High Level Assembler for MVS® & VM & VSE

Installation and Customization Guide

Release 4

SC26-3494-02

High Level Assembler for MVS® & VM & VSE

Installation and Customization Guide

Release 4

SC26-3494-02

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 213.

Fourth Edition (September 2000)

This edition applies to IBM High Level Assembler for MVS & VM & VSE, Release 4, Program Number 5696-234 and to any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

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

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

IBM Corporation, Department BWE/H3
P.O.Box 49023
SAN JOSE, CA 95161-9023
United States of America

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

Contents

About This Book . . . vii
Brief Overview of High Level Assembler . . . vii
Who Should Use This Book . . . vii
Experienced Users Installing on MVS . . . vii
Experienced Users Installing on VM . . . viii
Experienced Users Installing on VSE . . . viii
Terminology in This Book . . . viii
List of APARs and PTFs in this Book . . . viii
Other Documentation You Might Need . . . viii
Syntax Notation . . . viii

Summary of Changes . . . xi

On MVS: Planning for, Installing, Customizing, and Maintaining High Level Assembler . . . 1

Chapter 1. Planning for Installing High Level Assembler on MVS . . . 3
Worksheet: Planning for Installing High Level Assembler on MVS . . . 3
What You Receive with High Level Assembler . . . 4
What You Need to Install High Level Assembler . . . 5
FMIDs Deleted . . . 9
Installing with CBIPO or CBPDO . . . 10
Planning to Use SMP/E . . . 10
Planning to Change Installation Jobs . . . 10
Selecting National Language Support . . . 10
Program Support . . . 11
Program and Service Level Information . . . 12
Publications Useful during Installation . . . 12

Chapter 2. Planning for Customizing High Level Assembler on MVS . . . 13
Deciding Whether and What to Customize . . . 13
Impact of making SMP/E use High Level Assembler . . . 13
Planning to Customize High Level Assembler Options . . . 13
Planning to Customize the User Exits . . . 16
Planning to Customize Cataloged Procedures . . . 16
Planning to Install in the Link Pack Area . . . 19

Chapter 3. Installing High Level Assembler on MVS . . . 21
Overview of Installation . . . 21
Step 1: SMP/E Considerations for Installing High Level Assembler . . . 22
Step 2: Change the Sample JCL from the Product Tape . . . 23
Step 3: Perform SMP/E Receive . . . 24
Step 4: Set up Target and Distribution Libraries . . . 25
Step 5: Create DDDEF Entries . . . 25
Step 6: Perform SMP/E Apply . . . 25
Step 7: Run the Installation Verification Program . . . 26
Step 8: Perform SMP/E Accept . . . 26

Chapter 4. Customizing High Level Assembler on MVS . . . 29

SMP/E Considerations 29
Customization Checklist 29
Customization Steps 30

Chapter 5. Maintaining High Level Assembler on MVS 45
Re-installing High Level Assembler 45
Applying Service Updates 45
Removing High Level Assembler 47
Reporting a Problem with High Level Assembler 47
Obtaining Service Information 48

On VM: Planning for, Installing, Customizing, and Maintaining High Level Assembler 49

Chapter 6. Planning for Installing High Level Assembler on VM 51
Worksheet: Planning for Installing High Level Assembler on VM 51
What You Receive with High Level Assembler 52
What You Need to Install High Level Assembler 53
Planning to Install in a Saved Segment 56
Selecting National Language Support 56
Program Support 56
Program and Service Level Information 57
Publications Useful during Installation 58

Chapter 7. Planning for Customizing High Level Assembler on VM 59
Deciding Whether and What to Customize 59
Planning to Customize High Level Assembler Options 59
Planning to Customize the User Exits 62

Chapter 8. Installing High Level Assembler on VM 63
Overview of Installation 63
Step 1: Prepare to Install High Level Assembler 63
Step 2: Allocate Resources for Installing High Level Assembler 65
Step 3: Install High Level Assembler 68
Step 4: Verify the Installation in a Test Environment 70
Step 5: Define and Build High Level Assembler Saved Segment (Optional) . . 71
Step 6: Put High Level Assembler into Production 77

Chapter 9. Customizing High Level Assembler on VM 81
Changing Option and DDNAMES Defaults 81
General Rules for coding the ASMADOPT ASSEMBLE file 81
Changing Saved Segment Name 82
Changing the User Exits 82

Chapter 10. Maintaining High Level Assembler on VM 83
Re-installing High Level Assembler 83
Applying Service Updates 83
Removing High Level Assembler 96
To Report a Problem with High Level Assembler 96
Obtaining Service Information 97

On VSE: Planning for, Installing, Customizing, and Maintaining High Level Assembler . . . 99

Chapter 11. Planning for Installing High Level Assembler on VSE . . . 101
Worksheet: Planning for Installing High Level Assembler on VSE . . . 101
What You Receive with High Level Assembler . . . 102
What You Need to Install High Level Assembler . . . 103
Planning Where to Install High Level Assembler . . . 104
Selecting National Language Support . . . 104
Program Support . . . 105
Program and Service Level Information . . . 105
Publications Useful during Installation . . . 106

Chapter 12. Planning for Customizing High Level Assembler on VSE . . . 107
Deciding Whether and What to Customize . . . 107
Planning to Customize High Level Assembler Options . . . 107
Planning to Install High Level Assembler into the Shared Virtual Area (SVA) . . . 109

Chapter 13. Installing High Level Assembler on VSE . . . 111
Overview of Installation . . . 111
Step 1: Back Up the Original System . . . 111
Step 2: Allocate Space for the Library (Omit if Using the Default Sublibrary) . . . 112
Step 3: Install High Level Assembler . . . 113
Step 4: Verify the Installation of High Level Assembler . . . 119

Chapter 14. Customizing High Level Assembler on VSE . . . 123
Changing Default Options . . . 123
Placing High Level Assembler into the Shared Virtual Area (SVA) . . . 125

Chapter 15. Maintaining High Level Assembler on VSE . . . 127
Re-installing High Level Assembler . . . 127
Applying Service Updates . . . 127
Removing High Level Assembler . . . 130
To Report a Problem with High Level Assembler . . . 131
Obtaining Service Information . . . 131

Diagnosing Problems with High Level Assembler . . . 133

Chapter 16. Isolating the Problem . . . 135
Diagnosing the Problem . . . 135
Diagnostic Aids . . . 136
Installation Problems . . . 137

Chapter 17. Building a Keyword String . . . 139
Keyword Usage . . . 139
Using the Problem Identification Worksheet . . . 140
Step 1. Component Identification Keyword Procedure . . . 141
Step 2. Release Level Keyword Procedure . . . 141
Step 3. Type-of-Failure Keyword Procedure . . . 142
Step 4. System-Type Keyword . . . 149

Chapter 18. Using the Keyword String as a Search Argument . . . 151

How to Use the Keyword String . . . 151
Search Argument Procedure . . . 151

Chapter 19. Preparing an APAR . . . 153
Initiating an APAR . . . 153
Submitting the APAR Documentation . . . 156

Chapter 20. Modifier Keywords . . . 159
High Level Assembler Language Elements . . . 160
Assembler Options . . . 162

Chapter 21. Problem Identification Worksheet . . . 165

Chapter 22. Internal Trace Facility . . . 167
How to Invoke the Internal Trace Facility . . . 167
Specifying the Trace Data set . . . 167
ITF Assembler Option . . . 168

Appendixes . . . 169

Appendix A. High Level Assembler Options . . . 171
ASMAOPT . . . 171
ASMAOPT Options . . . 171

Appendix B. High Level Assembler DDNAMES (MVS and CMS) . . . 203
ASMADD Syntax . . . 203
ASMADD . . . 203

Appendix C. High Level Assembler Service . . . 207

Appendix D. Create Product Parameter File (PPF) Override (VM) . . . 209

Appendix E. Local Modification Procedures (VM) . . . 211

Notices . . . 213
Trademarks . . . 214

Glossary . . . 215

Bibliography . . . 223
High Level Assembler Publications . . . 223
Toolkit Feature Publications . . . 223
Related Publications (Architecture) . . . 223
Related Publications for MVS . . . 223
Related Publications for VM . . . 224
Related Publications for VSE . . . 224
General Publications . . . 224

Index . . . 225

About This Book

This book provides information needed to plan for, install, customize, maintain, and diagnose problems with High Level Assembler on MVS and VM and VSE.

Throughout this book, we use these indicators to identify platform-specific information:

- Prefix the text with platform-specific text (for example, “Under CMS...”)
- Add parenthetical qualifications (for example, “(CMS only)”)
- Bracket the text with icons. The following are some of the icons that we use:

Informs you of information specific to MVS


Informs you of information specific to CMS


Informs you of information specific to VSE


MVS is used in this manual to refer to Multiple Virtual Storage/Enterprise Systems Architecture (MVS/ESA™) and to OS/390®.

CMS is used in this manual to refer to Conversational Monitor System on Virtual Machine/Enterprise Systems Architecture (VM/ESA®).

VSE is used in this manual to refer to Virtual Storage Extended/Enterprise Systems Architecture (VSE/ESA®).

Brief Overview of High Level Assembler

High Level Assembler is an IBM* licensed program that helps you develop programs and subroutines using assembler language to provide functions not typically provided by other symbolic languages, such as COBOL, FORTRAN, and PL/I.

Who Should Use This Book

This book is for system programmers and system administrators who plan for, install, customize, and maintain High Level Assembler on MVS, VM or VSE.

It is also relevant to users who carry out diagnostic tasks on this product.

To use this book, you need to be familiar with the MVS, VM or VSE operating system, the publications that describe your system, and job control language (JCL) or EXEC processing.

Experienced Users Installing on MVS

If you are installing on MVS and are experienced in installing products with SMP/E, use the following fast path items:

1. DASD storage for target and distribution libraries is on Table 7 on page 8 and Table 8 on page 8.

2. DASD storage for SMP/E data sets are outlined in Table 6 on page 8 and Table 5 on page 7.
3. Carry out steps "Step 1: SMP/E Considerations for Installing High Level Assembler" on page 22 through to "Step 8: Perform SMP/E Accept" on page 26.

Experienced Users Installing on VM

If you are installing on VM and are experienced in installing products with VMSES/E, use the following fast path items:

1. Refer to Table 28 on page 55 for DASD storage requirements.
2. Carry out steps "Step 1: Prepare to Install High Level Assembler" on page 63 through to "Step 6: Put High Level Assembler into Production" on page 77.

Experienced Users Installing on VSE

For those installing on VSE who are experienced in installing products with MSHP, use the following fast path:

- Refer to Table 41 on page 104 for DASD storage requirements.
- Carry out steps "Step 1: Back Up the Original System" on page 111 through to "Step 4: Verify the Installation of High Level Assembler" on page 119.

Terminology in This Book

For simplicity, many of the full IBM product names are shortened to just the generic acronym. For example, MVS is used to mean MVS/ESA*.

List of APARs and PTFs in this Book

This book includes the numbers for APARs and PTFs that are included in this product (see Appendix C, "High Level Assembler Service" on page 207). You might want to use this list, for example, to determine whether a fix that you have applied to the previous release is in this release that is to be installed.

To obtain current service recommendations and to identify current product service requirements, get the Preventive Service Planning (PSP) information or check with the IBM Support Center.

Other Documentation You Might Need

The complete list of High Level Assembler publications and order numbers is published on the back cover. Publications from related IBM products are in "Bibliography" on page 223.

Syntax Notation

Throughout this book, syntax descriptions use the structure defined below.

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

  The ►►— symbol indicates the beginning of a statement.

The ⟶ symbol indicates that the statement syntax is continued on the next line.

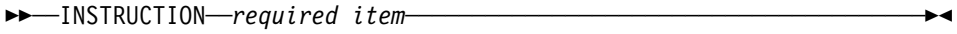
The ►— symbol indicates that a statement is continued from the previous line.

The ⟶◄ indicates the end of a statement.

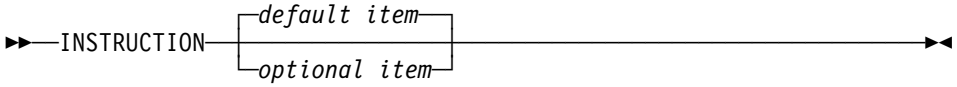
Diagrams of syntactical units other than complete statements start with the ►— symbol and end with the ⟶ symbol.

- **Keywords** appear in uppercase letters (for example, ASPACE) or upper and lower case (for example, PATHFile). They must be spelled exactly as shown. Lower case letters are optional (for example, you could enter the PATHFile keyword as PATHF, PATHFI, PATHFIL or PATHFILE).

  Variables appear in all lowercase letters in a special typeface (for example, *integer*). They represent user-supplied names or values.
- If punctuation marks, parentheses, or such symbols are shown, they must be entered as part of the syntax.
- Required items appear on the horizontal line (the main path).

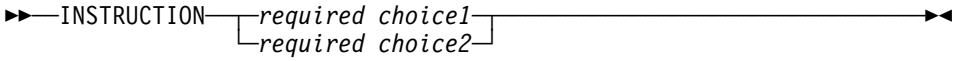


- Optional items appear below the main path. If the item is optional and is the default, the item appears above the main path.

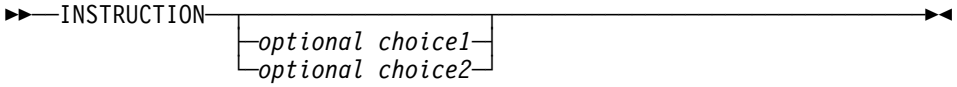


- When you can choose from two or more items, they appear vertically in a stack.

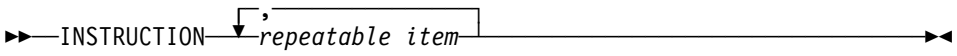
  If you **must** choose one of the items, one item of the stack appears on the main path.



  If choosing one of the items is optional, the whole stack appears below the main path.



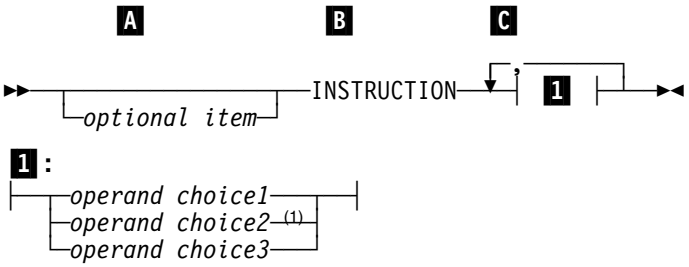
- An arrow returning to the left above the main line indicates an item that can be repeated. When the repeat arrow contains a separator character, such as a comma, you must separate items with the separator character.



  A repeat arrow above a stack indicates that you can make more than one choice from the stacked items, or repeat a single choice.

The following example shows how the syntax is used.

Format



Note:
[1] *operand choice2* and *operand choice3* must not be specified together

A The item is optional, and can be coded or not.

B The INSTRUCTION key word must be specified and coded as shown.

C The item referred to by **1** is a required operand. Allowable choices for this operand are given in the fragment of the syntax diagram shown below **1** at the bottom of the diagram. The operand can also be repeated. That is, more than one choice can be specified, with each choice separated by a comma.

Summary of Changes

Date of Publication September 2000

Form of Publication Fourth Edition, SC26-3494-02

Assembler options changes

- Options file allows options to be specified via an external file.
- *PROCESS OVERRIDE
- New options:

  THREAD

  CODEPAGE

New assembler statement

- XATTR statement allows attributes of external symbols to be specified.

Changed assembler statements

- DC new constant types:

  R PSECT address constant

  CU Unicode character constants

  AD Doubleword aligned 8-byte address

  FD Doubleword aligned 8-byte fixed point constant

  – Floating point symbolic value DMIN added
- AMODE
  – ANY31 operand added
  – 64 operand documented
- RMODE
  – 31 operand added
  – 64 operand documented

Miscellany

- Literals now always entered in literal pool
- Predefined absolute symbols may no longer be used in conditional assembly character expressions.

On MVS: Planning for, Installing, Customizing, and Maintaining High Level Assembler

Chapter 1. Planning for Installing High Level Assembler on MVS 3
Worksheet: Planning for Installing High Level Assembler on MVS 3
What You Receive with High Level Assembler 4
Distribution Media 4
Basic Material 4
Optional Material 5
Program Publications and Softcopy 5
Program Source Materials 5
What You Need to Install High Level Assembler 5
Required and Optional Software for the Target System 6
DASD Storage Required for the Target System 6
FMIDs Deleted 9
Installing with CBIPO or CBPDO 10
Planning to Use SMP/E 10
Planning to Change Installation Jobs 10
Selecting National Language Support 10
Program Support 11
Program Services 11
Preventive Service Planning 11
Statement of Support Procedures 11
Program and Service Level Information 12
Program Level Information 12
Service Level Information 12
Cumulative Service Tape 12
Publications Useful during Installation 12

Chapter 2. Planning for Customizing High Level Assembler on MVS . . . 13
Deciding Whether and What to Customize 13
Impact of making SMP/E use High Level Assembler 13
Planning to Customize High Level Assembler Options 13
Why Do It 14
Choices to Make Now 14
Planning to Customize the User Exits 16
Planning to Customize Cataloged Procedures 16
Why Do It 16
Choices to Make Now 17
An Example 18
Planning to Install in the Link Pack Area 19
Why Do It 19
Choices to Make Now 19

Chapter 3. Installing High Level Assembler on MVS 21
Overview of Installation 21
Modifying High Level Assembler Jobs 21
If Using a CBPDO 21
Installation Checklist 22
Step 1: SMP/E Considerations for Installing High Level Assembler 22
SMP/E Environment 22
SMP/E Options Subentry Values 23

Step 2: Change the Sample JCL from the Product Tape 23
Step 3: Perform SMP/E Receive 24
Method 1: Receive Using SMP/E Panels 24
Method 2: Receive Using Batch Jobs 24
Step 4: Set up Target and Distribution Libraries 25
Step 5: Create DDDEF Entries 25
Step 6: Perform SMP/E Apply 25
Method 1: Apply Using SMP/E Panels 25
Method 2: Apply Using Batch Jobs 26
Step 7: Run the Installation Verification Program 26
Step 8: Perform SMP/E Accept 26
Method 1: Accept Using SMP/E Panels 27
Method 2: Accept Using Batch Jobs 27

Chapter 4. Customizing High Level Assembler on MVS 29
SMP/E Considerations 29
Customization Checklist 29
Customization Steps 30
Step 1: Change SMP/E to use High Level Assembler 30
Step 2: Customize user exits 32
Step 3: Change Default Options and DDNAMES 32
Step 4: Place Procedures in Procedure Library 35
Step 5: Place High Level Assembler into Link Pack Area 41
Step 6: Assembler H Migration (optional) 41

Chapter 5. Maintaining High Level Assembler on MVS 45
Re-installing High Level Assembler 45
Applying Service Updates 45
What You Receive 45
Checklist for Applying Service 45
Step 1. Prepare to Install Service 46
Step 2. Receive the Service 46
Step 3. Accept Applied Service (Optional) 46
Step 4. Apply the Service 46
Step 5. Test the Service 46
Step 6. Accept the Service 47
Removing High Level Assembler 47
Reporting a Problem with High Level Assembler 47
Obtaining Service Information 48

Chapter 1. Planning for Installing High Level Assembler on MVS

This section contains the following planning information to help you properly install High Level Assembler on MVS:

- Worksheet: Planning for Installing High Level Assembler on MVS
- What You Receive with High Level Assembler
- What You Need to Install High Level Assembler
- FMIDs Deleted
- Installing with CBIPO* or CBPDO*
- Planning to Use SMP/E*
- Planning to Change Installation Jobs
- Selecting National Language Support
- Program Support
- Program and Service Level Information
- Publications Useful during Installation

Worksheet: Planning for Installing High Level Assembler on MVS

Before you begin the installation, you should:

__ 1. Verify the product parts to be installed.

__ FMID *HMQ4140*

__ Feature number _____

__ VOLSER *MQ4140*

Refer to “What You Receive with High Level Assembler” on page 4.

