warning issued, but phases are cataloged
             4 if warning or error issued, but phases are cataloged
             8 if phase not replaced in library (CALL I/F only)
            16 if severe error and phases are not cataloged
         - Returns to caller (MSHEOJ)
```

Basic Program Flow

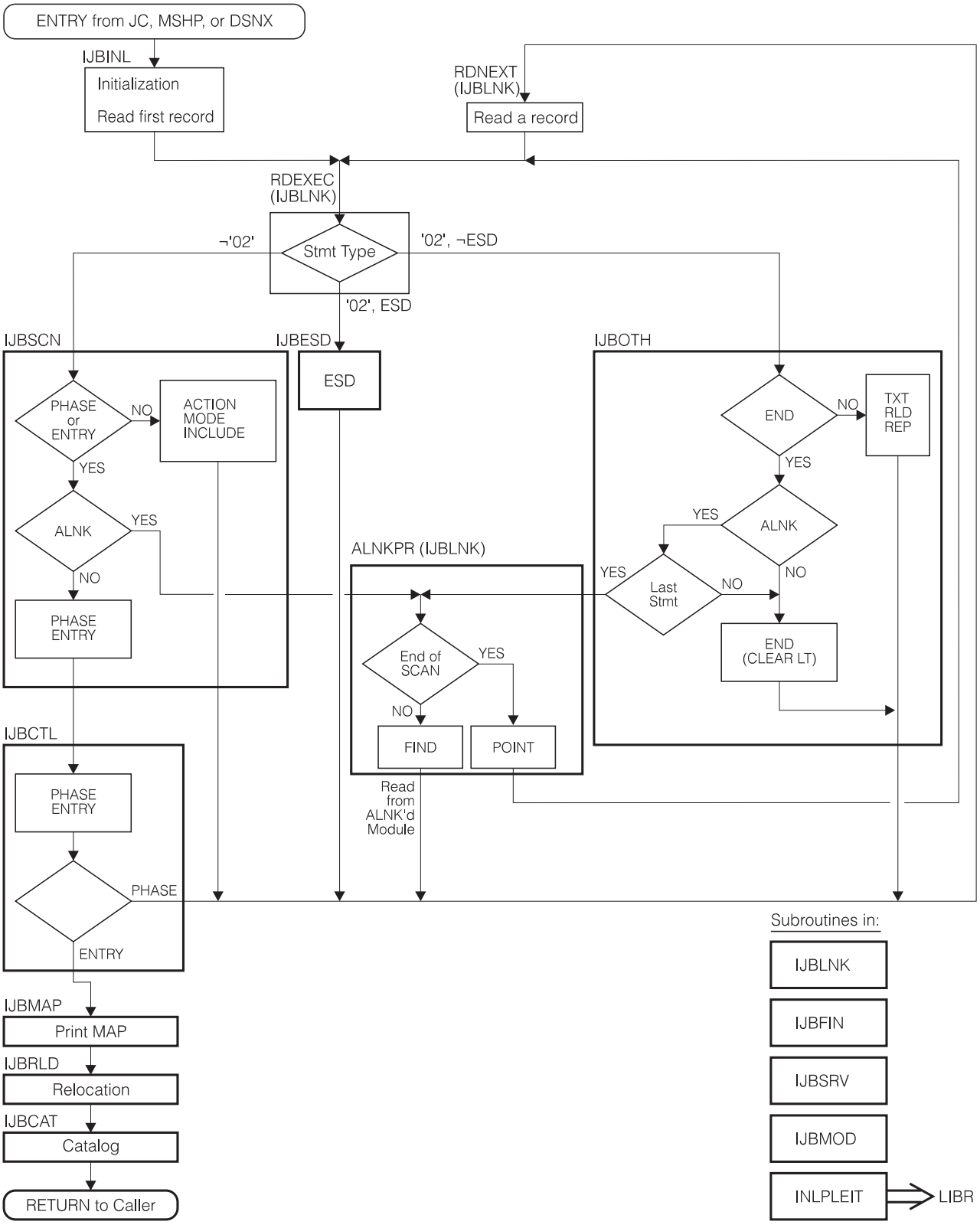


Figure 13. Control Flow

Data Areas

Library Record Formats

The linkage editor input records coming from the library have the format of library members of type OBJ. The records for output have the format of library members of type PHASE.

Input Record Formats

The input records for the linkage editor are in card image format:

The ESD Record

```
Card Columns
 1      Hex '02'; identifies an object module record.
 2 -  4 'ESD'
11 - 12 Number of bytes of information contained in this record
15 - 16 ESID number of the first SD, PC, CM, PR, or ER on this record.
17 - 72 Variable information:
  8 positions - Name
  1 position  - Type code hex 00, 01, 02, 04, 05, 06, or 0A,
                to indicate SD, LD, ER, PC, CM, PR, or WX.
  3 positions - Assembled origin if SD, PC, or LD.
  1 position  - AMODE/RMODE data if SD or PC.
                xxxx x...  not used
                .... .R..  RMODE data
                             0 = 24
                             1 = ANY
                .... ..AA  AMODE data
                             00,01 = 24
                                10 = 31
                                11 = ANY
              - Alignment factor if PR.
                       07 - doubleword alignment
                       03 - fullword alignment
                       01 - halfword alignment
                       00 - byte alignment
              - Blank if LD, CM, ER, or WX.
  3 positions - Length of control section if SD, CM, or PC.
              - ESID number of SD containing the label if LD.
              - Length of pseudo register if PR.

73 - 80 May be used by the programmer for identification.
```

The TXT Record

```
Card Columns

 1      Hex '02'; identifies an object module record.
 2 -  4 'TXT'
 6 -  8 Assembled origin (address of first byte to be loaded from this record)
11 - 12 Number of bytes of text to be loaded
15 - 16 ESID number of the CSECT (SD or PC) containing the text
17 - 72 Up to 56 bytes of text -- data or instructions to be loaded
73 - 80 May be used for program identification.
```

The RLD Record

```
Card Columns

 1      Hex '02'; identifies an object module record.
 2 -  4 'RLD'
11 - 12 Number of bytes of information contained in this record.
17 - 72 Variable information:
  2 positions -  R-pointer:
                 ESID of (target) CSECT (SD entry), if CSECT or entry label
                 within CSECT is referenced,
                 ESID of (target) COM (CM entry), if common control section
                 is referenced,
                 ESID of EXTRN (ER entry), if an external label is referenced,
                 X'0000', if CXD (cumulative pseudo register length) is defined.
  2 positions -  P-pointer:
                 ESID of CSECT (SD entry), in which address constant is located.
  1 position  -  flag bits indicating type of constant:
       0 - 1  ignored
       2 - 3  type:
              00 - a non-branch type load constant (A-con)
              01 - a branch type load constant (V-con)
              10 - a pseudo register (Q-con)
              11 - a cumulative pseudo register length (CXD)
       4 - 5  00 - load constant length = 1 byte
              01 - load constant length = 2 bytes
              10 - load constant length = 3 bytes
              11 - load constant length = 4 bytes
       6       0 - relocation factor is to be added
               1 - relocation factor is to be substracted
       7       0 - Next load constant has different R- and P-pointers.
                   R and P must be present.
               1 - Next load constant has the same R- and P-pointers.
                   Therefore they are both omitted.

  3 positions  -  assembled address of address constant.

73 - 80 May be used for program identification.
```

The END Record

Card Columns

```
 1      Hex '02'; identifies an object module record.
 2 -  4 'END'
 6 -  8 Assembled origin of the label supplied to the assembler in the
        END record (zero, if the label is an external symbol),optional.
15 - 16 ESID number of the CSECT or external reference to which this
        END record refers. (Only if 6-8 present)
17 - 22 Symbolic label supplied to the linkage editor, if this label was not
        defined within the assembly
29 - 32 CSECT length (if not specified in last SD or PC)
73 - 80 Not used
```

The REP Record

Card Columns