__ 2. Verify that required software (and optional software, if appropriate) is at the level needed.

Refer to “What You Need to Install High Level Assembler” on page 5.

__ 3. Verify that adequate DASD storage is available:

__ Target libraries

__ Distribution libraries

__ SMP/E data sets

Refer to “DASD Storage Required for the Target System” on page 6.

__ 4. Determine how you are going to use SMP/E:

__ Sample batch jobs

__ SMP/E dialogs

Refer to “Planning to Use SMP/E” on page 10.

__ 5. Plan for changes needed in installation jobs.

Refer to “Planning to Change Installation Jobs” on page 10.

__ 6. Decide what language to use for diagnostic messages.

Refer to “Selecting National Language Support” on page 10.

__ 7. Obtain latest service updates needed.

Refer to “Program Support” on page 11.

What You Receive with High Level Assembler

You receive the following when you order High Level Assembler:

FMIDs	Feature Numbers	System Name
HMQ4140	5891/5892/5488	MVS/ESA

Distribution Media

High Level Assembler is distributed on one of the following:

- Standard-labeled 9-track magnetic tape written at 6250 BPI
- 3480 tape cartridge
- 4mm cartridge

The tape or cartridge contains all the programs and data needed for installation. The tape and cartridge are in SMP/E relative file format. The first file contains the SMP/E modification control statements. Subsequent files contain IEBCOPY, unloaded partitioned data sets, which SMP/E processes.

Basic Material

Table 1 describes the tape or cartridge. Table 2 describes the file content of the program tape or cartridge.

Table 1. Basic Material: Program Tape

Medium	Feature Number	Physical Volume	External Label Identification	VOLSER
6250 tape.	5891	1	HLASM V1R4 for MVS	MQ4140
3480 cart.	5892	1	HLASM V1R4 for MVS	MQ4140
4mm cart.	5488	1	HLASM V1R4 for MVS	MQ4140

Table 2. Program Tape: File Content

Volser	File	Name	Dist. Library	ORG	RECFM	LRECL	BLK SIZE
MQ4140	1	SMPMCS	n/a	SEQ	FB	80	6400
	2	IBM.HMQ4140.F1	AASMSAM1	PDS	FB	80	8800
	3	IBM.HMQ4140.F2	AASMMAC1	PDS	FB	80	8800
	4	IBM.HMQ4140.F3	AASMMOD1	PDS	U	0	6144
	5	IBM.HMQ4140.F4	jclin	PDS	FB	80	8800

The recfm, lrecl, and blk size reflect the original values of the partitioned data sets (relative files) before being unloaded by the IEBCOPY utility to tape.

Note: If you are installing High Level Assembler using the MVS Custom-Built Product Delivery Offering (CBPDO) (5751-CS3), some of the information in these figures may not be valid. Consult the CBPDO documentation for actual values.

Optional Material

There are no optional machine-readable materials for High Level Assembler.

Program Publications and Softcopy

This section identifies the basic and optional publications for High Level Assembler.

One printed copy of the following manuals is included with the product:

- *HLASM Licensed Program Specifications* GC26-4944
- *HLASM Installation and Customization Guide* SC26-3494
- *HLASM Language Reference* SC26-4940
- *HLASM Programmer's Guide* SC26-4941
- *HLASM General Information* GC26-4943

For a list of books for related products, see “Bibliography” on page 223.

Program Source Materials

No program source materials or viewable program listings are provided with High Level Assembler.

What You Need to Install High Level Assembler

This section identifies the system requirements for installing High Level Assembler. You need to plan for two different system environments:

- *Target system*—the system comprising the set of libraries that are updated by SMP/E and system utilities during installation
- *Driving system*—the system on which the jobs are run to install the program

The machine-readable components for High Level Assembler are installed in target and distribution libraries.

- *Target libraries* are the data sets in which the run-time copy of High Level Assembler is stored. These data sets include the executable program code and any other components used during execution, such as sample JCL and messages.
- *Distribution libraries* are the data sets in which additional copies of these components are stored. The copy of High Level Assembler kept in the distribution libraries can be kept at a different service level than that in the target libraries. SMP/E can reconstruct the target libraries in whole or in part from the distribution libraries. Therefore, service or user modifications can be removed from the run-time copy of High Level Assembler

In many cases, the same system can be used as both a driving system and a target system. However, you may want to set up a clone of your system to use as a target system by making a separate IPL-able copy of the running system. The clone should include copies of all system libraries that SMP/E updates, copies of the SMP/E CSI data sets that describe the system libraries, and your PARMLIB and PROCLIB.

Some cases where two systems should be used include the following:

- When installing a new level of a product that is already installed, the new product will delete the old one. By installing onto a separate target system, you can test the new product while still keeping the old one in production.
- When installing a product that shares libraries or load modules with other products, the installation can disrupt the other products. Installing onto a test system or clone will allow you to assess these impacts without disrupting your production system.

Required and Optional Software for the Target System

This section describes the other products that must be installed on the target system in order to install and use High Level Assembler.

High Level Assembler runs on MVS with the required licensed programs listed in Table 3.

You should install the **minimum release listed or any subsequent release** for all required licensed programs your site needs.

Table 3. Required Programs

Required Licensed Program	Minimum Version Supported
MVS SP	Version 5 Release 1
OS/390	Version 1
SMP/E	Release 1.8.1 with PTF UR51070

The DESCRIPTION operand on the MCS ++FUNCTION statement is new for OS/390 Release 7 SMP/E. Previous releases of SMP/E will issue a informational message (GIM50050I) indicating the DESCRIPTION is being ignored, but processing for the FUNCTION will continue.

The following IFREQs identify the PTFs required for the previous releases of SMP/E in order to tolerate the DESCRIPTION operand on the ++FUNCTION statement:

```
++IF FMID(HMP1B00) THEN REQ(UR51068) /* OS/390 R5 SMP/E */.
++IF FMID(HMP1A00) THEN REQ(UR51067) /* OS/390 R3 SMP/E */.
++IF FMID(HMP1900) THEN REQ(UR51071) /* OS/390 R2 SMP/E */.
++IF FMID(HMP1802) THEN REQ(UR51070) /* SMP/E RELEASE 8.1 */.
```

Check with the IBM Support Center for any PTFs you might need to apply.

DASD Storage Required for the Target System

High Level Assembler libraries can reside on any currently supported DASD.

Table 4 lists the total space required for each type of library.

Table 4 (Page 1 of 2). Total DASD Space Required by High Level Assembler

Library Type	Total Space Required
Target	59 3390 Tracks

Table 4 (Page 2 of 2). Total DASD Space Required by High Level Assembler

Library Type	Total Space Required
Distribution	66 3390 Tracks

Table 5. Storage Requirements for SMP/E Work Data Sets

Library DDNAME	TYPE	ORG	RECFM	LRECL	No. of 3390 Trks	No. of DIR Blks
SMPWRK1	S	PDS	FB	80	24	111
SMPWRK2	S	PDS	FB	80	24	111
SMPWRK3	S	PDS	FB	80	28	111
SMPWRK4	S	PDS	FB	80	28	111
SMPWRK6	S	PDS	FB	80	28	111
SYSUT1	U	SEQ	--	--	75	0
SYSUT2	U	SEQ	--	--	75	0
SYSUT3	U	SEQ	--	--	75	0
SYSUT4	U	SEQ	--	--	75	0

Notes:

1. The data set sizes specified contain 15% extra space. You may wish to revise these numbers based on your plans for adding additional function or service.
2. IBM recommends use of system determined blocksizes for efficient DASD utilization for all non-RECFM U data sets. For RECFM U data sets, IBM recommends a blocksize of 32760, which is the most efficient from a performance and DASD utilization perspective.

   If you choose not to use system determined blocksizes, use the blocksizes and numbers of blocks specified to allocate the data sets. Data sets can be reblocked to a larger size. Please note that the maximum allowable blocksize will depend on the type of DASD on which the dataset will reside.
3. Abbreviations used for the data set type are:

   U Unique data set used by only the FMIDs listed. In order to determine the correct storage needed for this data set, this table provides all required information; no other tables (or program directories) need to be referenced for the data set size.

   S Shared data set used by more than the FMIDs listed. In order to determine the correct storage needed for this data set, the storage size given in this table needs to be added to other tables (perhaps in other program directories). If the data set already exists, it must have enough free space to accommodate the storage size given in this table.

   If you currently have a previous release of this product installed in these libraries, the installation of this release will delete the old one and reclaim the space used by the old release and any service that had been installed. You

can determine whether or not these libraries have enough space by deleting the old release with a dummy function, compressing the libraries, and comparing the space requirements with the free space in the libraries.

For more information on the names and sizes of the required data sets, please refer to Table 7 and Table 8.

The following table provides an estimate of the storage needed in the SMP/E data sets for High Level Assembler. The estimates must be added to those of any other programs and service being installed to determine the total additional storage requirements.

Table 6. Storage Requirements for SMP/E Data Sets

Library DDNAME	TYPE	ORG	RECFM	LRECL	No. of 3390 Trks	No. of DIR Blks
SMPMTS	S	PDS	FB	80	64	100
SMPPTS	S	PDS	FB	80	27	100
SMPSCDS	S	PDS	FB	80	15	100
SMPSTS	S	PDS	FB	80	15	100

The following tables list the target and distribution libraries (data sets) and their attributes required to install High Level Assembler. The storage requirements of High Level Assembler must be added to the storage required by other programs having data in the same data set (library).

Table 7. Storage Requirements for High Level Assembler Target Libraries

Library DDNAME	Member Type	Target Volume	TYPE	ORG	RECFM	LRECL	No. of 3390 Trks	No. of DIR Blks
SASMMOD1	LMOD	ANY	U	PDS	U	0	25	6
SASMSAM1	SAMPLE SOURCE PROC	ANY	U	PDS	FB	80	20	4
SASMMAC1	MACRO	ANY	U	PDS	FB	80	14	2

Table 8 (Page 1 of 2). Storage Requirements for High Level Assembler Distribution Libraries

Library DDNAME	TYPE	ORG	RECFM	LRECL	No. of 3390 Trks	No. of DIR Blks
AASMMOD1	U	PDS	U	0	32	18
AASMSAM1	U	PDS	FB	80	20	4

Table 8 (Page 2 of 2). Storage Requirements for High Level Assembler Distribution Libraries

Library DDNAME	TYPE	ORG	RECFM	LRECL	No. of 3390 Trks	No. of DIR Blks
AASMMAC1	U	PDS	FB	80	14	2

The following table list data sets that are not used by SMP/E, but are required for High Level Assembler to execute.

Table 9. Storage Requirements for High Level Assembler Non-SMP/E Data Sets

Data Set Name	TYPE	ORG	RECFM	LRECL	No. of 3390 Trks	No. of DIR Blks
SYSUT1	U	SEQ	--	--	75	0

The user may gain access to the supplied sample JCL after the SMP/E RECEIVE step in the install process. Alternatively, the user may copy the sample JCL directly from the installation tape into a temporary library. If this latter method is used then the dataset requirement is shown in the following table.

Table 10. Storage Requirements for temporary library

Data Set Name	TYPE	ORG	RECFM	LRECL	No. of 3390 Trks	No. of DIR Blks
JCL	U	SEQ	FB	80	20	4

FMIDs Deleted

Installing High Level Assembler will result in the deletion of the following FMIDs:

Table 11. FMIDs Deleted

Deleted FMID	Deleting FMID	Description
HMQ4100	HMQ4140	High Level Assembler Version 1 Release 1
HMQ4120	HMQ4140	High Level Assembler Version 1 Release 2
HMQ4130	HMQ4140	High Level Assembler Version 1 Release 3
JMQ412A	HMQ4140	High Level Assembler Toolkit Feature Version 1 Release 2
JMQ413A	HMQ4140	High Level Assembler Toolkit Feature Version 1 Release 3

If you do not wish to delete these FMIDs at this time, you must install High Level Assembler into separate SMP/E target and distribution zones.

Note: These FMIDs will not automatically be deleted from the Global Zone. Consult the SMP/E manuals for instructions on how to do this.

Installing with CBIPO or CBPDO

If you are installing High Level Assembler with an MVS Custom-Built Installation Process Offering (CBIPO) (5751-CS1), use the CBIPO Related Installation Materials (RIMs) provided with the CBIPO. The CBIPO RIMs will point you to specific sections of this book as needed.

If you are installing High Level Assembler with an MVS Custom-Built Product Delivery Offering (CBPDO) (5751-CS3), use the RCVPDO job in the CBPDO RIMLIB data set provided with the CBPDO to receive the product and service for this product. Any additional installation instructions should be obtained from the High Level Assembler documentation. However, before installing High Level Assembler, check with your IBM Support Center or use either Information/Access or SoftwareXcel Extended to see whether there is additional service information you need.

Planning to Use SMP/E

You must decide how to use SMP/E to install High Level Assembler. Some points to consider are:

1. Choose to use new global, target and distribution zones.
2. Choose to use existing global and new target and distribution zones.
3. Choose to use existing global, target and distribution zones.
4. Use supplied sample jobs or use SMP/E dialogs to RECEIVE, APPLY and ACCEPT. See Chapter 3, “Installing High Level Assembler on MVS” on page 21. This will install into existing global, target and distribution zones.

The REWORK level on the *++FUNCTION MCS* supplied in the *SMPMCS* file on the installation tape is *2000060*.

Planning to Change Installation Jobs

Each sample job has instructions on how it is to be modified, to meet the requirements and standards at your site. Refer to these standards as needed for determining the values for these parameters.

Selecting National Language Support

When installing High Level Assembler, you can choose which language you want to use for diagnostic messages. Languages available are English, German, Spanish and Japanese. English is also available in mixed case or uppercase.

The following combinations are possible:

English Uppercase Diagnostic messages and listing headings printed in uppercase English.

English Mixed Case Diagnostic messages and listing headings printed in mixed case English.

German Diagnostic messages in German and listing headings printed in mixed case English.

Japanese Diagnostic messages in Kanji and listing headings printed in uppercase English.

Spanish Diagnostic messages in Spanish and listing headings printed in mixed case English.

Program Support

This section describes the IBM support available for High Level Assembler.

Program Services

Contact your IBM representative for specific information about available program services.

Preventive Service Planning

Before installing High Level Assembler, you should review the current Preventive Service Planning (PSP) information. If you obtained High Level Assembler as part of a CBPDO, there is HOLDDATA and PSP information included on the CBPDO tape.

If you obtained High Level Assembler on a product tape, or if the CBPDO is more than two weeks old when you install it, you should contact the IBM Support Center or use S/390 SoftwareXcel to obtain the current “PSP Bucket.”

PSP Buckets are identified by UPGRADEs, which specify product levels, and SUBSETs, which specify the FMIDs for a product level. The UPGRADE and SUBSET values for High Level Assembler are:

Table 12. PSP Upgrade and Subset ID

UPGRADE	SUBSET	Description
HLASM140	HMQ4140	HLASM MVS

Statement of Support Procedures

Report any difficulties you have using this program to your IBM Support Center. If an APAR is required, the Support Center will provide the address to which any needed documentation can be sent.

Table 13 on page 12 identifies the component IDs (COMPID) for High Level Assembler.

Table 13. Component IDs

FMID	COMPID	Component Name	RETAIN Release
HMQ4140	569623400	MVS HIGH LEVEL ASM	140

Program and Service Level Information

This section identifies the program and any relevant service levels of High Level Assembler. The program level refers to the APAR fixes incorporated into the program. The service level refers to the PTFs integrated. Information about the cumulative service tape is also provided.

Program Level Information

A list of APAR fixes against previous releases of High Level Assembler that have been incorporated into this release is shown in Appendix C, "High Level Assembler Service" on page 207.

Service Level Information

No PTFs against this release of High Level Assembler have been incorporated into the product tape.

Cumulative Service Tape

A cumulative service tape, containing PTFs not incorporated into this release, might be included with this program. Installation instructions for cumulative service tapes can be found in the SMP/E publications.

If you received this product as part of a CBPDO or a ProductPac, PTFs not incorporated into this release are provided on the tape, and a separate cumulative service tape will not be provided.

Publications Useful during Installation

The publications listed in Table 14 may be useful during the installation of High Level Assembler. To order copies, contact your IBM representative.

Table 14. Publications Useful During Installation

Publication Title	Form Number
OS/390 MVS Initialization and Tuning Reference	SC28-1752
OS/390 MVS JCL Reference	GC28-1757
OS/390 SMP/E Messages and Codes	SC28-1738
OS/390 SMP/E Reference	SC28-1806
OS/390 SMP/E User's Guide	SC28-1740

Chapter 2. Planning for Customizing High Level Assembler on MVS

This chapter provides information for planning the customization of High Level Assembler on MVS. It includes:

- Deciding whether and what to customize
- Impact of making SMP/E use High Level Assembler as its assembler
- Planning to customize the IBM-supplied default option values
- Planning to customize the IBM-supplied default user exit
- Planning to customize the cataloged procedures
- Planning to install High Level Assembler into a link pack area

Deciding Whether and What to Customize

You need to consider whether the IBM-supplied values that come with High Level Assembler suit the needs of your site. These values control such features as:

- SMP/E assembler use
- Assembler options
- Default file names
- User exits
- Sample procedures

Make sure that High Level Assembler serves the needs of the application programmers at your site. Confer with them while you evaluate the customization options for High Level Assembler, particularly those concerning High Level Assembler options that are also available to the application programmers. This ensures that the modifications you make, best support the application programs being developed at your site.

The information in this chapter helps you plan your customization. See Chapter 4, “Customizing High Level Assembler on MVS” on page 29 for the actual customization procedure.

Impact of making SMP/E use High Level Assembler

If you plan to have an SMP/E OPTIONS entry point to a UTILITY entry, which has High Level Assembler as the assembler to be called by SMP/E, then High Level Assembler must reside in an Authorized Program Facility (APF) authorized library.

Planning to Customize High Level Assembler Options

High Level Assembler can be customized with a large range of options and DDNAMES for users. When customized they become the default options and DDNAMES.

Why Do It

The High Level Assembler options and DDNAMES should be reviewed to assess the required defaults for your site. Worksheets are provided for planning purposes.

Choices to Make Now

The following worksheets help you plan and code the options and DDNAMES appropriate for your site. To complete the worksheets, fill in the **Enter Selection** column.

Table 15 (Page 1 of 2). Worksheet: Options

Option	Enter Selection	IBM-Supplied Default	Description
ADATA	_____	NO	page 171
ADEXIT	_____	no exit specified	page 171
ALIGN	_____	YES	page 172
ALIGNWARN	_____	YES	page 172
ASA	_____	NO	page 173
BATCH	_____	YES	page 173
CODEPAGE	_____	047C	page 173
COMPAT	_____	NO	page 174
CONTWARN	_____	YES	page 175
DBCS	_____	NO	page 176
DECK	_____	NO	page 176
DELETE	_____	No options deleted	page 176
DSECT	_____	NO	page 177
DXREF	_____	YES	page 178
ESD	_____	YES	page 178
FLAG	_____	0	page 178
FOLD	_____	NO	page 179
GOFF	_____	NO	page 179
GOFFADATA	_____	NO	page 180
IMPLENWARN	_____	NO	page 180
INEXIT	_____	no exit specified	page 180
INFO	_____	NO	page 181
LANGUAGE	_____	EN	page 181
LIBEXIT	_____	no exit specified	page 182
LIBMAC	_____	NO	page 182
LIMIT	_____	NO	page 183
LINECOUNT	_____	60	page 183
LIST	_____	121	page 183
MAP	_____	YES	page 184
MXREF	_____	SOURCE	page 185

Table 15 (Page 2 of 2). Worksheet: Options

Option	Enter Selection	IBM-Supplied Default	Description
OBJECT	_____	YES	page 185
OBJEXIT	_____	no exit specified	page 186
OPTABLE	_____	UNI	page 186
PAGE0WARN	_____	NO	page 187
PCONTROL	_____	NO	page 188
PESTOP	_____	NO	page 190
PROFILE	_____	NO	page 190
PROFMEM	_____	ASMAPROF	page 190
PRTEXIT	_____	no exit specified	page 191
PUSHWARN	_____	YES	page 191
RA2	_____	NO	page 192
RECORDINFO	_____	YES	page 192
RENT	_____	NO	page 193
RLD	_____	YES	page 193
RXREF	_____	YES	page 193
SIZE	_____	MAX	page 194
STORAGE	_____	BELOW	page 195
SUBSTRWARN	_____	NO	page 196
SYSPARMV	_____	none specified	page 196
TERM	_____	NO	page 196
TEST	_____	NO	page 197
THREAD	_____	YES	page 197
TRANSLATE	_____	NO	page 198
TRMEXIT	_____	no exit specified	page 198
USING0WARN	_____	YES	page 198
WARN	_____	15	page 199
XREF	_____	(SHORT,UNREFS)	page 201

Table 16. Worksheet: DDNAMES

DDNAME	Enter Selection	IBM-Supplied Default	Syntax Description
ADATA	_____	SYSADATA	page 203
IN	_____	SYSIN	page 203
LIB	_____	SYSLIB	page 204
LIN	_____	SYSLIN	page 204
OPTN	_____	ASMAOPT	page 204
PRINT	_____	SYSPRINT	page 204
PUNCH	_____	SYSPUNCH	page 205
TERM	_____	SYSTERM	page 205
UT1	_____	SYSUT1	page 205

Planning to Customize the User Exits

Several exits are provided with the High Level Assembler, they are described in *HLASM Programmer's Guide*.

Note: If User Exits from a previous release are being used, these exits may need to be reassembled for this release.

Planning to Customize Cataloged Procedures

High Level Assembler provides four cataloged procedures to:

- Assemble a program
- Assemble and link-edit a program
- Assemble, link-edit and run a program
- Assemble a program and use the loader to run it

Why Do It

Providing procedures that can assemble, link-edit, or link-and-go allow the user to simply provide their assembler source, and select the appropriate procedure and run their job. It minimizes the amount of user JCL required and provides a standard convention to process assembler source.

These procedures provide the user with standard JCL that is tailored to your installation requirements. The user does not need to keep track of JCL changes.

The cataloged procedures distributed by IBM execute in as many MVS/ESA environments as possible. You should review these procedures and optimize them for your environment.

In general, installation conventions stipulate the options that you include in the PARM, UNIT, and SPACE parameters of the cataloged procedures.

Choices to Make Now

Four standard procedures are supplied; refer to Table 17 for their descriptions.

Table 17. Supplied Procedures

Procedure	Description
ASMAC	Assembles only
ASMACL	Assembles and link-edits program
ASMACLG	Assembles, link-edits program and runs program
ASMACG	Assembles and uses the loader to run program

If the Release 1 of High Level Assembler has been used, users might have become accustomed to using the old procedures names of HLASMC, HLASMCG, HLASCLG and HLASMCL. In order to make the transition from this previous release, in the SASMSAM1 library, these old procedure names appear as aliases for the sample supplied procedures, as shown in Table 18.

Table 18. New and Old Procedure Names

New Name	Old Name
ASMAC	HLASMC
ASMACL	HLASMCL
ASMACLG	HLASMCLG
ASMACG	HLASMCG

An Example

Figure 1 shows JCL that invokes a sample procedure to assemble a small assembler program.

```
//ASMAPTST JOB  <JOB CARD PARAMETERS>
//*
//********************************************************************
//*   Licensed Materials - Property of IBM                           *
//*                                                                  *
//*   5696-234  5647-A01                                             *
//*                                                                  *
//*   (C) Copyright IBM Corp. 1992, 2000. All Rights Reserved.       *
//*                                                                  *
//*   US Government Users Restricted Rights - Use,                   *
//*   duplication or disclosure restricted by GSA ADP                *
//*   Schedule Contract with IBM Corp.                               *
//*                                                                  *
//********************************************************************
//********************************************************************
//*                                                                  *
//* ASMAPTST JOB                                                     *
//*                                                                  *
//* THIS JCL WILL INVOKE A SUPPLIED PROCEDURE TO ASSEMBLE A SMALL    *
//* PROGRAM.                                                         *
//*                                                                  *
//*  CAUTION: THIS IS NEITHER A JCL PROCEDURE NOR A COMPLETE JOB.    *
//*  BEFORE USING THIS JOB, YOU WILL HAVE TO MAKE THE FOLLOWING      *
//*  MODIFICATIONS:                                                  *
//*                                                                  *
//*  1. CHANGE THE JOB CARD TO MEET YOUR SYSTEM REQUIREMENTS         *
//*  2. CHANGE #hlq TO THE HIGH LEVEL QUALIFIER FOR YOUR TARGET      *
//*     LIBRARY.                                                     *
//*                                                                  *
//********************************************************************
//ASMAPTST EXEC PROC=ASMAC
//SYSPRINT DD   SYSOUT=*
//STEPLIB  DD  DSN=#hlq.SASMMOD1,
//             DISP=SHR
//SYSIN    DD  *
TEST       CSECT
           XR   15,15
           BR   14
           END
/*
```

Figure 1. Use of sample procedure

This example is supplied as ASMAPTST in library SASMSAM1.

Planning to Install in the Link Pack Area

This section describes the reasons why you should place High Level Assembler modules in common storage, and lists the modules that can you can place there.

Why Do It

You might want to make some load modules resident in a link pack area in order to minimize the search path length when High Level Assembler is run.

All assembler load modules are reentrant and eligible for placement in the link pack area (LPA) and the modifiable link pack area (MLPA). The LPA and the MLPA are areas of storage that are shared by all users. By placing modules in these areas, you reduce the amount of storage needed in the user region.

By sharing the information you can:

- Save on real storage.
- Reduce I/O activity to the LOADLIB and/or page data sets.
- Reduce elapsed time of jobs.

Choices to Make Now

Table 19 is a list of module names, their link-edit attributes and approximate sizes.

Table 19 (Page 1 of 2). Module Information

Module name	Amode	Rmode	Other	Linkedit	Attributes	Approx. Size
ASMA90	ANY	24	RENT	REUS	REFR	7K
ASMA93	31	ANY	RENT	REUS	REFR	335K
ASMADOPT	31	ANY	RENT	REUS	REFR	2K
ASMAMUE	31	ANY	RENT	REUS	REFR	20K
ASMAMDE	31	ANY	RENT	REUS	REFR	24K
ASMAMES	31	ANY	RENT	REUS	REFR	24K
ASMAMJP	31	ANY	RENT	REUS	REFR	23K
ASMAOESA	31	ANY	RENT	REUS	REFR	33K
ASMAOXA	31	ANY	RENT	REUS	REFR	24K
ASMAO370	31	ANY	RENT	REUS	REFR	24K
ASMAODOS	31	ANY	RENT	REUS	REFR	14K
ASMALTAS	31	ANY	RENT	REUS	REFR	1K
ASMAINFO	31	ANY	RENT	REUS	REFR	2K
ASMAOEME	31	ANY	RENT	REUS	REFR	41K
ASMA0474	31	ANY	RENT	REUS	REFR	2K
ASMA0475	31	ANY	RENT	REUS	REFR	2K
ASMA0476	31	ANY	RENT	REUS	REFR	2K
ASMA0477	31	ANY	RENT	REUS	REFR	2K
ASMA0478	31	ANY	RENT	REUS	REFR	2K
ASMA0479	31	ANY	RENT	REUS	REFR	2K

Table 19 (Page 2 of 2). Module Information

Module name	Amode	Rmode	Other	Linkedit	Attributes	Approx. Size
ASMA047A	31	ANY	RENT	REUS	REFR	2K
ASMA047B	31	ANY	RENT	REUS	REFR	2K
ASMA047C	31	ANY	RENT	REUS	REFR	2K
ASMADOP	31	ANY	RENT	REUS	REFR	31K

If you prefer to move some or all of the load modules, move them with a USERMOD that has a `++MOVE` statement.

Placing Assembler Modules in the LPA and MLPA: Refer to the following publications for specific instructions about placing assembler modules into the LPA and MLPA:

MVS/ESA System Generation
OS/390 MVS Initialization and Tuning Reference

Chapter 3. Installing High Level Assembler on MVS

This chapter describes the installation method and the step-by-step procedures to install and to activate the functions of High Level Assembler.

Please note the following:

- If you want to install High Level Assembler into its own SMP/E environment, consult the SMP/E manuals for instructions on creating and initializing the SMPCSI and the SMP/E control data sets.
- Sample jobs have been provided to help perform some or all of the installation tasks. The SMP/E jobs assume that all DDDEF entries required for SMP/E execution have been defined in the appropriate zones.
- The SMP/E dialogs may be used instead of the sample jobs to accomplish the SMP/E installation steps.

Overview of Installation

You install this release of High Level Assembler by using the SMP/E RECEIVE, APPLY, and ACCEPT commands. This may be accomplished by using SMP/E dialogs or running batch jobs.

Modifying High Level Assembler Jobs

The installation jobs that IBM provides do not necessarily follow the conventions (such as file-naming conventions) for your site. You might need to modify them. Descriptions of possible modifications appear in the comments in the sample jobs.

If Using a CBPDO

If you obtained High Level Assembler as part of a CBPDO, you can use the RCVPDO job found in the CBPDO RIMLIB data set to RECEIVE the High Level Assembler FMID as well as any service, HOLDDATA, or preventive service planning (PSP) information included on the CBPDO tape. For more information, refer to the documentation included with the CBPDO. For more information, refer to *MVS CBPDO Memo to User Extension*, which is included with the CBPDO.

Installation Checklist

Table 20 lists the steps and associated jobs to install High Level Assembler. The remaining sections in this chapter describe each step. You can use Table 20 as a checklist.

Table 20. Summary of Steps for Installing High Level Assembler

Step	Description	Installation Job	page
_1	SMP/E Considerations for Installing High Level Assembler		22
_2	Change the Sample JCL from the Product Tape	ASMACOPY	23
_3	Perform SMP/E Receive	ASMARECV	24
_4	Set up Target and Distribution Libraries	ASMALLOC	25
_5	Create DDDEF Entries	ASMDDDEF	25
_6	Perform SMP/E Apply	ASMAAPLY	25
_7	Run the Installation Verification Program	ASMAIVP	26
_8	Perform SMP/E Accept	ASMAACPT	26

Step 1: SMP/E Considerations for Installing High Level Assembler

This release of High Level Assembler is installed using the SMP/E RECEIVE, APPLY, and ACCEPT commands. The SMP/E dialogs may be used to accomplish the SMP/E installation steps.

SMP/E Environment

All SMP/E installation jobs provided assume that all necessary DD statements for the execution of SMP/E are defined using DDDEFs.

Sample jobs are provided to assist you in installing High Level Assembler. After the RECEIVE step has been completed, the sample jobs can be found in SMPTLIB: **IBM.HMQ4140.F1**. Make a copy of these jobs in your own library and modify them to use during the installation of High Level Assembler. The sample jobs are:

ASMARECV Sample RECEIVE job

ASMALLOC Sample job to allocate target and distribution libraries

ASMDDDEF Sample job to define SMP/E DDDEFs

ASMAAPLY Sample APPLY job

ASMAIVP Sample install verification job

ASMAACPT Sample ACCEPT job

In the sample SMP/E jobs provided, the name of the SMP/E CSI is `#GLOBALCSI`. The global zone name in the SMP/E CSI is `GLOBAL`. The distribution zone name is `#DZONE`. The target zone name is `#TZONE`. The sample jobs should be updated to reflect the CSI and zone names used at your installation.

SMP/E Options Subentry Values

The recommended values for some SMP/E CSI subentries are shown in Table 21. Use of values lower than these may result in failures in the installation process. DSSPACE is a subentry in the GLOBAL options entry. PEMAX is a subentry of the GENERAL entry in the GLOBAL options entry. Refer to the SMP/E manuals for instructions on updating the global zone.

Table 21. SMP/E Options Subentry Values

SUB-ENTRY	Value	Comment
DSSPACE	(400,200,24)	Space Allocation for SMPTLIB data sets
PEMAX	SMP/E Default	IBM recommends using the SMP/E default for PEMAX.

Step 2: Change the Sample JCL from the Product Tape

This step can be bypassed if you plan to acquire the sample jobs from the SMPTLIB: **IBM.HMQ4140.F1** after the Receive Step.

Sample installation jobs are provided on the distribution tape to help you install High Level Assembler. The following sample JCL will copy the High Level Assembler jobs from the tape. Add a job card and modify the parameters in boldface to uppercase values to meet your site's requirements before submitting.

```
//COPY   EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//IN     DD DSN=IBM.HMQ4140.F1,UNIT=#tape,VOL=SER=MQ4140,
//       DISP=(OLD,KEEP),LABEL=(2,SL)
//OUT    DD DSN=#hlq.JCL,
//       DISP=(NEW,CATLG,DELETE),
//       RECFM=FB,LRECL=80,BLKSIZE=0,
//       SPACE=(8800,(120,40,4)),
//       UNIT=#unit,
//       VOL=SER=#dasdvol
//SYSUT3 DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN DD *
  COPY INDD=IN,OUTDD=OUT
/*
```

Figure 2. Load Installation Jobs

Where **#tape** is the unit value matching the product tape or cartridge, **#hlq.JCL** is the name of the data set where the sample jobs will reside, **#dasdvol** is the volume serial of the DASD device where the data set will reside, and **#unit** is the DASD unit type of the volume.

This sample job is also supplied as sample ASMACOPY in SMPTLIB: IBM.HMQ4140.F1.

Step 3: Perform SMP/E Receive

There are two methods of receiving High Level Assembler:

- The interactive capability of SMP/E described in "Method 1: Receive Using SMP/E Panels."
- The batch procedure described in "Method 2: Receive Using Batch Jobs."

Method 1: Receive Using SMP/E Panels

Select the Command Generation option from the SMP/E Primary Option menu.

Supply the following information as you progress through subsequent command generation panels for the RECEIVE command:

Name of your CSI	_____
Zone name	Global
HOLDDATA	No
All	No
Select	Yes
FMIDs you are installing (base)	_____
FMIDs you are installing (component)	_____
Data set name	_____
Volume serial	_____
Unit	_____
Label	_____
File	_____

Expected Return Codes and Messages: This should issue a return code of zero and no error messages.

Method 2: Receive Using Batch Jobs

Edit and submit job ASMARECV to receive High Level Assembler. Consult the instructions in the sample job for more information.

NOTE: if you obtained High Level Assembler as part of a CBPDO, you can use the RCVPDO job found in the CBPDO RIMLIB data set to RECEIVE the High Level Assembler FMIDs as well as any service, HOLDDATA, or preventive service planning (PSP) information included on the CBPDO tape. For more information, refer to the documentation included with the CBPDO.

Expected Return Codes and Messages: This should issue a return code of zero and no error messages.

Step 4: Set up Target and Distribution Libraries

Edit and submit job ASMALLOC to allocate the SMP/E target and distribution libraries for High Level Assembler. Consult the instructions in the sample job for more information.

Expected Return Codes and Messages: This should issue a return code of zero and no error messages.

Step 5: Create DDDEF Entries

Edit and submit job ASMDDDEF to create DDDEF entries for the SMP/E target and distribution libraries for High Level Assembler. Consult the instructions in the sample job for more information.

Expected Return Codes and Messages: This should issue a return code of zero and no error messages.

After this job, the SYSLIB concatenation needs to be updated in the target and distribution zones.

- SASMMAC1 and AASMMAC1 libraries need to be added to the SYSLIB concatenation in the target zone.
- AASMMAC1 library needs to be added to the SYSLIB concatenation in the distribution zone.

Step 6: Perform SMP/E Apply

There are two methods of applying High Level Assembler:

- The interactive capability of SMP/E described in “Method 1: Apply Using SMP/E Panels.”
- The batch procedure described in “Method 2: Apply Using Batch Jobs” on page 26.

Method 1: Apply Using SMP/E Panels

1. Select the Command Generation option from the SMP/E Primary Option menu.

   You will be asked to supply some of the information above as you progress through subsequent command generation panels for APPLY.

2. Perform an APPLY CHECK before the APPLY. Examine the output from the APPLY CHECK run.

   Expected Return Codes and Messages: This should issue a return code of zero and no error messages.

3. If it shows no conflict, rerun the APPLY with CHECK=NO.

   Expected Return Codes and Messages: This should issue a return code of zero and no error messages.