```
 1      Hex '02'; identifies an object module record.
 2 -  4 'REP'
 5 -  6 Blank
 7 - 12 Assembled hex address of the first byte to be replaced.
        Right justified with leading zeros.
        (Note that there is no check to determine if the
         the assembled address is actually within this CSECT)
13      Blank
14 - 16 ESID hex number of the CSECT (SD) containing the text.
        Right justified with leading zeros.
17 - 70 From 1 to 11 4-digit hex fields separated by commas. Each
        field takes two bytes. A blank shows end of information.
71 - 72 Blank
73 - 80 May be used for program identification
```

Object Member Record Format

In a sublibrary, an object module has only records of 80 bytes. These records contain either one and only one linkage editor statement or one of the five possible types of module record types (ESD, TXT, RLD, REP, or END).

When job control or a language translator writes them on SYSLNK it adds to them two bytes of control information, a blocking factor and the record length of 80 bytes, and possibly blocks them to a block length of 322 bytes.

Figure 14 shows the two possible record formats on SYSLNK.

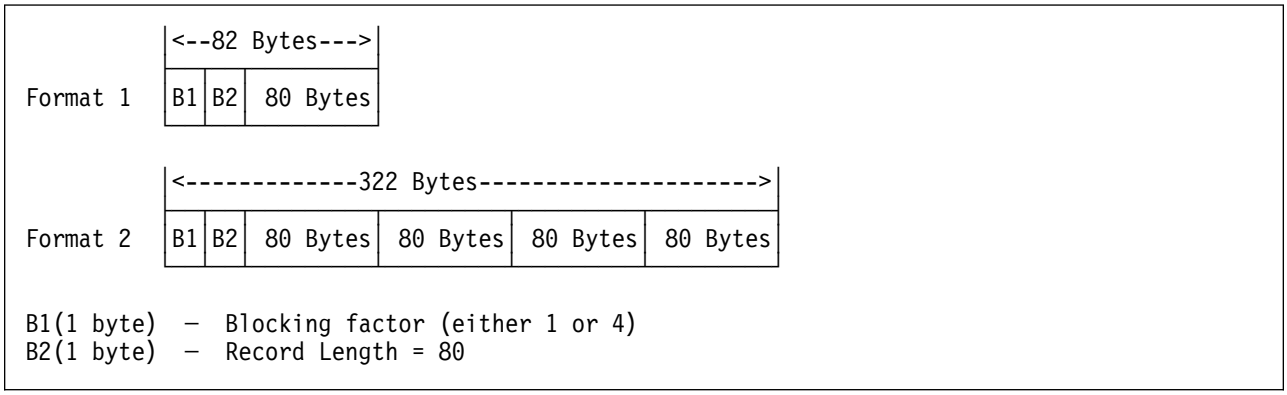


Figure 14. OBJ-Member Record Formats on SYSLNK

Figure 15 shows the format of a complete linkage editor input module on SYSLNK. The linkage editor statements may already be stored with the module in the sublibrary.

4	80	INCLUDE	INCLUDE	PHASE	ESD
4	80	ESD	ESD	TXT	TXT
4	80	TXT	TXT	TXT	TXT
4	80	TXT	TXT	TXT	REP
4	80	REP	RLD	RLD	RLD
4	80	RLD	END		

<---> 322 Bytes

Figure 15. Example of a Module on SYSLNK

SYSLNK Control Interval Format

Figure 16 shows a VSAM or FBA control interval (CI) with SYSLNK input records for the linkage editor.

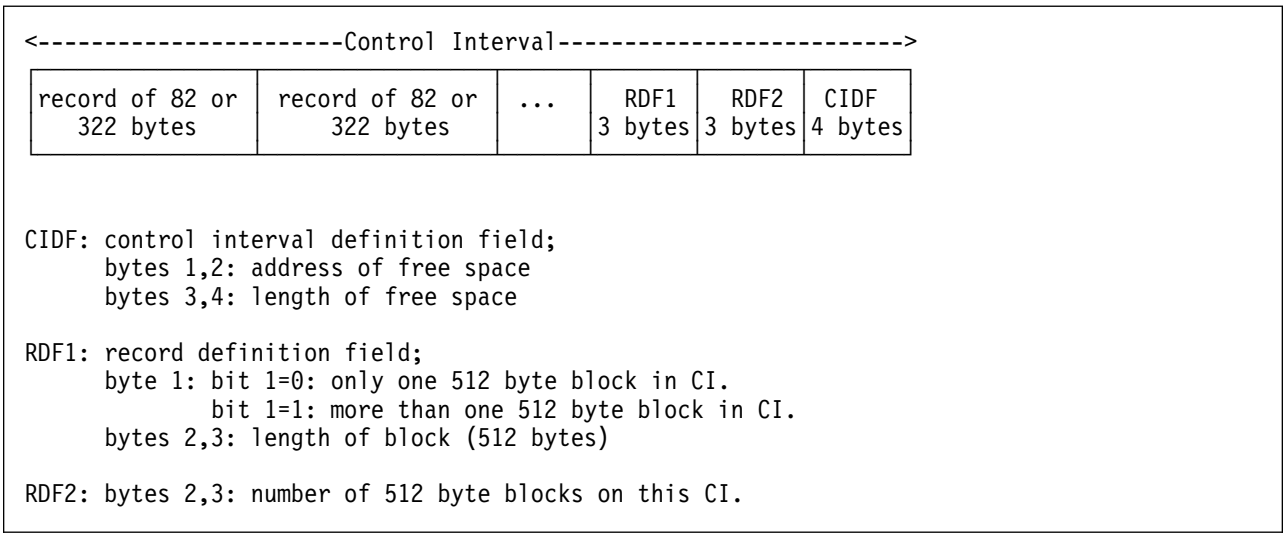


Figure 16. VSAM or FBA Format of SYSLNK Records

Output Record Formats

A phase is stored into the CATALOG sublibrary as a member of type PHASE. It contains executable code and, at the end of the phase, some RLD information, if the phase is relocatable.

RLD Block Format

The RLD items stored by CSECT IJBOTH in the RLD buffer are blocked to a length of 2020 bytes. If the RLD buffer is full, its content is written on workfile IJSYS01.

CSECT IJBRLD places the RLD items into the CATALOG sublibrary. The layout of an RLD block is shown in Figure 17.

Offset	Length	Content
0	4	Number of RLD items in one block
4	2016	RLD items

Figure 17. Layout of an RLD Block

RLD Item Format in Relocatable Phases

The four byte entries listed at the end of a relocatable phase has the following format:

```
Bit    Function
0..2   zero
3,4    length code of ADCON
5,6    zero
7      zero, if relocation factor positive
       one, if relocation factor negative
8..31  ADCON address as generated for partition
```

Linkage Editor Data Areas

Control Dictionary (CD)

Function: The CD holds information on phases and modules for the address adjustment of the linkage editor.

Name: Control Dictionary

Label or Identifier: None

Location: Partition

Initialized By: IJBINL

Pointed to By: CBASE, CDENT1

Used By: All CSECTs except IJBINL and INLPLEIT

Format of CD Entries: The CD has entries for ESD information, for modules, and for phases. CD space is also used to save directory entry information of a phase (takes 4 entries). The entries for phases and for ESD records are first built one by one in CSECT IJBLNK or stored temporarily in the current CD entries. During processing, the references to CD information are sometimes to fields of such “current entries” in the CSECTs instead of the CD entry field. In IJBLNK, a current phase entry is stored under the label CPHENT and a current ESD entry is stored under the label CESDENT. The layout of CD entries is the following:

```
OFFSETS  TYPE        LENGTH  NAME      DESCRIPTION
 0  (0)  STRUCTURE     0    CDENTRY
 0  (0)  SIGNED        4

ENTRY FOR SD,PC,CM,LD,LR,ER,WX,PR RECORDS
 0  (0)  CHARACTER     8    NAMED     NAME OF ESD ITEM
 8  (8)  HEX           1    ESDTYPD   ESD TYPE
 9  (9)  HEX           3    ASSORGD   ASSEMBLED ORIGIN OR
 9  (9)  HEX           3    LNGTHD    LENGTH OF CM
12  (C)  SIGNED        4    RELFACD   RELOCATION FACTOR
12  (C)  HEX           4    PRDISPLD  DISPLACEMENT OF PR
12  (C)  SIGNED        2
14  (E)  SIGNED        2    CSNUMD    OR C/D # OR ESID #
                                      OF CSECT FOR LD/LR
14  (E)  SIGNED        2    PHNUMED   OR PHASE # FOR ER/WX
16 (10)  SIGNED        2    PHNUMD    PHASE # FOR SD/PC
18 (12)  HEX           1    SWITCHD   DIVERSE SWITCHES
19 (13)  HEX           1    ALGNFACD  ALIGNMENT FACTOR FOR PR OR
19 (13)  HEX           1    SDMODED   AMODE/RMODE of CSECT for SD/PC
20 (14)  SIGNED        2    SDXREFD   XREF TO INPUT MODULE
22 (16)  SIGNED        2              RESERVED
24 (18)  HEX           1              RESERVED
25 (19)  HEX           3    SDLNGTHD  LENGTH OF CSECT
28 (1C)  HEX          20              RESERVED

   ENTRY FOR PHASE
 0  (0)  CHARACTER     8    PHNAMED   PHASE NAME
 8  (8)  HEX           1    ESDTYPED  ESD TYPE
 9  (9)  HEX           5              RESERVED
14  (E)  SIGNED        2    RLDITEMD  # OF RLD ITEMS
16 (10)  SIGNED        4    TXTLEND   TEXT LENGTH IN BYTES
20 (14)  SIGNED        4    RLDLRBD   LRBA FOR RLD ITEMS
24 (18)  SIGNED        4    ORPHRGD   PHASE ORIGIN
28 (1C)  SIGNED        4    NXPHRGD   NEXT PHASE ORIGIN
32 (20)  SIGNED        4    TRFRADD   TRANSFER ADDRESS
36 (24)  SIGNED        4    LINKSTRD  START OF PARTIT.
40 (28)  HEX           1    PHMODED   AMODE/RMODE VALUES FROM MODE CONTROL STATEMENT
                                      OR PARM FIELD OF EXEC STATEMENT
41 (29)  HEX           1    PHESDMD   AMODE/RMODE VALUES ASSIGNED
                                      FROM ESD DATA TO THE PHASE
42 (2A)  CHARACTER     1    PHTYPED   PHASE TYPE
43 (2B)  CHARACTER     1    PHTYP2D   ADDITIONAL FLAGS
  1... ....                 PHTYP2R   ROOT PHASE
  .1.. ....                 PHTYP2T   FIRST PHASE PROCESSED
  ..1. ....                 PHDUMMY   PHASE ENTRY WITHOUT TEXT
44 (2C) SIGNED         4    DENTXREF  XREF TO CORRESP. DICT.ENTRY

 ENTRY FOR MODULE NAME
 0  (0)  CHARACTER     6
 6  (6)  CHARACTER     2    LCDFLG    RESERVED
 8  (8)  HEX           1    LCDMTYP   TYPE
 9  (9)  HEX           1    LCDMVER   VERSION
10  (A)  HEX           1    LCDMMOD   MODIFICATION LEVEL
11  (B)  HEX           1
12  (C)  CHARACTER     8    LCDMNAM   MODULE NAME
20 (14)  SIGNED        4
```

```
MASKS FOR ESDTYPD / ESDTYPED
  .... ....      SD   SECTION DEFINITION
  .... ...1      LD   LABEL DEFINITION
  .... ..1.      ER   EXTERNAL REFERENCE
  .... ..11      LR   LABEL REFERENCE
  .... .1..      PC   PRIVATE CODE
  .... .1.1      CM   COMMON
  .... .11.      PR   PSEUDO REGISTER
  .... .111      PH   PHASE ENTRY
  .... 1.1.      WX   WEAK EXTERNAL
  ...1 ....      IC   INCLUDED MODULE NAME
  ..1. ....      DE   SAVED DIRECTORY ENTRY

MASK FOR SWITCHD
  .... ...1     UNASSG
  .... ..1.     WXTRN    THE ER IS WEAK EXTERNAL
  .... .1..     NOAUTOL  NO AUTOLINK NECESSARY
  .... 1...     SETER    ENTRY SET TO ER
  ...1 ....     ZEROLEN  ZERO LENGTH WAS SPECIFIED IN SD/PC ESD ENTRY
  ..1. ....     LATELINK DUPLICATE UNASSIGNED LD FOUND BY ENTRY ER
  1111 111.     ASSG     MASK TO ASSIGN LD/LR

MASKS FOR SDMODED,PHMODED AND PHESDMD
  1... ....     RMODEASS  RMODE ASSIGNED
  .1.. ....     AMODEASS  AMODE ASSIGNED
  ..1. ....     MODESTMT  VALUES SPECIFIED VIA MODE STATEMENT
  ...1 ....     PARMFLD   VALUES SPECIFIED VIA PARM FIELD
  .... .1..     RMODE     1: RMODE=ANY, 0: RMODE=24
  .... ..11     AMODE     11:       AMODE=ANY
                          10:       AMODE=31
                          00 or 01: AMODE=24