Method 2: Apply Using Batch Jobs

1. Perform an Apply with Check

   Use the sample job ASMAAPLY to apply High Level Assembler.

   The installation job ASMAAPLY invokes SMP/E to apply High Level Assembler. ASMAAPLY performs an APPLY with the CHECK option to check for possible conflicts in the target zone.

   Edit and submit sample job ASMAAPLY to perform an SMP/E APPLY CHECK. Consult the instructions in the sample job for more information.

   To receive the full benefit of the SMP/E Causer SYSMOD Summary Report, do *not* bypass the following on the APPLY CHECK: PRE, ID, REQ, and IFREQ. This is because the SMP/E root cause analysis identifies the cause only of **ERRORS** and not of **WARNINGS** (SYSMODs that are bypassed are treated as warnings, not errors, by SMP/E).

   Expected Return Codes and Messages: This should issue a return code of zero and no error messages.

2. Perform an Apply

   Edit sample job ASMAAPLY to remove CHECK operand and resubmit.

   Note: The GROUPEXTEND operand indicates that SMP/E accept all requisite SYSMODs. The requisite SYSMODS might be applicable to other functions.

   Expected Return Codes and Messages: This should issue a return code of zero and no error messages.

Step 7: Run the Installation Verification Program

1. Assemble the Verification Program

   A sample job, ASMAIVP, is provided to verify that the product has installed correctly. This job assembles the source statements from member ASMASAMP. The assembly of ASMASAMP verifies that the product is installed and functions correctly.

   Edit and submit sample job ASMAIVP. Consult the instructions in the sample job for more information.

2. Run the Verification Program (optional)

   The program being assembled is not intended to be run; however, if it is run, it sets a return code of zero and returns to the caller.

Expected Return Codes and Messages: This should issue a return code of zero and no error messages.

Step 8: Perform SMP/E Accept

To permanently install High Level Assembler, use SMP/E ACCEPT processing. You should accept High Level Assembler *before* you apply any user modifications or install any maintenance because *the SMP/E RESTORE command restores High Level Assembler only to the level of the last version you accept*.

There are two methods of accepting High Level Assembler:

- The interactive capability of SMP/E described in "Method 1: Accept Using SMP/E Panels" on page 27.
- The batch procedure described in "Method 2: Accept Using Batch Jobs."

Method 1: Accept Using SMP/E Panels

1. Select the Command Generation option from the SMP/E Primary Option menu.

   Supply the information requested as you progress through subsequent command generation panels for ACCEPT.

2. Perform an ACCEPT CHECK before running the ACCEPT command. Examine the output from the ACCEPT CHECK run.

   Expected Return Codes and Messages: This should issue a return code of zero and no error messages.

3. Perform an ACCEPT.

   If the ACCEPT CHECK run shows no conflict, rerun the ACCEPT with CHECK=NO.

   Expected Return Codes and Messages: This should issue a return code of zero and no error messages.

Method 2: Accept Using Batch Jobs

1. Perform an ACCEPT with CHECK

   The job ASMAACPT performs an ACCEPT with the CHECK option to check for possible conflicts in the distribution zone.

   Edit and submit sample job ASMAACPT to perform an SMP/E ACCEPT CHECK for High Level Assembler. Consult the instructions in the sample job for more information.

   To receive the full benefit of the SMP/E Causer SYSMOD Summary Report, do *not* bypass the following on the APPLY CHECK: PRE, ID, REQ, and IFREQ. This is because the SMP/E root cause analysis identifies the cause only of **ERRORS** and not of **WARNINGS** (SYSMODs that are bypassed are treated as warnings, not errors, by SMP/E).

   Expected Return Codes and Messages:. This should issue a return code of zero and no error messages.

2. Perform an ACCEPT

   When this job is successful, rerun this job without the CHECK option.

   Before using SMP/E to load new distribution libraries, it is recommended that you set the ACCJCLIN indicator in the distribution zone. This will cause entries produced from JCLIN to be saved in the distribution zone whenever a SYSMOD containing inline JCLIN is ACCEPTed. For more information on the ACCJCLIN indicator, see the description of inline JCLIN in the SMP/E manuals.

   Note: The GROUPEXTEND operand indicates that SMP/E accept all requisite SYSMODs. The requisite SYSMODS might be applicable to other functions.

   Expected Return Codes and Messages: This should issue a return code of zero and no error messages.

Chapter 4. Customizing High Level Assembler on MVS

You can customize, or modify, High Level Assembler only after installing the product. This chapter includes:

- SMP/E Considerations
- Customization Checklist
- Customization Steps

SMP/E Considerations

The following SMP/E information should be considered prior to commencing any customizing task.

1. Install High Level Assembler as described in Chapter 3, “Installing High Level Assembler on MVS” on page 21.
2. Apply the USERMODs to the target libraries, *but do not accept them into the distribution libraries.*
3. Use SMP/E RESTORE to remove a USERMOD before you apply service to the modules it changes.
4. Reapply the USERMOD after successful installation of High Level Assembler service.
5. Sample jobs are provided to assist you in customizing High Level Assembler. The sample jobs can be found in the SASMSAM1 library. The sample jobs are:

   ASMAASM Sample SMP/E job to change CSI

   ASMAOPTS A sample SMP/E USERMOD job to change default options

   ASMASTD Sample job to copy procedures

   ASMAIEV Sample SMP/E USERMOD job to create IEV90 Alias

   In the sample SMP/E jobs provided, the name of the SMP/E CSI is `#GLOBALCSI`. The global zone name in the SMP/E CSI is `GLOBAL`. The distribution zone name is `#DZONE`. The target zone name is `#TZONE`. The sample jobs should be updated to reflect the CSI and zone names used at your installation.

Customization Checklist

Table 22 on page 30 lists the steps and associated jobs to customize High Level Assembler. The remaining sections in this chapter describe each step. You can use Table 22 on page 30 as a checklist.

Table 22. Summary of Steps for customizing High Level Assembler

Step	Description	Customizing Job	page
__ 1	Change SMP/E to use High Level Assembler	ASMAASM	30
__ 2	Customize user exits.	_	32
__ 3	Change default options and DDNAMES.	ASMAOPTS	32
__ 4	Place Procedures in Procedure Library.	ASMASTD	35
__ 5	Place High Level Assembler into link pack area.	_	41
__ 6	Assembler H Migration (optional)	ASMAIEV	41

Customization Steps

Depending on your requirements not all of the following steps need to be performed. For example, if you do not wish to change the supplied default options and DDNAMES, you can bypass that step.

Step 1: Change SMP/E to use High Level Assembler

With the sample copy job shown in Figure 3 on page 31 you can update SMP/E with the name of the assembler to be called by SMP/E.

Depending on how your SMP/E system is configured, you must specify IEV90 or ASMA90 as the name of the program to be called in the SMP/E Utility entry.

The user should also consider changing other existing OPTIONS entries. That is, alter the ASM subentry of each OPTIONS entry to point to the new UTILITY entry for High Level Assembler.

If you change the OPTIONS entry to use High Level Assembler as the assembler in SMP/E then the High Level Assembler executable code must reside in an Authorized Program Facility (APF) authorized library.

Edit and submit job ASMAASM to change the OPTIONS entry. Consult the instructions in the sample job for more information. The sample job is shown in Figure 3 on page 31.

```
//ASMAASM  JOB  <JOB CARD PARAMETERS>
//*
//*********************************************************************
//*   Licensed Materials - Property of IBM                            *
//*                                                                   *
//*   5696-234  5647-A01                                              *
//*                                                                   *
//*   (C) Copyright IBM Corp. 1992, 2000. All Rights Reserved.        *
//*                                                                   *
//*   US Government Users Restricted Rights - Use,                    *
//*   duplication or disclosure restricted by GSA ADP                 *
//*   Schedule Contract with IBM Corp.                                *
//*                                                                   *
//*********************************************************************
//*********************************************************************
//*                                                                   *
//* ASMAASM  JOB                                                      *
//*                                                                   *
//* THIS JCL WILL ALTER THE UTILITY NAME FOR THE ASSEMBLER.           *
//*                                                                   *
//*  CAUTION: THIS IS NEITHER A JCL PROCEDURE NOR A COMPLETE JOB.     *
//*  BEFORE USING THIS JOB, YOU WILL HAVE TO MAKE THE FOLLOWING       *
//*  MODIFICATIONS:                                                   *
//*                                                                   *
//*  1. CHANGE THE JOB CARD TO MEET YOUR SYSTEM REQUIREMENTS          *
//*  2. CHANGE #globalcsi TO THE DATASET NAME OF YOUR GLOBAL          *
//*     CSI DATASET.                                                  *
//*                                                                   *
//*********************************************************************
//ALTER    EXEC PGM=GIMSMP,REGION=4096K
//SMPCSI   DD DSN=#globalcsi,
//            DISP=SHR
//SMPOUT   DD SYSOUT=*
//SMPRPT   DD SYSOUT=*
//SMPLIST  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SMPCNTL  DD *
 SET BDY(GLOBAL).              /* INIT GLOBAL ZONE CSI               */
 UCLIN .
   ADD OPTIONS(GBLOPT)         /* ADD AN OPTIONS ENTRY               */
       ASM(HLASM)              /* SPECIFY NAME OF ASSEMBLER          */
       .
   ADD UTILITY(HLASM)          /* ADD UTILITY ENTRY FOR ASSEMBLER    */
                               /* DEFINED IN ABOVE OPTIONS ENTRY     */
       NAME(ASMA90)            /* NAME OF PROGRAM TO BE INVOKED      */
       RC(4)                   /* RETURN CODE THRESHOLD              */
       PRINT(SYSPRINT)         /* DDNAME FOR SYSPRINT OUTPUT         */
       PARM(NOOBJ,DECK,XREF(SHORT,UNREFS))   /* DEFAULTS             */
       .
 ENDUCL.
 LIST ALLZONES.                /* LIST ZONE INFORMATION              */
/*
```

Figure 3. Change SMP/E to use High Level Assembler

Step 2: Customize user exits

For information concerning user exits; see *HLASM Programmer's Guide*. The following user exits are provided:

- Source exit
- ADATA exit
- Listing exit

In order to assist users the following sample source is provided.

- ASMAXINV

  This is a sample source exit.

- ASMAXPRT

  This is a sample listing exit.

- To support the ADATA exit the following are provided

  1. ASMAXFSK

     Sample skeleton SYSADATA filter routine

  2. ASMAXFMT

     Sample filter management table

  3. ASMAXFLU

     Sample filter module to dump ADATA records

  4. ASMAXADT

     Sample ADATA exit to load filter routine

Step 3: Change Default Options and DDNAMES

The job ASMAOPTS supplies a USERMOD to SMP/E to create a modified options module ASMADOPT.

Before you run ASMAOPTS:

1. If you have changed the SMP/E OPTIONS entry to use High Level Assembler as the assembler in SMP/E then ensure that the High Level Assembler product resides in an Authorized Program Facility (APF) authorized library. Refer to "General Rules for coding the ASMADOPT ASSEMBLE file" on page 81 for more information on how to enter the options into the sample job.
2. Review the worksheets in Table 15 on page 14 and Table 16 on page 16, and enter these options and DDNAMES values into the sample job ASMAOPTS.
3. Edit and submit job ASMAOPTS to change the default options. Consult the instructions in the sample job for more information as shown in Figure 4 on page 33 and change where required.

```
//ASMAOPTS JOB  <JOB CARD PARAMETERS>
//*
//********************************************************************
//*   Licensed Materials - Property of IBM                           *
//*                                                                  *
//*   5696-234  5647-A01                                             *
//*                                                                  *
//*   (C) Copyright IBM Corp. 1992, 2000. All Rights Reserved.       *
//*                                                                  *
//*   US Government Users Restricted Rights - Use,                   *
//*   duplication or disclosure restricted by GSA ADP                *
//*   Schedule Contract with IBM Corp.                               *
//*                                                                  *
//********************************************************************
//********************************************************************
//*                                                                  *
//* ASMAOPTS JOB                                                     *
//*                                                                  *
//* THIS JCL IS A USERMOD TO ALLOW CHANGES TO THE HIGH LEVEL         *
//* ASSEMBLER OPTIONS MODULE (ASMADOPT).                             *
//*                                                                  *
//*  CAUTION: THIS IS NEITHER A JCL PROCEDURE NOR A COMPLETE JOB.    *
//*  BEFORE USING THIS JOB, YOU WILL HAVE TO MAKE THE FOLLOWING      *
//*  MODIFICATIONS:                                                  *
//*                                                                  *
//*  1. CHANGE THE JOB CARD TO MEET YOUR SYSTEM REQUIREMENTS         *
//*  2. CHANGE #globalcsi TO THE DATASET NAME OF YOUR GLOBAL         *
//*     CSI DATASET.                                                 *
//*  3. CHANGE #hlq TO THE HIGH LEVEL QUALIFIER USED FOR THE TARGET  *
//*     LIBRARIES.                                                   *
//*  4. CHANGE #date ON THE ++USERMOD STATEMENT TO BE THE CURRENT    *
//*     DATE. IT IS IN THE FORM OF YYYYDDD.                          *
//*  5. CHANGE #tzone TO THE NAME OF YOUR TARGET ZONE.               *
//*                                                                  *
//*  NOTES:                                                          *
//*                                                                  *
//*  1. IN THE USERMOD BELOW, AS AN EXAMPLE, THE ADATA OPTION IS SET *
//*     ON AND BECOMES AN INSTALLATION DEFAULT. ALSO THE DDNAME OF   *
//*     THE ASSEMBLER WORK FILE IS CHANGED TO SYSUT2 AS AN           *
//*     INSTALLATION DEFAULT.                                        *
//*                                                                  *
//********************************************************************
```

Figure 4 (Part 1 of 2). Changing Default Options and DDNAMES

```
//USERMOD  EXEC PGM=GIMSMP,REGION=4096K
//SMPCSI   DD DSN=#globalcsi,
//            DISP=SHR
//SMPOUT   DD SYSOUT=*
//SMPRPT   DD SYSOUT=*
//SMPLIST  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSLIB   DD  DSN=#hlq.SASMMAC1,
//             DISP=SHR
//STEPLIB  DD  DSN=#hlq.SASMMOD1,
//             DISP=SHR
//SMPPTFIN DD  *
++USERMOD(ML00001) REWORK(#date).     1
++VER(Z038) FMID(HMQ4140).            2
++SRC(ASMADOPT) DISTLIB(AASMSAM1).
    ASMAOPT ADATA=YES                 3
    ASMADD  UT1=SYSUT2                4
    END
/*
//SMPCNTL  DD  *
   SET BDY(GLOBAL).
   RECEIVE S(ML00001) SYSMODS.
   SET BDY(#tzone).
   APPLY S(ML00001) ASSEM.            5
/*
```

Figure 4 (Part 2 of 2). Changing Default Options and DDNAMES

To change the default options and DDNAMES:

1. In area **1** use the REWORK option if you change the USERMOD frequently.

   Specify REWORK if the USERMOD name will be reused every time and you do not prefer to use the REJECT command before receiving the USERMOD.

   Every time you re-apply the USERMOD, increment the value of REWORK using the form of *yyyyddd*, where *yyyy* is the year (for example, 2000) and *ddd* is the day of the year (for example, 083 is the 24th of March). For more information refer to the *OS/390 SMP/E Reference*.

   If you do not increment the USERMOD *sysmod-id*, then on subsequent APPLY commands the REDO operand is required on the SMP/E APPLY statement.

2. In area **2** the ++VER may require a PRE(UNxxxxx) parameter on the statement if any maintenance has already been applied to the product. Please see the SMP/E Reference and Guide manuals for more details.

3. In area **3** specify new default assembler values for options. The sample shows *ADATA=YES* being selected as an example.

4. In area **4** specify new default assembler values for DDNAMES. The sample shows *UT1=SYSUT2* being selected as an example.

5. In area **5** you might want to use the COMPRESS option on the APPLY statement to minimize storage used for some or all of the target libraries.

6. Ensure that all PTF service which affects part ASMADOPT is ACCEPTed before applying USERMOD ML00001 to avoid regressing service on that part.

For the USERMOD, skip the ACCEPT processing.

Step 4: Place Procedures in Procedure Library

Four standard procedures - ASMAC, ASMACL, ASMACLG, and ASMACG - are supplied with this product. The target library SASMSAM1 contains these procedures (see Figure 6 on page 37, Figure 7 on page 38, Figure 8 on page 39, and Figure 9 on page 40).

As an alternative to moving these procedures to a JES defined procedure library, the use of the JCLLIB JCL statement should be considered.

Table 23. Supplied Procedures

Procedure	Description
ASMAC	Assembles only
ASMACL	Assembles and link-edits program
ASMACLG	Assembles, link-edits program and runs program
ASMACG	Assembles and uses the loader to run program

Because the name of the supplied procedures changed between Release 1 and Release 2 of High Level Assembler, the original Release 1 procedures are supplied as aliases in the target library SASMSAM1. If you want to move them to SYS1.PROCLIB or other JES defined procedure libraries then include another SELECT statement in the ASMASTD job. For example:

```
SELECT MEMBER=HLASMC,HLASMCG,HLASMCLG,HLASMCL
```

Carry out the following task to move the supplied procedures.

- Edit and submit job ASMASTD to move the procedures. Consult the instructions in the sample job for more information as shown in Figure 5 on page 36 and change where required.