MASKS FOR PHTYPED
  1... ....     SELFRELO  SELFRELOCATING PHASE
  .1.. ....     RELPHASE  RELOCATABLE PHASE
  ..1. ....     SVAELIG   SVA ELIGIBLE
  .... ...1     SVAPFIX   PHASE SHOULD BE PFIXED IN SVA
EQU   X'00'               NOT RELOCATABLE
```

Linkage Table

Function: The language translator gives each ESD entry (except LD entries) a number called ESID. This number is unique in the module. The information of all entries encountered during the link job are accumulated in the control dictionary. Each CD entry is identified by a CD number. For the time a module is handled by the linkage editor the linkage table is used to link the modules' ESIDs to the related CD numbers.

Name: Linkage Table

Label or Identifier: None

Location: Partition

Changed By: IJBESD and IJBCTL

Used By: All except INLPLEIT

Pointed to By: LTMINE (= address of first item in linkage table minus 3)

Layout: A linkage table has up to 32768 3-byte entries. Figure 18 shows the format of an entry. Each object module has its own linkage table. When an END record is processed, signalling the end of a module, the table is reset to zeros.

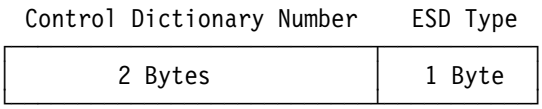


Figure 18. Linkage Table Entry Format

LIFO Stack (PERIDA)

Function: The LIFO stack is built when an INCLUDE statement is processed and used to

1. obtain the address of the next record after the END statement
2. determine the end of processing for an object module
3. control the nesting of INCLUDE statements to give priorities.

Name: LIFO stack

Label: PERIDA

Location: Top in IJBLNK

Initialized By: IJBINL

Changed By: IJBCTL

Pointed to By: LCSTBEG

Layout:

Label	Offset	Length	Contents
PERIDA	0	6	NOTE information for SYSLNK records
PERISW	6	1	Status information: X'01' SYSLNK input X'02' Named submodular X'08' Autolink active for current module X'20' SYSLNK on FBA or VSAM managed space X'80' Library input of type OBJ
PERILRC	7	1	Record count within block
PERIRRN	8	4	Relative record count
PERIMNO	C	2	Cross reference to module name for CSECTs being linked (CD offset)
PERIRL	E	20	NOTE information for relocatable modules

Figure 19. Layout of the LIFO Area

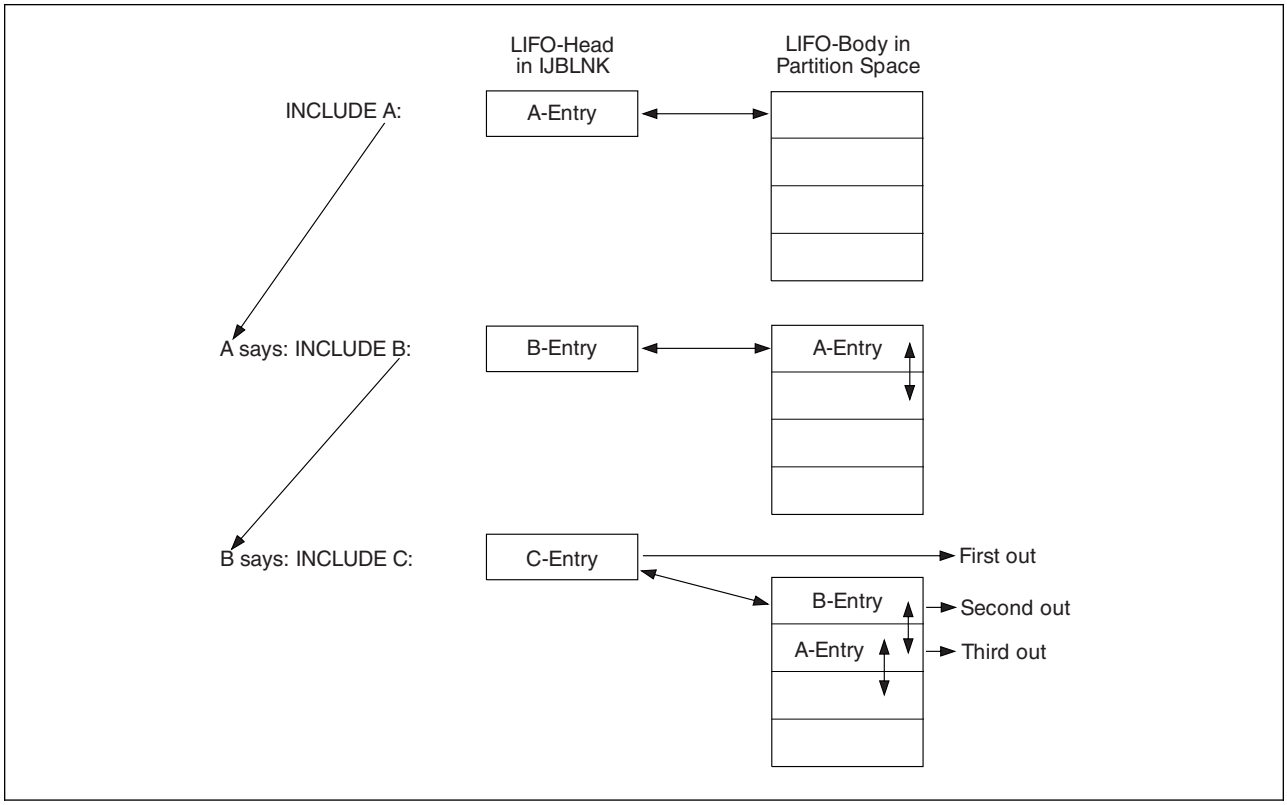


Figure 20. How the LIFO Stack Works

Communication Area LEITPL

Function: Communication area between module (and CSECT) INLPLEIT and the CSECTs of module IJBLE1.

Name: LEITPL

Label: LEITPL

Location: CSECT IJBLNK

Changed By: All CSECTs of module IJBLE1.

Layout:

```
OFFSETS TYPE        LENGTH  NAME      DESCRIPTION

0   (0) STRUCTURE     0    LEITPL
0   (0) SIGNED        4

   LINKAGE EDITOR AND LIBRARIAN SERVICES INTERFACE
0   (0) V-ADDRESS     4    LEITA     "V(INLPLEIT)" ENTRY ADDR
                                     TO LE INTERFACE MODULE
4   (4) SIGNED        4    LEITPL    LE INTERFACE PARM LIST
4   (4) HEX           1    LEITFC    FUNCTION CODE
5   (5) HEX           1    LEITSW    SWITCHES
        1... ....           LEITCHK   CHECK ON RECORD REQUEST
        .1.. ....           LEITMOLD  CONNECT USING OLD MACB
        ..1. ....           LEITLINK  LINK OPTION
        ...1 ....           LEITMSCN  RUN UNDER MSHP CNTL
        .... 1...           LEITMSPA  PHASE UNDER MSHP CNTL
        .... .1..           LEITMSBY  MSHP BYPASS REQUEST
        .... ..1.           LEITMS1T  1TH TIME MSHP
        .... ...1           LEITMFND  PHASE ALREADY IN LIBRARY
6   (6) HEX           1    LEITSW1
        1... ....           LEITNOCA  NO CATALOG SUBLIB GIVEN
        .1.. ....           LEITST1T  1ST TIME STOW
        ..1. ....           LEITOOM   REQUEST OUTSIDE OF MODULE
        ...1 ....           LEITSEVE  SEVERE ERROR OCCURED
        .... 1...           LEITFULL  LIBRARY FULL
        .... .1..           LEITSDEL  FORCE STOW DELETE
        .... ..1.           LEITGVIS  LAMB AVAILABLE FOR GETVIS
        .... ...1           LEITCCON  CATALOG LIBRARY CONNECTED
7   (7) HEX           1    LEITSW2
        1... ....           LEITDCAN  DELAYED CANCEL FORCED
        .1.. ....           LEITCARQ  CANCEL REQUEST PENDING
        ..1. ....           LEITSCON  A SEARCH SUBLIBRARY CONNECTED
        ...1 ....           LEITERRS  ERROR DURING STOW
        .... 1...           LEITEOM   END OF OBJ MODULE
```

```
OFFSETS TYPE        LENGTH  NAME      DESCRIPTION

8   (8) HEX           8     LEITSBUF  SBUF DESCRIPTOR
16  (10) HEX          8     LEITPBUF  PBUF DESCRIPTOR
24  (18) A-ADDRESS    4     LEITCOMM  COMMUN. AREA PTR
28  (1C) CHARACTER    8     LEITPNAM  PHASE NAME
36  (24) SIGNED       4     LEITMARK  HI WATER MARK
40  (28) SIGNED       4     LEITOFFS  OFFSET FOR PUT
44  (2C) A-ADDRESS    4     LEITTXT   ADDR. OF TEXT
48  (30) SIGNED       4     LEITTXTL  LENGTH OF TEXT
52  (34) A-ADDRESS    4     LEITSTOW  ADDR OF STOW LIST
56  (38) HEX          1
57  (39) HEX          1     LEITRC    RETURN CODE
58  (3A) HEX          1
59  (3B) HEX          1     LEITEC    ERROR CODE
60  (3C) SIGNED       4     LEITVIOO  PHASE START ADDRESS
64  (40) A-ADDRESS    4     LEITPARB  ADDR. OF CALL I/F PARM LIST
68  (44) HEX          1     LEITCISW  CALL I/F SWITCH BYTE
        1... ....           LEITCITA  CALL I/F ACTIVE
        .1.. ....           LEITCIMS  ID IN INLCPARB IS MSHP
        ..1. ....           LEITCITC  OPTION CATAL VIA CALL I/F
        ...1 ....           LEITNORP  NOREPLACE VIA CALL I/F
        .... 1...           LEITNOLG  NO SYSLOG EXIT
        .... .1..           LEITNOLS  NO SYSLST EXIT
69  (45) HEX          1     LEITMODE  AMODE/RMODE VIA CALL I/F
70  (46) HEX          1
71  (47) HEX          1
72  (48) A-ADDRESS    4     LEITLOGA  ADDR. OF SYSLOG OUTPUT AREA
76  (4c) A-ADDRESS    4     LEITLOGL  SYSLOG OUTPUT AREA LENGTH
80  (50) A-ADDRESS    4     LEITLSTA  ADDR. OF SYSLST OUTPUT AREA
84  (54) A-ADDRESS    4     LEITLSTL  SYSLST OUTPUT AREA LENGTH
88  (58) A-ADDRESS    4     LEITDTF   ADDR. OF LIST DTF
92  (5c) A-ADDRESS    4     LEITIO    ADDR. OF DTF I/O AREA
96  (60) HEX        188     LEITARG   LENGTH IS MULTIPLE OF (CD_ENTRY_LGT - 1) as
                              DIR.ENTRY FOR STOW CD FIELDS

 FUNCTION CODES IN LEITFC
        .... ...1           OPENCAT
        .... ..1.           NEWMEM
        .... ..11           DISCMEM
        .... .1..           DISCLIB
        .... .1.1           PUTREC
        .... .11.           GETREC
        .... .111           STOW
        .... 1...           FIND
        .... 1..1           NOTE
        .... 1.1.           POINT
        .... 1.11           UPDTE
        .... 11..           OPENOBJ
        .... 111.           FINISH
        .... 1111           CHKCIT