  This job moves the procedures into SYS1.PROCLIB; this should be altered to match your system procedure library.

```
//ASMASTD  JOB  <JOB CARD PARAMETERS>
//*
//********************************************************************
//*   Licensed Materials - Property of IBM                           *
//*                                                                  *
//*   5696-234  5647-A01                                             *
//*                                                                  *
//*   (C) Copyright IBM Corp. 1992, 2000. All Rights Reserved.       *
//*                                                                  *
//*   US Government Users Restricted Rights - Use,                   *
//*   duplication or disclosure restricted by GSA ADP                *
//*   Schedule Contract with IBM Corp.                               *
//*                                                                  *
//********************************************************************
//********************************************************************
//*                                                                  *
//* ASMASTD  JOB                                                     *
//*                                                                  *
//* THIS JCL WILL COPY THE STANDARD ASSEMBLER PROCEDURES.            *
//*                                                                  *
//*  CAUTION: THIS IS NEITHER A JCL PROCEDURE NOR A COMPLETE JOB.    *
//*  BEFORE USING THIS JOB, YOU WILL HAVE TO MAKE THE FOLLOWING      *
//*  MODIFICATIONS:                                                  *
//*                                                                  *
//*  1. CHANGE THE JOB CARD TO MEET YOUR SYSTEM REQUIREMENTS         *
//*  2. CHANGE #hlq TO MATCH THE HIGH LEVEL QUALIFIER OF THE         *
//*     TARGET LIBRARIES.                                            *
//*                                                                  *
//********************************************************************
//         EXEC PGM=IEBCOPY
//SASMSAM1 DD  DSN=#hlq.SASMSAM1,
//             DISP=SHR
//OUT      DD  DSN=SYS1.PROCLIB,DISP=OLD
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
         COPY INDD=SASMSAM1,OUTDD=OUT
         SELECT MEMBER=ASMAC,ASMACG,ASMACLG,ASMACL
/*
```

Figure 5. Copying Procedures from Sample Library to Procedure Library

Supplied procedures: The following figures display the contents of each supplied procedure.

```
//ASMAC    PROC
//*
//*********************************************************************
//*   Licensed Materials - Property of IBM                            *
//*                                                                   *
//*   5696-234   5647-A01                                             *
//*                                                                   *
//*   (C) Copyright IBM Corp. 1992, 2000. All Rights Reserved.        *
//*                                                                   *
//*   US Government Users Restricted Rights - Use,                    *
//*   duplication or disclosure restricted by GSA ADP                 *
//*   Schedule Contract with IBM Corp.                                *
//*                                                                   *
//*********************************************************************
//*                                                                   *
//*   ASMAC                                                           *
//*                                                                   *
//* THIS PROCEDURE RUNS THE HIGH LEVEL ASSEMBLER AND CAN BE USED      *
//* TO ASSEMBLE PROGRAMS.                                             *
//*                                                                   *
//*********************************************************************
//C        EXEC PGM=ASMA90
//SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1   DD  DSN=&&SYSUT1,SPACE=(4096,(120,120),,,ROUND),
//             UNIT=SYSALLDA,DCB=BUFNO=1
//SYSPRINT DD  SYSOUT=*
//SYSLIN   DD  DSN=&&OBJ,SPACE=(3040,(40,40),,,ROUND),
//             UNIT=SYSALLDA,DISP=(MOD,PASS),
//             DCB=(BLKSIZE=3040,LRECL=80,RECFM=FB,BUFNO=1)
```

Figure 6. ASMAC Procedure

```
//ASMACL   PROC
//*
//*********************************************************************
//*   Licensed Materials - Property of IBM                            *
//*                                                                   *
//*   5696-234   5647-A01                                             *
//*                                                                   *
//*   (C) Copyright IBM Corp. 1992, 2000. All Rights Reserved.        *
//*                                                                   *
//*   US Government Users Restricted Rights - Use,                    *
//*   duplication or disclosure restricted by GSA ADP                 *
//*   Schedule Contract with IBM Corp.                                *
//*                                                                   *
//*********************************************************************
//*                                                                   *
//*     ASMACL                                                        *
//*                                                                   *
//* THIS PROCEDURE RUNS THE HIGH LEVEL ASSEMBLER, LINK-EDITS THE      *
//* NEWLY ASSEMBLED PROGRAM.                                          *
//*                                                                   *
//*********************************************************************
//*
//C        EXEC PGM=ASMA90
//SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1   DD  DSN=&&SYSUT1,SPACE=(4096,(120,120),,,ROUND),
//             UNIT=SYSALLDA,DCB=BUFNO=1
//SYSPRINT DD  SYSOUT=*
//SYSLIN   DD  DSN=&&OBJ,SPACE=(3040,(40,40),,,ROUND),
//             UNIT=SYSALLDA,DISP=(MOD,PASS),
//             DCB=(BLKSIZE=3040,LRECL=80,RECFM=FB,BUFNO=1)
//L        EXEC PGM=HEWL,PARM='MAP,LET,LIST,NCAL',COND=(8,LT,C)
//SYSLIN   DD  DSN=&&OBJ,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//SYSLMOD  DD  DISP=(,PASS),UNIT=SYSALLDA,SPACE=(CYL,(1,1,1)),
//             DSN=&&GOSET(GO)
//SYSUT1   DD  DSN=&&SYSUT1,SPACE=(1024,(120,120),,,ROUND),
//             UNIT=SYSALLDA,DCB=BUFNO=1
//SYSPRINT DD  SYSOUT=*
```

Figure 7. ASMACL Procedure

```
//ASMACLG  PROC
//*
//*********************************************************************
//*   Licensed Materials - Property of IBM                            *
//*                                                                   *
//*   5696-234   5647-A01                                             *
//*                                                                   *
//*   (C) Copyright IBM Corp. 1992, 2000. All Rights Reserved.        *
//*                                                                   *
//*   US Government Users Restricted Rights - Use,                    *
//*   duplication or disclosure restricted by GSA ADP                 *
//*   Schedule Contract with IBM Corp.                                *
//*                                                                   *
//*********************************************************************
//*                                                                   *
//*     ASMACLG                                                       *
//*                                                                   *
//* THIS PROCEDURE RUNS THE HIGH LEVEL ASSEMBLER, LINK-EDITS THE      *
//* NEWLY ASSEMBLED PROGRAM AND RUNS THE PROGRAM AFTER                *
//* THE LINK-EDIT IS ACCOMPLISHED.                                    *
//*                                                                   *
//*********************************************************************
//*
//C        EXEC PGM=ASMA90
//SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1   DD  DSN=&&SYSUT1,SPACE=(4096,(120,120),,,ROUND),
//             UNIT=SYSALLDA,DCB=BUFNO=1
//SYSPRINT DD  SYSOUT=*
//SYSLIN   DD  DSN=&&OBJ,SPACE=(3040,(40,40),,,ROUND),
//             UNIT=SYSALLDA,DISP=(MOD,PASS),
//             DCB=(BLKSIZE=3040,LRECL=80,RECFM=FB,BUFNO=1)
//L        EXEC PGM=HEWL,PARM='MAP,LET,LIST',COND=(8,LT,C)
//SYSLIN   DD  DSN=&&OBJ,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//SYSLMOD  DD  DISP=(,PASS),UNIT=SYSALLDA,SPACE=(CYL,(1,1,1)),
//             DSN=&&GOSET(GO)
//SYSUT1   DD  DSN=&&SYSUT1,SPACE=(1024,(120,120),,,ROUND),
//             UNIT=SYSALLDA,DCB=BUFNO=1
//SYSPRINT DD  SYSOUT=*
//G        EXEC PGM=*.L.SYSLMOD,COND=((8,LT,C),(8,LT,L))
```

Figure 8. ASMACLG Procedure

```
//ASMACG   PROC
//*
//*********************************************************************
//*   Licensed Materials - Property of IBM                            *
//*                                                                   *
//*   5696-234   5647-A01                                             *
//*                                                                   *
//*   (C) Copyright IBM Corp. 1992, 2000. All Rights Reserved.        *
//*                                                                   *
//*   US Government Users Restricted Rights - Use,                    *
//*   duplication or disclosure restricted by GSA ADP                 *
//*   Schedule Contract with IBM Corp.                                *
//*                                                                   *
//*********************************************************************
//*                                                                   *
//*     ASMACG                                                        *
//*                                                                   *
//* THIS PROCEDURE RUNS THE HIGH LEVEL ASSEMBLER AND WILL USE         *
//* THE LOADER PROGRAM TO RUN THE NEWLY ASSEMBLED PROGRAM.            *
//*                                                                   *
//*********************************************************************
//*
//C        EXEC PGM=ASMA90
//SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1   DD  DSN=&&SYSUT1,SPACE=(4096,(120,120),,,ROUND),
//             UNIT=SYSALLDA,DCB=BUFNO=1
//SYSPRINT DD  SYSOUT=*
//SYSLIN   DD  DSN=&&OBJ,SPACE=(3040,(40,40),,,ROUND),
//             UNIT=SYSALLDA,DISP=(MOD,PASS),
//             DCB=(BLKSIZE=3040,LRECL=80,RECFM=FB,BUFNO=1)
//G        EXEC PGM=LOADER,PARM='MAP,LET,PRINT',COND=(8,LT,C)
//SYSLIN   DD  DSN=&&OBJ,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//SYSLOUT  DD  SYSOUT=*
```

Figure 9. ASMACG Procedure

Except for the ASMAC procedure, you might receive the following message when running these procedures:

```
IEF686I DDNAME REFERRED TO ON DDNAME KEYWORD IN PRIOR STEP WAS NOT RESOLVED
```

This warning message indicates that there were no linkage editor control statements. If you supply these statements, they would follow a SYSIN DDNAME. If you supplied no linkage-editor control statements this message can be ignored. Figure 5 on page 36 shows an example of how to copy the procedures from the sample library to the procedure library. Make sure no other procedures with the same name are already present in the procedure library.

If the user has chosen to place the High Level Assembler modules in a private library, then it will be necessary to add the appropriate STEPLIB DD record to the procedures.

Step 5: Place High Level Assembler into Link Pack Area

Refer to Table 19 on page 19 for the list of modules and their attributes that you want to move.

These modules are within the target library SASMMOD1. The modules may be copied to a linklist library, an LPA library or a private library depending upon the user's requirements.

Step 6: Assembler H Migration (optional)

In order to simplify migration of existing assembler procedures the IEV90 module can be given as an alias of ASMA90. A sample usermod, ASMAIEV supplied in the SASMSAM1 library can perform that function.

Edit and submit job ASMAIEV to create an alias of ASMA90. Consult the instructions in the sample job for more information as shown in Figure 10 on page 42 and change where required.

```
//ASMAIEV  JOB  <JOB CARD PARAMETERS>
//*
//*********************************************************************
//*   Licensed Materials - Property of IBM                            *
//*                                                                   *
//*   5696-234  5647-A01                                              *
//*                                                                   *
//*   (C) Copyright IBM Corp. 1992, 2000. All Rights Reserved.        *
//*                                                                   *
//*   US Government Users Restricted Rights - Use,                    *
//*   duplication or disclosure restricted by GSA ADP                 *
//*   Schedule Contract with IBM Corp.                                *
//*                                                                   *
//*********************************************************************
//*********************************************************************
//*                                                                   *
//* ASMAIEV  JOB                                                      *
//*                                                                   *
//* THIS JCL IS A USERMOD TO CREATE AN ALIAS of IEV90 FOR ASMA90.     *
//*                                                                   *
//*  CAUTION: THIS IS NEITHER A JCL PROCEDURE NOR A COMPLETE JOB.     *
//*  BEFORE USING THIS JOB, YOU WILL HAVE TO MAKE THE FOLLOWING       *
//*  MODIFICATIONS:                                                   *
//*                                                                   *
//*  1. CHANGE THE JOB CARD TO MEET YOUR SYSTEM REQUIREMENTS          *
//*  2. CHANGE #globalcsi TO THE DATASET NAME OF YOUR GLOBAL          *
//*     CSI DATASET.                                                  *
//*  3. CHANGE #hlq TO THE HIGH LEVEL QUALIFIER USED FOR THE TARGET   *
//*     LIBRARIES.                                                    *
//*  4. CHANGE #date ON THE ++USERMOD STATEMENT TO BE THE CURRENT     *
//*     DATE. IT IS IN THE FORM OF YYYYDDD.                           *
//*  5. CHANGE #tzone TO THE NAME OF YOUR TARGET ZONE.                *
//*                                                                   *
//*                                                                   *
//*********************************************************************
```

Figure 10 (Part 1 of 2). Create an IEV90 Alias

```
//USERMOD  EXEC PGM=GIMSMP,REGION=4096K
//SMPCSI   DD DSN=#globalcsi,
//            DISP=SHR
//SMPOUT   DD SYSOUT=*
//SMPRPT   DD SYSOUT=*
//SMPLIST  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSLIB   DD  DSN=#hlq.SASMMAC1,
//             DISP=SHR
//STEPLIB  DD  DSN=#hlq.SASMMOD1,
//             DISP=SHR
//SMPPTFIN DD  DATA,DLM=QX
++USERMOD(ML00009) REWORK(#date).       1
++VER(Z038) FMID(HMQ4140).
++JCLIN.
//S2      EXEC LINKS,
//             PARM='XREF,MAP,NCAL,RENT,REUS,REFR,AMOD=ANY,RMOD=24',
//             UNIT='',SER=,N=,NAME=SASMMOD1,P1='',
//             MOD=,P2='',OBJ=,CLASS=
//AASMMOD1 DD  DISP=SHR,DSN=#hlq.AASMMOD1
//SYSLIN   DD  *
 ORDER ASMA90
 INCLUDE AASMMOD1(ASMACPR)
 INCLUDE AASMMOD1(ASMA90)
 ALIAS IEV90
 ENTRY ASMA90
 NAME  ASMA90(R)
/*
++MOD(ASMA90) LKLIB(SASMMOD1).
QX
//SMPCNTL  DD  *
   SET BDY(GLOBAL).
   RECEIVE S(ML00009) SYSMODS.
   SET BDY(#tzone).
   APPLY S(ML00009).           2
/*
```

Figure 10 (Part 2 of 2). Create an IEV90 Alias

When running this usermod consider the following:

1. Ensure that all PTF service which affects part ASMA90 is ACCEPTed before applying USERMOD ML00009 to avoid regressing service to that part.
2. In area **1** use the REWORK option if you change the USERMOD frequently.

   Specify REWORK if the USERMOD name will be reused every time and you do not prefer to use the REJECT command before receiving the USERMOD.

   Every time you re-apply the USERMOD, increment the value of REWORK using the form of *yyyyddd*, where *yyyy* is the year (for example, 2000) and *ddd* is the day of the year (for example, 083 is the 23rd of March). For more information refer to the *OS/390 SMP/E Reference*.

   If you do not increment the USERMOD *sysmod-id*, then on subsequent APPLY commands the REDO operand is required on the SMP/E APPLY statement.
3. In area **2** you might want to use the COMPRESS option on the APPLY statement to minimize storage used for some or all of the target libraries.

For the USERMOD, skip the ACCEPT processing.

Chapter 5. Maintaining High Level Assembler on MVS

This chapter describes how to re-install High Level Assembler and how to apply service updates to High Level Assembler. To use the maintenance procedures effectively, you should have already installed High Level Assembler and any required products.

In addition, this chapter describes how to remove High Level Assembler.

Re-installing High Level Assembler

The action required here depends on the circumstance. If you want to re-install and you did not use the SMP/E ACCEPT command then use a SMP/E APPLY REDO command. However, if you did use the SMP/E ACCEPT command, then the product should be deleted before installing again. For more information refer to “Removing High Level Assembler” on page 47.

Applying Service Updates

You might need to apply maintenance or service updates to High Level Assembler periodically.

What You Receive

If you report a problem with High Level Assembler to your IBM Support Center, you will receive a tape containing one or more APARs or PTFs that have been created to solve your problem.

You might also receive a list of pre-requisite APARs or PTFs, which should have been applied to your system before applying the current service. These prerequisite APARs or PTFs, might relate to High Level Assembler or any other licensed product you have installed, including MVS.

To help you understand the service process, the following overview familiarizes you with applying service for High Level Assembler.

Checklist for Applying Service

Table 24 on page 46 lists the steps and associated SMP/E commands for installing corrective service on High Level Assembler. You can use Table 24 on page 46 as a checklist.

Table 24. Summary of Steps for Installing Service on High Level Assembler

Step	Description	SMP/E Command	page
__ 1	Prepare to install service.		46
__ 2	Receive service.	RECEIVE	46
__ 3	Accept previously applied service. (optional)	ACCEPT	46
__ 4	Apply service.	APPLY	46
__ 5	Test service.		46
__ 6	Accept service.	ACCEPT	47

Step 1. Prepare to Install Service

Before you start applying service:

1. Create a backup copy of the current High Level Assembler. Save this copy of High Level Assembler until you have completed installing the service and you are confident that the service runs correctly.
2. Research each service tape through the IBM Support Center for any errors and/or additional information. Note all errors on the tape that were reported by APARs and apply the applicable fixes.

Step 2. Receive the Service

Receive the service using SMP/E RECEIVE command. This can be done from the SMP/E dialogs in ISPF or using a batch job.

Step 3. Accept Applied Service (Optional)

Accept any service you applied earlier but did not accept, if you are satisfied that the earlier service is not causing problems in your installation. This can be done from the SMP/E dialogs in ISPF or using a batch job. Accepting the earlier service allows you to use the SMP/E RESTORE command to return to your current level if you encounter a problem with the service you are currently applying. This can be done from the SMP/E dialogs in ISPF or using a batch job.

Step 4. Apply the Service

Apply the service using SMP/E APPLY command. You should use the SMP/E APPLY command with the CHECK operand first. Check the output; if it shows no conflict, rerun the APPLY without the CHECK option. This can be done from the SMP/E dialogs in ISPF or using a batch job.

Do not apply the documented USERMODs until PTF service has been ACCEPTed. This is to avoid regressing service to the affected parts.

Step 5. Test the Service

Thoroughly test your updated High Level Assembler. Run the installation verification program to ensure that the product functions properly. A job to run the installation verification program is found in the supplied sample job ASMAIVP from the target library SASMSAM1. Do not accept a service update until you are confident that it runs correctly.

In the event of a serious problem, you can restore the backup copy of High Level Assembler.

Step 6. Accept the Service

Accept the service using SMP/E ACCEPT command. You should use the SMP/E ACCEPT command with the CHECK operand first. Check the output; if it shows no conflict, rerun the ACCEPT without the CHECK option. This can be done from the SMP/E dialogs in ISPF or using a batch job.

Removing High Level Assembler

To delete High Level Assembler, you must:

- Make sure no other products depend on it.
- Use a dummy function SYSMOD to delete it.
- Receive, apply and accept the dummy function, and run the UCLIN to delete the SYSMOD entries for the deleted function and the dummy function.

Edit and submit job ASMADEL0 to delete High Level Assembler. Consult the instructions in the sample job for more information.

Expected Return Codes and Messages: You receive message `GIM39701W` because the dummy function SYSMOD has no elements. The SMP/E RECEIVE command returns a return code of 4. If any USERMODs have been applied then the SMP/E APPLY command issues a `GIM44502W` message indicating USERMOD changes will be lost with a return code of 4. Both these warning messages can be ignored.

The target and distribution libraries can now be deleted. They are shown in Table 7 on page 8 and Table 8 on page 8.

Reporting a Problem with High Level Assembler

When reporting any difficulties refer to the diagnostic process as shown in Chapter 16, “Isolating the Problem” on page 135.

Report any difficulties with this product to your IBM Support Center. In the United States, if an APAR is required, submit the data to the location identified in the *Field Engineering Programming System General Information* manual (PSGIM), G229-2228, as being responsible for the failing component.

Table 25 identifies the component ID (COMP ID) for High Level Assembler.

Table 25. Component IDs

FMID	COMP ID	Component Name	REL
HMQ4140	569623400	MVS HIGH LEVEL ASM	140

Obtaining Service Information

Preventive Service Planning (PSP) information is continually updated as fixes are made available for problems. Check with your IBM Support Center or use either Information/Access or SoftwareXcel Extended to see whether there is additional PSP information you need. To obtain this information, specify the following UPGRADE and SUBSET values: HLASM140 and HMQ4140.