 ERROR  CODES IN LEITEC
        ...1 11..           NOCATSL    NO CATALOG SUBLIB
        ..1. ....           VIOEX      VIO EXHAUSTED
        ..11 ....           MSHPCNPA   PHASE CANNOT BE REPLACED
        .1.. ....           LIBFULL    LIBRARY IS FULL
```

Librarian Data Areas Used by the Linkage Editor

Stow Table

Function: In this table the directory information for all phases produced during a linkage editor run is collected for the sublibrary directory.

Name: Stow Table

Label: None

Pointed to By: LTMINE in IJBLNK

Initialized By: IJBCAT

Changed By: IJBCAT

Layout: See *VSE/AF Diagnosis Reference: Librarian* SC33-6330, data area INLCDENT.

Buffers for Librarian Services

Function: The linkage editor provides these buffers for the librarian services it requests:
The S-buffer is a shared buffer for library management of the CATALOG sublibrary. The P-buffers are used for member I/O, one to put phases into the CATALOG sublibrary and one to read input (modules or phases) from OBJ-search chain sublibraries.

Names: S-Buffer and P-Buffers

Labels: None

Pointed to By: SHARBUF, PHAPBUF, OBJPBUF in IJBLNK

Location: Partition of linkage editor

Allocated By: IJBINL

Formatted By: Librarian

Changed By: Librarian

Layout: See *VSE/AF Diagnosis Reference: Librarian* SC33-6330, data areas INLCBUCB and INLCBHDR.

Diagnostics

Linkage Editor Listing and Map

The linkage editor program lists for each execution the errors encountered and actions taken. This listing is followed by a linkage editor map prepared by CSECT IJBMAP.

For a detailed description of the linkage editor listing and map see *VSE/ESA Diagnosis Tools* SC33-6614.

The linkage editor map has the following columns:

1. Phase name
2. Transfer address
3. Start and end of the virtual storage location
4. AMODE and RMODE assigned to the phase
5. For each CSECT of the phase:
   - labels in ascending order
   - load address
   - relocation factor
   - offset of CSECT in the partition where link-edited
   - offset of CSECT in phase
   - AMODE and RMODE of each CSECT contributing to the phase
   - name of object module from which the CSECT was taken (or SYSLNK)

Figure 21 on page 50 shows an example of a linkage editor map.