On VM: Planning for, Installing, Customizing, and Maintaining High Level Assembler

Chapter 6. Planning for Installing High Level Assembler on VM 51
Worksheet: Planning for Installing High Level Assembler on VM 51
What You Receive with High Level Assembler 52
Distribution Media 52
Basic Material 52
Optional Material 53
Cumulative Service Tape 53
Program Publications and Softcopy 53
What You Need to Install High Level Assembler 53
Operating System Requirements 53
Other Program Product Requirements 54
VMSES/E Considerations 54
DASD Storage Required 54
Planning to Install in a Saved Segment 56
Selecting National Language Support 56
Program Support 56
Program Services 57
Preventive Service Planning 57
Statement of Support Procedures 57
Program and Service Level Information 57
Program Level Information 57
Service Level Information 57
Publications Useful during Installation 58

Chapter 7. Planning for Customizing High Level Assembler on VM 59
Deciding Whether and What to Customize 59
Planning to Customize High Level Assembler Options 59
Why Do It 59
Choices to Make Now 59
Planning to Customize the User Exits 62

Chapter 8. Installing High Level Assembler on VM 63
Overview of Installation 63
Checklist for Installing High Level Assembler 63
Step 1: Prepare to Install High Level Assembler 63
Step 2: Allocate Resources for Installing High Level Assembler 65
Common Instructions for Allocating Storage Resources 65
If Installing on a Minidisk 66
If Installing on a Shared File System 66
Step 3: Install High Level Assembler 68
Step 4: Verify the Installation in a Test Environment 70
Step 5: Define and Build High Level Assembler Saved Segment (Optional) . . 71
Step 6: Put High Level Assembler into Production 77

Chapter 9. Customizing High Level Assembler on VM 81
Changing Option and DDNAMES Defaults 81
General Rules for coding the ASMADOPT ASSEMBLE file 81
Example of Modifying Option and DDNAME Values 81
Changing Saved Segment Name 82

Changing the User Exits 82

Chapter 10. Maintaining High Level Assembler on VM 83
Re-installing High Level Assembler 83
Applying Service Updates 83
What You Receive 83
Checklist for Applying Service 83
Step 1. Prepare to Install Service 84
Step 2. Clear the Alternate APPLY Disk 86
Step 3. Receive the New Service 86
Step 4. Apply the New Service 87
Step 5. Reapply Local Service 88
Step 6.Update Build Status Table 88
Step 7. Rebuild Serviced Parts (Objects) 90
Step 8. Verify the Service 91
Step 9. Place the Service into Production 91
Removing High Level Assembler 96
To Report a Problem with High Level Assembler 96
Obtaining Service Information 97

Chapter 6. Planning for Installing High Level Assembler on VM

This section contains the following planning information to help you properly install High Level Assembler on VM:

- Worksheet
- What you receive with High Level Assembler
- Identifying required and optional software
- VMSES/E considerations
- Verifying that you have enough DASD storage
- Deciding to install in a saved segment
- Planning which national language support you require
- Checking service updates

Worksheet: Planning for Installing High Level Assembler on VM

Before you begin the installation you should:

__ 1. Determine which of the following you are installing High Level Assembler as:

   __ Part of a VM/System Offering or VM/ESA System Delivery Offering

   __ By itself

   If installing High Level Assembler by itself, determine the product parts to be installed:

   __ COMPID *569623400*

   __ Feature number ____

   See “What You Receive with High Level Assembler” on page 52.

__ 2. Verify that required software (and optional software, if appropriate) is at the level needed. See “What You Need to Install High Level Assembler” on page 53.

__ 3. Ensure you are familiar with VMSES/E. See “VMSES/E Considerations” on page 54.

__ 4. Verify that adequate storage is available:

   __ Minidisks

   __ Shared File System

   See “DASD Storage Required” on page 54.

__ 5. Decide whether to install in a saved segment.

   See “Planning to Install in a Saved Segment” on page 56.

__ 6. Decide which national language you want used for diagnostic messages. See “Selecting National Language Support” on page 56.

__ 7. Obtain latest service updates needed. See “Program Support” on page 56.

What You Receive with High Level Assembler

You receive one of the following when you order High Level Assembler:

COMPIDs	Feature Numbers	System Name
569623400	5871	VM/ESA
569623400	5872	VM/ESA
569623400	5448	VM/ESA

Distribution Media

High Level Assembler is distributed on one of the following:

- Unlabeled 9-track magnetic tape written at 6250 BPI
- 3480 tape cartridge
- 4mm DAT cartridge

The tape or cartridge contains all the programs and data needed for installation.

Basic Material

Table 26 describes the tape or cartridge. Table 27 describes the file content of the program tape or cartridge.

Table 26. Basic Material: Program Tape

Medium	Feature Number	Physical Volume	External Label Identification	VOLSER
6250 tape	5871	1	HLASM V1R4 for VM	unlabeled
3480 cart.	5872	1	HLASM V1R4 for VM	unlabeled
4 mm cart.	5448	1	HLASM V1R4 for VM	unlabeled

Table 27. Program Tape: File Content

Tape File	Content
1	Tape Header
2	Tape Header
3	Product Header
4	Product Memo
5	Service Apply Lists
6	PTFPARTS
7	High Level Assembler Service
8	High Level Assembler Service
9	High Level Assembler Base Code
10	High Level Assembler Sample and Customization files
11	High Level Assembler Executable code

Optional Material

There are no optional materials for High Level Assembler.

Cumulative Service Tape

You might receive an additional tape containing cumulative service with your order. The PTFs on this tape have not yet been incorporated into this release.

If you received this product as part of a VM/ESA System Delivery Offering, PTFs that have not been incorporated into this release are provided on the tape. A separate cumulative service tape is *not* provided.

Program Publications and Softcopy

This section identifies the basic and optional publications for High Level Assembler.

One printed copy of the following manuals is included with the product:

- *HLASM Licensed Program Specifications* GC26-4944
- *HLASM Installation and Customization Guide* SC26-3494
- *HLASM Language Reference* SC26-4940
- *HLASM Programmer's Guide* SC26-4941
- *HLASM General Information* GC26-4943

For a list of books for related products, see “Bibliography” on page 223.

What You Need to Install High Level Assembler

The following sections identify the system requirements for installing High Level Assembler.

Operating System Requirements

High Level Assembler supports the following VM operating systems:

- VM/ESA Version 2 Release 4
- VM/ESA Version 2 Release 3
- VM/ESA Version 2 Release 2
- RSU9904 service level or above must be applied to VMSES/E on VM/ESA 2.4.0 prior to installing High Level Assembler
- RSU9905 service level or above must be applied to VMSES/E on VM/ESA 2.3.0 prior to installing High Level Assembler
- RSU9904 service level or above must be applied to VMSES/E on VM/ESA 2.2.0 prior to installing High Level Assembler

VMSES/E product packaging APAR VM62316 has to be applied. This will allow installation of products packaged in the Year 2000. It is contained in the above RSU's.

Other Program Product Requirements

No other products are required for High Level Assembler.

VMSES/E Considerations

This section describes items that should be considered before you install High Level Assembler

- VMSES/E is required to install and service this product.
- If multiple users install and maintain licensed products on your system there might be a problem getting the necessary write access to MAINT's 51D disk. If you find that there is contention for write access to the 51D disk, you can eliminate it by converting the software inventory from minidisk to the Shared File System (SFS). See the *VMSES/E Introduction and Reference* manual section "Changing the Software Inventory to an SFS Directory Entry Format," for information on how to make this change.
- You no longer install and service High Level Assembler strictly using the MAINT user ID, but use a new user ID P696234E. This is IBM's suggested user ID name. Customers are free to change this to any user ID they wish, however, a PPF override must be created.

  It might be easier to make this change during the substep 6 on page 64, after you have installed this product.
- If you are going to install to minidisks, keep in mind that the saved segment must be built from the installation user ID P696234E. If you want to build the segment from a common user ID, install High Level Assembler using the SFS system, so that the common user ID can access High Level Assembler code when building segments.

DASD Storage Required

Before installing you need to understand the user ID and DASD storage requirements. They are shown in Table 28 on page 55. Some important points to consider are:

- The installation user ID and minidisks are defined in "Step 2: Allocate Resources for Installing High Level Assembler" on page 65. They are listed in this chapter so that you get an idea of the resources that you need prior to allocating them.
- P696234E is the default user ID and can be changed. If you choose to change the name you need to create a Product Parameter File (PPF) override to change the name. This can be done in substep 6 on page 64.
- If you choose to install High Level Assembler on a common user ID, the default minidisk addresses for High Level Assembler might already be defined. If any of the default minidisks required by High Level Assembler are already in use, create an override to change the minidisks for High Level Assembler so they are unique.

Table 28. DASD Storage Requirements for Target Minidisks

Minidisk Owner (user ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
P696234E	2B2	9345 3390 3380 3375 3350	12 10 12 19 15	14400	1800	Contains all the base code shipped with High Level Assembler. **VMSYS:P696234E.HLASM.OBJECT**
P696234E	2C2	9345 3390 3380 3375 3350	4 4 4 7 5	4800	600	Contains sample files, user local modifications and user exits for High Level Assembler. **VMSYS:P696234E.HLASM.LOCAL**
P696234E	2D2	9345 3390 3380 3375 3350	12 10 12 19 15	14400	1800	Contains serviced files. **VMSYS:P696234E.HLASM.DELTA**
P696234E	2A6	9345 3390 3380 3375 3350	1 1 1 2 2	1200	150	Contains AUX files and software inventory files that represents your test service level of High Level Assembler. **VMSYS:P696234E.HLASM.APPLYALT**
P696234E	2A2	9345 3390 3380 3375 3350	1 1 1 2 2	1200	150	Contains AUX files and software inventory files that represents the service level of High Level Assembler that is currently in production. **VMSYS:P696234E.HLASM.APPLYPROD**
P696234E	29E	9345 3390 3380 3375 3350	8 6 8 12 9	8640	1080	Test build disk. This code will be copied to a production disk, (for example, MAINT 19E) so the production disk will also require this amount of free space. **VMSYS:P696234E.HLASM.TBUILD**
P696234E	191	9345 3390 3380 3375 3350	5 5 5 8 7	6000	750	P696234E 191 minidisk. **VMSYS:P696234E.**

Note: Cylinder values defined in this table are based on 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. The FBA blocks are listed as 1/2K but should be CMS formatted at 1K size.

If you plan to use SFS directories, then calculate the number of 4K blocks required from Table 28.

Planning to Install in a Saved Segment

Defining frequently used data (such as licensed programs) as saved segments provides several advantages:

- Because several users can access the same physical storage, use of real storage is minimized.
- Using saved segments decreases the input/output rate and the DASD paging space requirements, thereby improving the performance of the virtual machine.
- Saved segments attached to a virtual machine can reside above its defined virtual storage.
- In the install process a physical saved segment is defined and its logical saved segment loaded.

Segment planning should be done at the system level. A VMSES/E tool, VMFSGMAP provides system-level management support functions.

In “Step 5: Define and Build High Level Assembler Saved Segment (Optional)” on page 71 the process of defining and building a saved segment is shown in detail. If you are unfamiliar with this process, review the chapter concerning saved segments in the *VM/ESA Planning and Administration* manual.

Selecting National Language Support

When installing High Level Assembler, you can choose which language you want to use for diagnostic messages. Languages available are English, German, Spanish and Japanese. English is also available in mixed case or uppercase.

The following combinations are possible:

English Uppercase Diagnostic messages and listing headings printed in uppercase English

English Mixed Case Diagnostic messages and listing headings printed in mixed case English

German Diagnostic messages in German and listing headings printed in mixed case English.

Japanese Diagnostic messages in Kanji and listing headings printed in uppercase English.

Spanish Diagnostic messages in Spanish and listing headings printed in mixed case English.

Program Support

This section describes the IBM support available for High Level Assembler.

Program Services

Contact your IBM representative for specific information about available program services.

Preventive Service Planning

Before installing High Level Assembler, you should review the current Preventive Service Planning (PSP) information.

PSP Buckets are identified by UPGRADEs, which specify product levels, and SUBSETs, which specify the FMIDs for a product level. The UPGRADE and SUBSET values for High Level Assembler are:

Table 29. PSP Upgrade and Subset ID

UPGRADE	SUBSET	Description
HLASM140	HLASMVM340	HLASM VM

Statement of Support Procedures

Report any difficulties you have using this program to your IBM Support Center. If an APAR is required, the Support Center will provide the address to which any needed documentation can be sent.

Table 13 on page 12 identifies the component IDs (COMPID) for High Level Assembler.

Table 30. Component IDs

FMID	COMPID	Component Name	RETAIN Release
340	569623400	VM HIGH LEVEL ASM	340

Program and Service Level Information

This section identifies the program and any relevant service levels of High Level Assembler. The program level refers to the APAR fixes incorporated into the program. The service level refers to the PTFs integrated. Information about the cumulative service tape is also provided.

Program Level Information

A list of APAR fixes against previous releases of High Level Assembler that have been incorporated into this release is shown in Appendix C, “High Level Assembler Service” on page 207.

Service Level Information

No PTFs against this release of High Level Assembler have been incorporated into the product tape.

Publications Useful during Installation

Table 31. Publications Useful During Installation/Service on VM/ESA Version 2.2.0

Publication Title	Form Number
VM/ESA: VMSES/E Introduction and Reference	SC24-5747
VM/ESA: Service Guide	SC24-5749
VM/ESA: Planning and Administration	SC24-5750
VM/ESA: CMS Command Reference	SC24-5776
VM/ESA: CMS File Pool Planning, Administration, and Operation	SC24-5751
VM/ESA: System Messages and Codes	SC24-5784

Table 32. Publications Useful During Installation / Service on VM/ESA Version 2.3.0 or higher

Publication Title	Form Number
VM/ESA: VMSES/E Introduction and Reference	GC24-5837
VM/ESA: Service Guide	GC24-5838
VM/ESA: Planning and Administration	SC24-5750
VM/ESA: CMS Command Reference	SC24-5776
VM/ESA: CMS File Pool Planning, Administration, and Operation	SC24-5751
VM/ESA: System Messages and Codes	GC24-5841

Chapter 7. Planning for Customizing High Level Assembler on VM

This section provides information for planning the customization of High Level Assembler on VM. It includes:

- Deciding whether and what to customize
- Planning to customize the default option and DDNAMES
- Planning to customize the user exits

Deciding Whether and What to Customize

You need to consider whether the IBM-supplied values that come with High Level Assembler suit the needs of your site. These values control such features as:

- Assembler Options
- Default file names (DDNAMES)
- User exits

Make sure that High Level Assembler serves the needs of the application programmers at your site. Confer with them while you evaluate the customization options for High Level Assembler, particularly those concerning High Level Assembler options that are also available to the application programmers. Doing so will ensure that the modifications you make best support the application programs being developed at your site.

The information in this chapter helps you plan your customization. See Chapter 9, “Customizing High Level Assembler on VM” on page 81 for the actual customization procedure.

Planning to Customize High Level Assembler Options

High Level Assembler can be customized with a large range of options and DDNAMES for users. When customized they become the default options and DDNAMES.

Why Do It

The High Level Assembler options and DDNAMES should be reviewed to assess the required defaults for your site. Worksheets are provided for planning purposes.

Choices to Make Now

The following worksheets help you plan and code the options and DDNAMES appropriate for your site. To complete the worksheets, fill in the **Enter Selection** column.

Table 33 (Page 1 of 2). Worksheet: Options

Option	Enter Selection	IBM-Supplied Default	Description
ADATA	_____	NO	page 171
ADEXIT	_____	no exit specified	page 171
ALIGN	_____	YES	page 172
ALIGNWARN	_____	YES	page 172
ASA	_____	NO	page 173
BATCH	_____	YES	page 173
CODEPAGE	_____	047C	page 173
COMPAT	_____	NO	page 174
CONTWARN	_____	YES	page 175
DBCS	_____	NO	page 176
DECK	_____	NO	page 176
DELETE	_____	No options deleted	page 176
DSECT	_____	NO	page 177
DXREF	_____	YES	page 178
ESD	_____	YES	page 178
FLAG	_____	0	page 178
FOLD	_____	NO	page 179
GOFF	_____	NO	page 179
GOFFADATA	_____	NO	page 180
IMPLENWARN	_____	NO	page 180
INEXIT	_____	no exit specified	page 180
INFO	_____	NO	page 181
LANGUAGE	_____	EN	page 181
LIBEXIT	_____	no exit specified	page 182
LIBMAC	_____	NO	page 182
LIMIT	_____	NO	page 183
LINECOUNT	_____	60	page 183
LIST	_____	121	page 183
MAP	_____	YES	page 184
MXREF	_____	SOURCE	page 185
OBJECT	_____	YES	page 185
OBJEXIT	_____	no exit specified	page 186
OPTABLE	_____	UNI	page 186
PAGE0WARN	_____	NO	page 187
PCONTROL	_____	NO	page 188
PESTOP	_____	NO	page 190
PROFILE	_____	NO	page 190
PROFMEM	_____	ASMAPROF	page 190

Table 33 (Page 2 of 2). Worksheet: Options

Option	Enter Selection	IBM-Supplied Default	Description
PRTEXIT	_____	no exit specified	page 191
PUSHWARN	_____	YES	page 191
RA2	_____	NO	page 192
RECORDINFO	_____	YES	page 192
RENT	_____	NO	page 193
RLD	_____	YES	page 193
RXREF	_____	YES	page 193
SIZE	_____	MAX	page 194
STORAGE	_____	BELOW	page 195
SUBSTRWARN	_____	NO	page 196
SYSPARMV	_____	none specified	page 196
TERM	_____	NO	page 196
TEST	_____	NO	page 197
THREAD	_____	YES	page 197
TRANSLATE	_____	NO	page 198
TRMEXIT	_____	no exit specified	page 198
USING0WARN	_____	YES	page 198
WARN	_____	15	page 199
XREF	_____	(SHORT,UNREFS)	page 201

Table 34. Worksheet: DDNAMES

DDNAME	Enter Selection	IBM-Supplied Default	Syntax Description
ADATA	_____	SYSADATA	page 203
IN	_____	SYSIN	page 203
LIB	_____	SYSLIB	page 204
LIN	_____	SYSLIN	page 204
OPTN	_____	ASMAOPT	page 204
PRINT	_____	SYSPRINT	page 204
PUNCH	_____	SYSPUNCH	page 205
TERM	_____	SYSTERM	page 205
UT1	_____	SYSUT1	page 205

Planning to Customize the User Exits

Several exits are provided with the High Level Assembler, they are described in *HLASM Programmer's Guide*.

Note: If User Exits from a previous release are being used, these exits may need to be reassembled for this release.

Chapter 8. Installing High Level Assembler on VM

This chapter describes the installation method and the step-by-step procedures you use to install and activate the functions of High Level Assembler.

Overview of Installation

You install this release of High Level Assembler by using VMSES/E commands.

Checklist for Installing High Level Assembler

Table 35 lists the steps and associated VMSES/E commands for installing High Level Assembler. The remaining sections in this chapter describe each step. You can use Table 35 as a checklist.

Table 35. Summary of Steps for Installing High Level Assembler

Step	Description	VMSES/E Command	page
__ 1	Prepare to install High Level Assembler.	VMFINS	63
__ 2	Allocate resources for installing High Level Assembler.		65
__ 3	Install High Level Assembler.	VMFINS	68
__ 4	Verify the installation in a test environment.	VMFINS	70
__ 5	Define and build High Level Assembler saved segment (optional).	VMFSGMAP VMFBLD	71
__ 6	Place High Level Assembler into production.	VMFCOPY	77

Step 1: Prepare to Install High Level Assembler

Carry out the following preparatory tasks:

1. Log on as the High Level Assembler installer planner, typically the MAINT user ID. IBM recommends that you use P696234E as the product installation user ID.

   Normally, the MAINT user ID is used for the planning and setup for High Level Assembler, then the P696234E user ID is used to install the product.

   You can log on as any user ID that has read access to MAINT's 5E5 minidisk and write access to the MAINT 51D minidisk.

2. Mount installation tape.

   Attach the tape to the user ID at virtual address 181. Using virtual address 181 is required by the VMFINS EXEC.

3. Establish read access to VMSES/E, using the following command:

Command	Explanation
`link MAINT 5e5 5e5 rr` `access 5e5 b`	Commands to access VMSES/E code.

4. Establish write access to the software inventory disk.

Command	Explanation
link MAINT 51d 51d mr **access 51d d**	Commands to acquire write access to software inventory disk.

If another user already has the software inventory disk linked in write mode (R/W), you are not allowed to link to that disk in write mode. Have the other user relink to the software inventory disk in read-only mode, then try the above procedure again.

The VMSES/E system-level software inventory and other dependent files reside on the MAINT 51D disk.

You cannot proceed with the installation procedure until you have write access to the software inventory disk.

5. Load the High Level Assembler product control files onto the software inventory disk.

Command	Explanation
vmfins install info (nomemo	This command will: • Load Memo-to-Users file • Load product control files including the Product Parameter File (PPF) and the PRODPART files • Create VMFINS PRODLIST on your A-disk. The VMFINS PRODLIST contains a list of products on the installation tape.

Options:

NOMEMO Loads the memo from the tape but does not issue a prompt to send it to the system printer. If you replace it with the MEMO option, you are prompted for printing the memo.

6. Obtain the resource planning information.

Command	Explanation
vmfins install ppf 5696234E {HLASM|HLASMSFS} (plan nomemo	You are prompted to change the installation defaults (VMFINS2601R). Please refer to the notes below. Creates 5696234E PLANINFO file on your A disk. Use `HLASM` if High Level Assembler is to be installed on minidisk. Use `HLASMSFS` if High Level Assembler is to be installed in SFS.

This command *does not* load High Level Assembler.

Options:

NOMEMO Loads the memo from the tape but does not issue a prompt to send it to the system printer. If you replace it with the MEMO option, you are prompted for printing the memo.

PLAN VMFINS performs requisite checking, plans system resources, and provides an opportunity to override the defaults in the product parameter file.

Notes:

a. You can override:

- The name of the product parameter file
- The default user IDs
- Minidisk/directory definitions

b. If you change the PPF name, a default user ID, or other parameters via a PPF override, you'll need to use your changed values instead of those indicated (when appropriate), throughout the rest of the installation instructions, as well as the instructions for servicing High Level Assembler. For example, you'll need to specify your PPF override file name instead of 5696234E for certain VMSES/E commands.

c. If you're not familiar with creating PPF overrides using VMFINS, you should review the 'Using the Make Override Panel' section in Chapter 3 of the *VMSES/E Introduction and Reference* before you continue. Also refer to Appendix D, “Create Product Parameter File (PPF) Override (VM)” on page 209.

7. Review the install message log file

Command	Explanation
`vmfview install`	View the install message log.

Review the install message log ($VMFINS $MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA: System Messages and Codes,* or use online HELP.

Step 2: Allocate Resources for Installing High Level Assembler

Use the information from the 5696234E PLANINFO file to allocate storage resources for your installation.

In the planning chapter, you decided whether to install on minidisk or shared file system (SFS), see “DASD Storage Required” on page 54.

Common Instructions for Allocating Storage Resources

1. Obtain the user directory from the 5696234E PLANINFO file.
   - User directory entries contain all of the links and privilege classes necessary for the P696234E user ID.
   - Use the directory entry in PLANINFO as a model for input to your system directory.

The user directory entry resides at the bottom of the PLANINFO file at the end of the resource requirements section.

2. Add the P696234E directory to the system directory.

   Change the password for P696234E from xxxxx to a valid password in accordance with the security guidelines at your site.

Minidisk installers: Go on to "If Installing on a Minidisk."

SFS installers: Go on to "If Installing on a Shared File System."

If Installing on a Minidisk

After obtaining the user directory from the 5696234E PLANINFO file:

- Add the MDISK statements to the directory entry for P696234E.
- Use Table 28 on page 55 in Chapter 6, "Planning for Installing High Level Assembler on VM" to obtain the minidisk requirements.
- Place the new directory online using VM/Directory Maintenance (DIRMAINT) or an equivalent CP directory maintenance method.

Go on to "Step 3: Install High Level Assembler" on page 68.

If Installing on a Shared File System

After completing the common steps above, you must then:

1. If you want to use an SFS directory as the work space for the P696234E top directory, include the following in the P696234E directory entry. This change requires the directory to be rebuilt and placed online.

Directory Statement	Explanation
IPL CMS PARM FILEPOOL VMSYS	This directory statement allows automatic access to the P696234E's top directory as file mode A.

2. Place the new directory online using VM/Directory Maintenance (DIRMAINT) or an equivalent CP directory maintenance method.
3. Use the number of 4K blocks calculated from Table 28 on page 55 in Chapter 6, "Planning for Installing High Level Assembler on VM."
   - If you are installing all the default High Level Assembler SFS directories, the block requirements are summarized in the table.
   - If you are selectively installing SFS directories, calculate the number of 4K blocks your installation requires by adding up the storage required by each of the directories you plan to install.
4. From a user ID that is an administrator for the VMSYS filepool, issue the following command:

Command	Explanation
enroll user P696234E vmsys: (blocks *blocks*	This command enrolls the user P696234E in the VMSYS filepool. Where VMSYS is the default filepool.

Options:

BLOCKS Where *blocks* is the number of 4K blocks that you calculated previously.

This makes available a top directory of VMSYS:P696234E. to the user.

5. Compare your storage needs to the amount of storage available in the filepool.

Command	Explanation
query filepool status vmsys:	Command to get a list of directories in the filepool and the number of free blocks. *VMSYS* is the default filepool ID.

6. Determine if there are enough blocks available in the filepool to install High Level Assembler. This information can be obtained from the QUERY FILEPOOL STATUS command. Near the end of the output from this command is a list of minidisks in the filepool and the number of blocks free.

   If the number of free blocks is smaller than the number of blocks you need to install High Level Assembler (which you calculated in step 3 on page 66, above), you need to add space to the filepool.

   For instructions on adding space to a filepool, see *VM/ESA CMS File Pool Planning, Administration, and Operation*.

7. Create the necessary `vmsys:P696234E.HLASM` subdirectories used in the P696234E PLANINFO file.

Command	Explanation
set filepool vmsys: **create directory** *dirid*	Commands to create subdirectories

Options:

dirid Where *dirid* is the name of the SFS directory you're creating.

Examples of the command above include:

```
create directory vmsys:P696234E.HLASM
create directory vmsys:P696234E.HLASM.object
  ⋮
```

A complete list of default High Level Assembler SFS directories is provided in Table 28 on page 55 in Chapter 6, “Planning for Installing High Level Assembler on VM.” For information about the CREATE DIRECTORY command, see the *VM/ESA CMS Command Reference* manual.

There is no need to create the top directory (VMSYS:P696234E.) as this has been implicitly created by the ENROLL command.

8. Give the MAINT user ID READ authority to the general-use test build directory, using the GRANT AUTHORITY command.

Command	Explanation
grant auth vmsys:P696234E.HLASM.TBUILD to MAINT (read newread	The GRANT command permits copying files to MAINT's 19E production disk.

Options:

READ Gives the user READ authority on a file or directory.

NEWREAD Indicates that the user (or users) automatically receive READ authority for any new files added to the directory.

If necessary, see the *VM/ESA CMS Command Reference* manual for more information about the GRANT AUTHORITY command.

Go on to "Step 3: Install High Level Assembler"

Step 3: Install High Level Assembler

1. Log on to the installation user ID, P696234E.
2. Format minidisks

   If minidisks are to be used rather than SFS directories, then format these disks now. For a list of disk addresses, refer to Table 28 on page 55.
3. Create a PROFILE EXEC that contains the ACCESS commands for the MAINT 5E5 and 51D minidisks.

Command	Explanation
xedit profile exec a ===> input /**/ ===> input 'access 5e5 b' ===> input 'access 51d d' ===> file	Commands to edit a PROFILE EXEC and sample input statements.

4. Execute the profile to access MAINT's minidisks.

Command	Explanation
profile	Command to execute profile.

5. Verify that you have write access to the software inventory disk.

Command	Explanation
q 51d	Command to query access status of the software inventory disk.

   If the disk is not in read write mode (R/W) then see substep 4 on page 64.
6. Have the High Level Assembler installation tape mounted and attached to user ID P696234E at virtual address 181. VMFINS EXEC requires the tape drive be at a virtual address of 181.

7. Install High Level Assembler product from tape

   You might be prompted for additional information during VMFINS INSTALL processing, depending on your installation environment. If you're unsure how to respond to a prompt, refer to the 'Installing Products with VMFINS' in the *VMSES/E Introduction and Reference*.

Command	Explanation
vmfins install ppf 5696234E {HLASM|HLASMSFS} (nomemo nolink	Command to install High Level Assembler. Use `HLASM` if High Level Assembler is installed on minidisk. Use `HLASMSFS` if High Level Assembler is installed in SFS. You are prompted to create an override for High Level Assembler when this command executes. Refer to message text for response.

   Options:

   NOMEMO Loads the memo from the tape but does not issue a prompt to send them to the system printer. If you replace it with the MEMO option, you are prompted for printing the memo.

   NOLINK Indicates that you don't want VMFINS to link to the appropriate minidisks, only access them if they are not already accessed.

   Notes:

   a. You can override:

      - The name of the product parameter file
      - The default user IDs
      - Minidisk/directory definitions

   b. If you change the PPF name, a default user ID, or other parameters via a PPF override, you'll need to use your changed values instead of those indicated (when appropriate), throughout the rest of the installation instructions, as well as the instructions for servicing High Level Assembler. For example, you'll need to specify your PPF override file name instead of 5696234E for certain VMSES/E commands.

   c. If you're not familiar with creating PPF overrides using VMFINS, you should review the 'Using the Make Override Panel' section in Chapter 3 of the *VMSES/E Introduction and Reference* before you continue. Also refer to Appendix D, “Create Product Parameter File (PPF) Override (VM)” on page 209.

8. Review the install message log ($VMFINS $MSGLOG).

Command	Explanation
vmfview install	View the install message log.

Correct any problems before you go on. For information about handling specific error messages, see *VM/ESA: System Messages and Codes* or use online HELP.

Step 4: Verify the Installation in a Test Environment

If you need to change any of the default options or DDNAMES then refer to Chapter 9, “Customizing High Level Assembler on VM” on page 81. Upon completion of any customization, return to this step.

When you issue the command to update the build status table for High Level Assembler, VMSES/E also executes the verification EXEC, V5696234.

1. Update Build Status Table and run verification EXEC.

Command	Explanation
vmfins build ppf 5696234E {HLASM|HLASMSFS} (serviced nolink	Update VM SYSBLDS software inventory file for High Level Assembler Use `HLASM` if High Level Assembler is installed on minidisk. Use `HLASMSFS` if High Level Assembler is installed in SFS.

Options:

SERVICED Identifies build requirements and builds those objects flagged as SERVICED in the service-level build status table.

NOLINK Indicates that you don't want VMFINS to link to the appropriate minidisks, only to access them if they are not already accessed.

A verification test is run automatically to ensure that High Level Assembler was installed successfully. Figure 11 below shows sample screen output from the verification EXEC.

```
  ⋮
VMFINB2173I Executing verification exec V5696234
**************************************************
*** Verification assembly completed successfully
***
**************************************************
VMFINS2760I VMFINS processing completed successfully
```

Figure 11. Sample install verification output

2. Review the install message log ($VMFINS $MSGLOG).

Command	Explanation
vmfview install	View the install message log.

Correct any problems before you go on. For information about handling specific error messages, see *VM/ESA: System Messages and Codes* or use online HELP.

Step 5: Define and Build High Level Assembler Saved Segment (Optional)

If you do not plan to place High Level Assembler into a saved segment then bypass this step and continue onto "Step 6: Put High Level Assembler into Production" on page 77.

A logical saved segment provides several advantages such as saving on real storage, decreased input/output and a saved segment can reside above the defined virtual storage of a virtual machine. To build a saved segment you must:

1. Define the physical saved segment using the segment mapping tool VMFSGMAP.
2. Use VMFBLD to build it.

For more information on using VMSES/E for saved segments, review the chapter concerning saved segments in the *VM/ESA Planning and Administration* manual. Also refer to the SEGGEN command in *VM/ESA CMS Command Reference*.

Defining and building the High Level Assembler saved segment should be performed from the installation user ID. If you move any segments that are currently defined on your system, you must ensure that they are rebuilt from the user ID that maintains them.

1. Logon to the installation user ID P696234E.
2. Establish read access to VMSES/E.

Command	Explanation
`link MAINT 5e5 5e5 rr` `access 5e5 b`	Commands to access VMSES/E code.

3. Establish write access to the software inventory disk.

Command	Explanation
`link MAINT 51d 51d mr` `access 51d d`	Commands to access software inventory files

   If another user already has the software inventory disk linked in write mode (R/W), you are not allowed to link to that disk in write mode. Have the other user relink to the software inventory disk in read-only mode, then try the procedure above again.

   The MAINT 51d disk is where the VMSES/E system-level software inventory and other dependent files reside.

   You cannot proceed with the installation procedure until you have write access to the software inventory disk.

4. Add High Level Assembler segment object definitions to the SEGBLIST EXC00000 build list.

All the modules that are to reside in the logical saved segment have an RMODE=ANY and a AMODE=31. Therefore, the physical saved segment that is about to be defined can reside above 16MB.

Command	Explanation
vmfsgmap segbld esasegs segblist	This command displays a panel for making segment updates.

This command displays a panel for making segment updates, shown in Figure 12.

```
                            VMFSGMAP - Segment Map                  More: +
                                                                 Lines 1 to nn of nn

                 000-MB          001-MB          002-MB          003-MB
  Name     Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
M CMS      SYS W-W-------------1.............2.............3.............
M GCS      SYS W---------------1.............2.............3.............

                 004-MB          005-MB          006-MB          007-MB
  Name     Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
  CMSPIPES DCS 4.............5.............6.............RRRRRR----------
M GCS      SYS RRRRRRNNNNNNNNNNNNNNNNNNNNNNNN6.............7.............

                 008-MB          009-MB          00A-MB          00B-MB
  Name     Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
  DOSBAM   SPA 8.............9.............A.............====------------
  CMSBAM   MEM 8.............9.............A.............RRRR............
  CMSDOS   MEM 8.............9.............A.............R.............
  CMSVMLIB DCS RRRRRRRRRRRRRRR9.............A.............B.............

                 00C-MB          00D-MB          00E-MB          00F-MB
  Name     Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
  HELPINST DCS RRRRRRRRRRRRRRRD.............E.............F.............
 F1=Help     F2=Chk Obj   F3=Exit      F4=Chg Obj   F5=File       F6=Save
 F7=Bkwd     F8=Fwd       F9=Retrieve  F10=Add Obj  F11=Del Obj   F12=Cancel
 ====>__
```

Figure 12. Segment Map panel example.

Browse through this segment. If the ASMAPSEG segment is encountered it should be removed by placing the cursor under the ASMAPSEG name and pressing PF11. A successful deletion will give message

```
VMFSMD2046I Segment ASMAPSEG has been deleted
```

5. Go to Add Segment Definition panel by pressing PF10.

   Press **F10** to take you from the Segment Map panel to the Add Segment Definition panel. See Figure 13 on page 73 to see the Add Segment Definition panel displays.

```
                              Add Segment Definition
                                                Lines 1 to nn of nn

OBJNAME....:  ????????
DEFPARMS...:
SPACE......:
TYPE.......:  SEG
OBJDESC....:
OBJINFO....:
GT_16MB....:  NO
DISKS......:
SEGREQ.....:
PRODID.....:
BLDPARMS...:  UNKNOWN




 F1=Help     F2=Get Obj   F3=Exit      F4=Add Line  F5=Map        F6=Chk MEM
 F7=Bkwd     F8=Fwd       F9=Retrieve  F10=Seginfo  F11=Adj MEM   F12=Cancel
====>
```

Figure 13. Add Segment Definition panel example.

6. Obtain the High Level Assembler segment definitions from the PRODPART file. Enter the following highlighted values into the 'Add Segment Definition Panel'.

Command	Explanation
OBJNAME....: **ASMAPSEG** PRODID.....: **5696234E {HLASM|HLASMSFS}**	Use `HLASM` for building a segment from a minidisk or `HLASMSFS` for building a segment from SFS directories.
F10	**F10** obtains the High Level Assembler segment information from the 5696234E PRODPART file.

See Figure 14 on page 74 for the refreshed Add Segment definition panel that is displayed.

If you have created a PPF override file then replace the PPF name on the *BLDPARMS*. In this example, replace *5696234E* with your PPF override name.

```
                              Add Segment Definition                More: +
                                                                 Lines 1 to nn of nn



OBJNAME....:  ASMAPSEG
DEFPARMS...:  920-9D0 SR
SPACE......:
TYPE.......:  PSEG
OBJDESC....:  SEGMENT DEFINITION FOR HLASM R4
OBJINFO....:
GT_16MB....:  YES
DISKS......:
SEGREQ.....:
PRODID.....:  5696234E HLASM
BLDPARMS...:  PPF(5696234E HLASM ASMBLSEG)




VMFSMD2760I SEGINFO processing completed SUCCESSFULLY

 F1=Help     F2=Get Obj   F3=Exit      F4=Add Line  F5=Map        F6=Chk MEM
 F7=Bkwd     F8=Fwd       F9=Retrieve F10=Seginfo   F11=Adj MEM  F12=Cancel
====>
```

Figure 14. Add Segment Definition panel showing the new segment

7. Go back to the Segment Map panel.

Command	Explanation
F5	**F5** returns you to the Segment Map panel.