```
PHASE    XFR-AD  LOCORE  HICORE  CSECT/   LOADED  RELOC. PARTIT PHASE  TAKEN AMODE/RMODE
                                 ENTRY    AT      FACTOR OFFSET OFFSET FROM
-------------------------------------------------------------------------------------
ASSEMBLY 03C282  03A078  03CC40                                              *P ANY 24
-------------------------------------------------------------------------------------
                                 IPKAJ000 03A078  03A078 000000 000000 MODA    24  24
                                 IPKAA002 03A080  039600 000008 000008 MODA    24  24
                                 IPKAA000 03AA38  03AA38 0009C0 0009C0 MODA    24  24
                                *IPKAA501 03B228
                                *IPKAA502 03B1F8
                                *IPKAA503 03B1C0
                                *IPKAA504 03B1E8
                                *IPKAA505 03B1E8
                                *IPKAA506 03B200
                                *IPKAA507 03B1C8
                                *IPKAA508 03B258
                                *IPKAA509 03B258
                                *IPKAA511 03B1E8
                                *IPKAA512 03B210
                                +IPKAA101 03AAD0
                                +IPKAA102 03AB98
                                +IPKAA103 03AC60
                                 IPKAB000 03B378  03B378 001630 001630 MODB    24  24
                                +IPKAB100 03B388
                                +IPKAB101 03B520
                                +IPKAB103 03B58E
                                +IPKAB102 03B5C0
                                 IPKAG000 03B6B0  03B6B0 001638 001638 MODB    24  24
                                 IJJCPD2  03B6B8  03B6B0 001640 001640 MODB    24  24
                                *IJJCPD3  03B6B8
                                 IPKAD000 03B8E8  03B8E8 001870 001870 MODC    24  24
                                *IPKAD101 03B8F8
                                 IPKAD100 03B900  03B8E8 001888 001888 MODC    24  24
                                 IPKBA000 03BF80  03BF80 001F08 001F08 MODC    24  24
-------------------------------------------------------------------------------------
ASSETA   03B8F0  03B8E8  03C388                                              *M 31  24
-------------------------------------------------------------------------------------
                                 IPKTA000 03B8E8  03B8E8 001870 000000 MODD    24  24
                                 IJ2M0093 03C010  03B8E8 001F98 000728 MODD    24  24
                                 IJJCPD1N 03C030  03B8E8 001FB8 000748 MODD    24  24
-------------------------------------------------------------------------------------
ASSECA   03BF88  03BF80  03F16F                                                24  24
-------------------------------------------------------------------------------------
                                 IPKCA001 03BF80  03BF80 001F08 000000 MODE    24  24
                                +IPKCA998 03BF88
etc.
```

'+' indicates a referenced entry point
'*' indicates an unreferenced entry point

Figure 21. Example of a Linkage Editor Map Printout

Interfaces

All librarian and supervisor services for retrieval of linkage editor library input and for storage of linkage editor library output are requested via CSECT INLPLEIT. This CSECT uses librarian and supervisor macros and interface control blocks. These are all listed in the operation description of INLPLEIT.

In the partition, buffers are provided for librarian services. They are discussed as “Buffers for Librarian Services” in the Chapter “Data Areas.”

The stow table of the librarian is used by CSECT IJBCAT.

Cross-References

Label-to-CSECT Cross-Reference

The following labels are those which appear in the Sequence of Operation description in this manual.

The CSECT names are shortened where they begin by IJB to show only the last characters.

Label	CSECT	Label	CSECT	Label	CSECT
ABTERR	IJBLNK	GETMODED	IJBMOD	PRINT	IJBLNK
ACTISEC	SCN	HNDLCNT	SRV	PRTLST	LNK
ALNKOF	LNK	IJBLETR	FIN	RDEXEC	LNK
ALNKPR	LNK	INCCRD	SCN	RDNEXT	LNK
CALCWA	INL	INLN000	INL	READ	INLPEIT
CANCL	LNK	LCFIND	FIN	READIN	IJBFIN
CDENT1	LNK	LCLOSE	FIN	REPPRC	OTH
CDSIZE	LNK	LCSTBEG	LNK	RLDCON	RLD
CESDENT	LNK	LEITCALL	FIN	RLDOR	RLD
CHECKCIT	INLPLEIT	LEITPL	LNK	RLDPRC	OTH
CHKMODE	IJBMOD	LNKNOT	FIN	SCANCD	CAT
CNVHDX	LNK	LNKPOINT	FIN	SHARBUF	LNK
CONCLIB	INL	LOGMSG	LNK	SPACE1	LNK
CONNECT	INLPLEIT	LTESID	LNK	SRCHCD	LNK
CPHENT	IJBLNK	LTMINE	LNK	SRPCOD	LNK
DETESDMD	MOD	MODCRD	SCN	STOWML	INLPLEIT
ELBCM	ESD	NOTCTL	LNK	TESTCNT	IJBRLD
ELBPR	ESD	NOTEM	INLPLEIT	TSTUNR	RLD
ELBER	ESD	OBJBUF	LNK	TXTPRC	OTH
ELBLD	ESD	OPCATAL	INL	UPDATE	INLPLEIT
ELBSD	ESD	OPENC	INLPLEIT	WARNING	IJBLNK
ENDPRC	OTH	OPENJ	INLPLEIT	WRITE	INLPLEIT
ENTCRD	SCN	PHAPBUF	IJBLNK	WRLST	IJBRLD
ERROR	LNK	PHCRD	SCN	WRTRFR	CTL
ESDRET	ESD	PHSPRC	CTL	XTPHGT	LNK
EXLOAD	LNK	POINTM	INLPLEIT	XTPHNO	LNK
FINDM	INLPLEIT				

Message-to-CSECT Cross-Reference

(only the causing CSECT is listed here)

Message	CSECT
2100	IJBOTH
01	SCN
02	LNK, SCN
10	SCN
11	SCN
12	SCN
13	OTH, SCN
14	SCN
15	CTL
16	SCN, LNK
18	SCN
19	INL
20	CTL
21	SCN
22	CTL
23	CTL
24	CTL
25	SCN
32	SCN, LNK
33	SCN
35	SCN
36	SCN
39	ESD
40	ESD
41	ESD
43	ESD
44	LNK, FIN
45	ESD
46	ESD
47	OTH
48	ESD
49	MAP, RLD
50	OTH
51	OTH
55	OTH

Message	CSECT
2156	IJBOTH
58	OTH
60	CTL
61	SCN
65	MAP
66	MAP
67	MAP
68	MAP
69	MAP
70	LNK
71	SCN
72	CTL
73	INL
74	ESD
75	INL
76	SCN
77	MAP
78	MAP
79	MAP
80	SCN
81	CTL
82	CTL
83	LNK
85	MAP
86	CTL
87	MAP
88	OTH
89	MAP
90	INL
91	OTH, INL
92	CAT
93	INLPLEIT
95	INLPLEIT
97	IJBLNK
99	MAP,INL

Input-to-CSECT Cross-Reference

Statement	CSECT
ACTION	IJBSCN
INCLUDE	IJBSCN
MODE	IJBSCN
PHASE	IJBSCN, IJBCTL
ENTRY	IJBSCN, IJBCTL

Input Record	CSECT
ESD	IJBESD
TXT	IJBOTH
RLD	IJBOTH
REP	IJBOTH
END	IJBOTH

Phase-to-Module Cross-Reference

Phase	Module
$LNKEDT	IJBLE1
	INLPLEIT
	INLPQNAM

Index

M

N

O

P

R

S

T

V

W

Communicating Your Comments to IBM

IBM VSE/Enterprise Systems Architecture
VSE Central Functions
Linkage Editor
Diagnosis Reference
Version 6 Release 1

Publication No. SC33-6328-00

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of the book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF form and either send it postage-paid in the United States, or directly to:

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

- If you prefer to send comments by FAX, use this number:
  - (Germany): 07031-16-3456
  - (Other countries): (+49)+7031-16-3456
- If you prefer to send comments electronically, use this network ID:

  INTERNET: s390id@de.ibm.com

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.

Readers' Comments — We'd Like to Hear from You

IBM VSE/Enterprise Systems Architecture
VSE Central Functions
Linkage Editor
Diagnosis Reference
Version 6 Release 1

Publication No. SC33-6328-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
SC33-6328-00

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

PLACE
POSTAGE
STAMP
HERE

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

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

File Number: S370/S390-37
Program Number: 5686-066

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC33-6328-00