See Figure 15 on page 75 for the refreshed Segment Map panel that displays.

```
                          VMFSGMAP - Segment Map           More: -
                                                 Lines nn to nn of nn




               000-MB          001-MB          002-MB          003-MB
  Name     Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
M CMS      SYS W-W-------------1.............2.............3.............
M GCS      SYS W---------------1.............2.............3.............

               004-MB          005-MB          006-MB          007-MB
  Name     Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
  CMSPIPES DCS 4.............5.............6.............RRRRRR----------
M GCS      SYS RRRRRRNNNNNNNNNNNNNNNNNNNNNNNNNNNN6.............7.............

               008-MB          009-MB          00A-MB          00B-MB
  Name     Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
  DOSBAM   SPA 8.............9.............A.............====------------
  CMSBAM   MEM 8.............9.............A.............RRRR............
  CMSDOS   MEM 8.............9.............A.............R.............
  CMSVMLIB DCS RRRRRRRRRRRRRRR9.............A.............B.............
  DOSINST  DCS 8.............R--------------A.............B.............
P ASMAPSEG DCS 8.............--RRRRRRRRRRR--A.............B.............

               00C-MB          00D-MB          00E-MB          00F-MB
 F1=Help     F2=Chk Obj   F3=Exit      F4=Chg Obj   F5=File       F6=Save
 F7=Bkwd     F8=Fwd       F9=Retrieve  F10=Add Obj  F11=Del Obj   F12=Cancel
====>
```

Figure 15. Segment Map panel with added segments.

8. Save the new information and exit from the Segment Map panel.

   Press **F5** to save all changed information and exit the map panel.

9. Prepare to build the segment.

   a. ESA Mode considerations

      If you are placing the segment above 16MB, then prior to the IPL of CMS, use the SET MACHINE command to change the architecture mode to ESA.

Command	Explanation
set machine esa	Option ESA designates ESA/390 mode.

   b. Set storage and ipl virtual machine

Command	Explanation
def stor 24M	Issue this CP command. It defines the virtual machine storage to be greater than where the segment is loaded. The default VM size for this user ID is 8M. If defining a saved segment above 16MB then the virtual machine storage should be above 16MB. In general the storage should be large enough to hold the segment. If you are using SFS in this userid, even if only for the A disk, and you are placing the ASMAPSEG segment below 16Mb at 920-9D0 then use a 'DEF STOR 24M' command. This will allow CMS control blocks not to interfere with the storage area to be used by the High Level Assembler saved segment.
ipl cms parm clear nosprof instseg no mtseg no ***** DO NOT press ENTER at the VM READ*****	This command clears your virtual machine. This command bypasses the execution of the system profile (SYSPROF EXEC) and the loading of the installation saved segment (CMSINST). If using SFS, append to the **IPL CMS** command **filepool vmsys**. This assumes `VMSYS` is the default filepool.
access (noprof	Bypass the execution of the PROFILE EXEC.

c. Access disks

Command	Explanation
access 5e5 b	Access the VMSES/E code.
link MAINT 51d 51d mr **access 51d d**	Establish write access the software inventory disk.

10. Issue VMFBLD command to build the High Level Assembler segment.

Command	Explanation
vmfbld ppf segbld esasegs segblist ASMAPSEG (serviced	

Options:

SERVICED Identifies build requirements and builds those objects flagged as SERVICED in the service-level build status table.

11. Update CMS system disk with SYSTEM SEGID file.

    The system segment identification file (SYSTEM SEGID) must reside on the CMS system disk (usually file mode S) and must be named SYSTEM SEGID

so that it is available to CMS at initialization time. This allows CMS to recognize the logical saved segment name.

The file must also be copied to the test CMS system disk to prevent back-levelling during application of service.

The following functions can only be performed by authorized user IDs. Contact your system support personnel, if required.

a. Release disk for use by MAINT user ID.

Command	Explanation
rel d (det	Release the software inventory disk.

b. Log on to MAINT user ID.

c. Move SYSTEM SEGID file

Command	Explanation
acc 190 t	Access the production CMS system disk (usually MAINT 190).
acc 490 v	Access the test CMS system disk (usually MAINT 490).
acc 51d d	Access the software inventory disk.
copyfile system segid d system segid t2 (replace olddate	Copy file to production CMS system disk.
copyfile system segid d system segid v2 (replace olddate	Copy file to test CMS system disk.

Options:

OLDDATE Uses the date and time on each input file as the date and time of the last update of each corresponding output file.

REPLACE Causes the output file to replace an existing file with the same file identifier.

The CMS system is not resaved here, it is done after moving High Level Assembler to MAINT's 19E disk in “Step 6: Put High Level Assembler into Production.”

Step 6: Put High Level Assembler into Production

1. Log on to MAINT if you plan to put High Level Assembler general-use code on the 'Y' disk (MAINT's 19E disk). Or log on to the owner of the disk that is to contain the production level of the High Level Assembler code.
2. Link to High Level Assembler code and access MAINT's Y disk

   Choose which access is required depending on whether you have used a minidisk or SFS.

   Minidisk access is :

Command	Explanation
`link P696234E 29e 29e rr` `access 29e e`	Access High Level Assembler code on a minidisk.
`access 19e f`	Access CMS's Y disk.

SFS access is :

Command	Explanation
`access vmsys:P696234E.HLASM.TBUILD e`	Access High Level Assembler code on a SFS directory.
`access 19e f`	Access CMS's Y disk.

3. Check if a previous High Level Assembler Release is present.

   If the Y disk contains a High Level Assembler Release, then remove it. Migration is covered in substep 5 on page 79. This Release code can be identified by modules with a prefix of *ASMA* and the *HLASM* module (Release 1 only).

4. Move High Level Assembler to the Y disk.

   If you have created a saved segment for High Level Assembler, then the only modules that should be copied by the VMFCOPY command are:

   - ASMAHL
   - ASMA90
   - ASMADOPT
   - ASMAINFO

   Therefore the VMFCOPY command must be modified to only copy the four modules which do not reside in the saved segment.

   If you have not created a saved segment then all the High Level Assembler modules should be placed onto the Y disk using the VMFCOPY command.

   In both cases (saved segment or no saved segment), the ASMAMAC MACLIB should be copied.

Command	Explanation
`vmfcopy * MODULE e = = f2 (prodid 5696234E%HLASM olddate replace`	
	Please refer to the previous note about which modules should be copied if you have created a logical saved segment.
`vmfcopy * MACLIB e = = f2 (prodid 5696234E%HLASM olddate replace`	

Options:

OLDDATE Uses the date and time on each input file as the date and time of the last update of each corresponding output file.

REPLACE Causes the output file to replace an existing file with the same file identifier.

If you do not want to use the Y disk for general use code, log on as the owner of the disk where you put the production level of the High Level Assembler code.

The VMFCOPY command updates the VMSES PARTCAT file on the 19E disk.

5. Migration from previous Assemblers.

   If Release 1 of High Level Assembler has been removed then the ASMAHL MODULE must be copied to the Y disk as HLASM MODULE. This will allow VM service to function correctly.

   If you want to migrate from previous assemblers, you should consider the following:

   - HLASM MODULE in High Level Assembler Release 1 can be replaced by ASMAHL MODULE in High Level Assembler Release 2 or higher.
   - HASM MODULE in Assembler H can be replaced by ASMAHL MODULE in High Level Assembler Release 2 or higher. However, ASMAHL MODULE does not support the *NUM* and *STMT* options.
   - IEV90 MODULE in Assembler H can be replaced by the ASMA90 MODULE in High Level Assembler Release 2 or higher. If this is done, the ASMAHL MODULE should also be copied as HASM as well.

   You can then copy them to the Y disk.

Command	Explanation
`vmfcopy ASMAHL MODULE e HLASM = f2 (prodid 5696234E%HLASM olddate replace`	
`vmfcopy ASMAHL MODULE e HASM = f2 (prodid 5696234E%HLASM olddate replace`	
`vmfcopy ASMA90 MODULE e IEV90 = f2 (prodid 5696234E%HLASM olddate replace`	

Options:

OLDDATE Uses the date and time on each input file as the date and time of the last update of each corresponding output file.

REPLACE Causes the output file to replace an existing file with the same file identifier.

6. Move User Exits (Optional)

   User exits are supplied as sample assembler source on the LOCAL disk. If these exits are to be made generally available, they can be moved (ASMAX... ASSEMBLE) to the MAINT 19E disk, or the LOCAL disk can be made available to the users.

7. Resave the CMS saved system

   This updates the shared Y-STAT (the saved Y disk file directory). If you have created a saved segment, it also updates the S-STAT (the saved S disk file directory) to reflect the updated SYSTEM SEGID file.

a. Define named saved system

This creates a named saved system for CMS.

Command	Explanation
`vmfsetup esa cms`	Allows access to the SAMPNSS EXEC which resides on MAINT's 193 disk.
`sampnss cms`	Create named saved system for CMS.

b. Resave the CMS system

Command	Explanation
`ipl 190 clear parm savesys cms`	Resave CMS saved system to reflect the change to the S-STAT (the saved S disk file directory).

This completes the installation process.

Chapter 9. Customizing High Level Assembler on VM

You can customize, or modify, High Level Assembler only after installing the product (as described in Chapter 8, “Installing High Level Assembler on VM” on page 63).

Changing Option and DDNAMES Defaults

Consult worksheet on “Planning to Customize High Level Assembler Options” on page 59 to identify what default options or DDNAMES need to be changed.

You use VMSES/E to create a local modification that changes the default options and DDNAMES. You modify ASMADOPT ASSEMBLE on the LOCAL disk.

Appendix E, “Local Modification Procedures (VM)” on page 211 shows how a local modification procedure is carried out for this product. For further information refer to *VM/ESA Service Guide*.

Before modifying ASMADOPT, read the following sections.

General Rules for coding the ASMADOPT ASSEMBLE file

- Code in columns 2 through 71.
- Do not put a comma in front of the first option in your macro.
- Begin any continuation lines in column 16 and place a non-blank character in column 72 of the previous line. You can break the coding after any comma.
- Place an END statement after the macro instruction lines.
- Modify only some of the default values in the macro, High Level Assembler uses the IBM-supplied defaults for the unchanged ones.
- For a description of full syntax for the High Level Assembler options, see Appendix A, “High Level Assembler Options” on page 171.
- For a description of full syntax for the High Level Assembler DDNAMES, see Appendix B, “High Level Assembler DDNAMES (MVS and CMS)” on page 203.

Example of Modifying Option and DDNAME Values

ASMADOPT looks likes this:

```
ASMAOPT
END
```

Figure 16. ASMADOPT Example

For example, you might make the following changes, shown in Figure 17 on page 82.

- Change the default value for the XREF option to SHORT, and set the PESTOP option to YES.
- Change the default value for the UT1 default to SYSUT2.

```
ASMADOPT        CSECT
                ASMAOPT XREF=SHORT,PESTOP=YES
                ASMADD  UT1=SYSUT2
                END
```

Figure 17. ASMADOPT Example Modified

If you require more than one line to code your changes, observe the standard statement continuation rule.

Changing Saved Segment Name

The default name for the physical saved segment is ASMAPSEG. There is no restriction on the name of the physical saved segment. High Level Assembler uses the SEGMENT LOAD command to load the logical saved segment. This command uses the name of the logical saved segment.

For testing purposes, High Level Assembler can reside in more than one physical saved segment. To switch between them, use SEGMENT ASSIGN command. The name of the logical saved segment is ASMA93 and cannot be changed.

Follow the instructions outlined in “Step 5: Define and Build High Level Assembler Saved Segment (Optional)” on page 71 with the following changes:

1. The name of the physical saved segment can be changed on the 'Add Segment Definition' panel within the VMFSGMAP tool as shown in Figure 14 on page 74.
2. You will also need to create a *PSEG* file. This could be done by copying the *ASMAPSEG PSEG* file on the test build disk and renaming it to the new name.
3. Change the physical segment name on the *VMFBLD* command in substep 10 on page 76 from ASMAPSEG to the new name.

Changing the User Exits

Refer to *HLASM Programmer's Guide*. The user exits are supplied in source form as guides so the user can modify them to suit their individual needs. In order to assist users the following sample source is provided.

- ASMAXINV
  This is a sample source exit.
- ASMAXPRT
  This is a sample listing exit.
- To support the ADATA exit the following are provided:
  1. ASMAXFSK
     Sample skeleton SYSADATA filter routine
  2. ASMAXFMT
     Sample filter management table
  3. ASMAXFLU
     Sample filter module to dump ADATA records
  4. ASMAXADT
     Sample ADATA exit to load filter routine

Chapter 10. Maintaining High Level Assembler on VM

This chapter describes how to re-install, or remove High Level Assembler and how to apply service updates. To effectively use the maintenance procedures, you must have already installed High Level Assembler and any required products.

To become more familiar with service using VMSES/E, you should read the introductory chapters in *VMSES/E Introduction and Reference*. This manual also contains the command syntax for the VMSES/E commands listed in the procedure.

Each step of the servicing instructions must be followed; do not skip any step unless otherwise directed. All instructions showing accessing of disks assume default minidisk addresses. If different minidisk addresses are used, or if using a shared-file system, change the instructions appropriately.

Re-installing High Level Assembler

You should delete the product and commence the installation process from the beginning. To delete the product refer to “Removing High Level Assembler” on page 96. When the product is deleted, start the installation from “Step 1: Prepare to Install High Level Assembler” on page 63.

Applying Service Updates

This section describes how to apply maintenance or service updates to High Level Assembler.

What You Receive

If you report a problem with High Level Assembler to your IBM Support Center, you will receive a tape containing one or more APARs or PTFs which solve your problem.

You might also receive a list of pre-requisite APARs or PTFs, which should have been applied to your system before applying the current service. These pre-requisite APARs or PTFs might relate to High Level Assembler or any other licensed product you have installed, including VM/ESA.

The following overview familiarizes you with some of the aspects of applying service for High Level Assembler.

Checklist for Applying Service

Table 36 on page 84 lists the steps and associated VMSES/E commands for installing corrective service on High Level Assembler. You can use Table 36 on page 84 as a checklist.

Table 36. Summary of Steps for Installing Service on High Level Assembler

Step	Description	VMSES/E Command	page
__ 1	Prepare to install service.		84
__ 2	Clear the alternate APPLY disk. Doing this will allow you to remove the new service easily later if necessary.	VMFMRDSK	86
__ 3	Receive the new service.	VMFREC	87
__ 4	Apply the new service.	VMFAPPLY	88
__ 5	Reapply local service by entering local service into the software inventory. (if applicable)		88
__ 6	Update Build Status Table.	VMFBLD	90
__ 7	Rebuild Serviced Parts (Objects)	VMFBLD	90
__ 8	Verify the service that has been applied and built.		91
__ 9	Place into production		91

Step 1. Prepare to Install Service

Carry out preliminary steps prior to receiving service.

Electronic Service (envelope file)

If you have received the service electronically or on CD-ROM, follow the appropriate instructions to retrieve and decompact the envelope file to your A-disk. The decompaction is currently done by using the DETERSE MODULE. The file names of the decompacted files will be of the format:

- VLST*num* for the documentation envelope
- VPTF*num* for the service envelope

The file type for both of these files must be SERVLINK. You will need to enter the file name on the VMFREC commands that follow.

1. Create a backup copy of the current High Level Assembler before applying the service tape. Save this copy of High Level Assembler until you have completed installing the service and you are confident that the service runs correctly.
2. Log on to the High Level Assembler service user ID: P696234E.
3. Establish read-write (R/W) access to the software inventory disk.

Command	Explanation
`link MAINT 51d 51d mr` `access 51d d`	Commands to access the software inventory disk.

4. Mount the High Level Assembler corrective service tape and verify that it is attached to user ID P696234E. Use an address of 181. (If you have a SERVLINK file make sure that it is available on the A-disk.)

5. Establish the correct minidisk access order.

   The VMFSETUP command accesses all the required disks or SFS directories to establish the needed filemodes.

Command	Explanation
vmfsetup 5696234E {HLASM|HLASMSFS}	5696234E is the PPF that is shipped with High Level Assembler. If you have your own PPF override, substitute your PPF name for 5696234E. Use `HLASM` if High Level Assembler is installed on minidisk. Use `HLASMSFS` if High Level Assembler is installed in SFS.

6. Receive Documentation. VMFREC with the INFO option loads the documentation and displays a list of all the products on the tape.

   a. if receiving the service from tape

Command	Explanation
vmfrec info	The INFO option loads the documentation (including the product service memo) to the 191 disk and displays a list of products on the tape.

   b. if receiving the service from an envelope file

Command	Explanation
vmfrec info (env vlst*num*	The INFO option loads the documentation (including the product service memo) to the 191 disk and displays a list of products on the tape.

7. Check the receive message log ($VMFREC $MSGLOG) for warning and error messages.

Command	Explanation
vmfview receive	View the receive message log

   Make a note of which products and components have service on the tape. To do this, use the PF5 key to show all the status messages that identify the products on the tape.

Step 2. Clear the Alternate APPLY Disk

Clear the alternate APPLY disk to ensure that you have a clean disk for new service.

1. Clear Disk

   Merge previously applied service to ensure that you have a clean alternate APPLY disk for new service.

Command	Explanation
vmfmrdsk 5696234E {HLASM|HLASMSFS} apply	Command to clear the alternate APPLY disk. Use HLASM if High Level Assembler is installed on minidisk. Use HLASMSFS if High Level Assembler is installed in SFS.

2. Check merge message log

Command	Explanation
vmfview mrd	Command to review the merge message log ($VMFMRD $MSGLOG). Correct any problems before you go on. For information about handling specific error messages, refer to *VM/ESA: System Messages and Codes*, or use online help.

Step 3. Receive the New Service

1. Receive New Service

   Note: if you are installing multiple service tapes, you can receive all the service for this product before applying and building it. For each service tape or electronic envelope you want to receive, do the following:

   a. if receiving the service from tape

Command	Explanation
vmfrec ppf 5696234E {HLASM|HLASMSFS}	Command to receive service from the service tape. All new service is loaded to the alternate DELTA disk. Use HLASM if High Level Assembler is installed on minidisk. Use HLASMSFS if High Level Assembler is installed in SFS.

   b. if receiving the service from the PTF envelope file

Command	Explanation
vmfrec ppf 5696234E {HLASM|HLASMSFS} (env vptf*num*	This command receives service from your service envelope. All new service is loaded to the DELTA disk. Use HLASM if High Level Assembler is installed on minidisk. Use HLASMSFS if High Level Assembler is installed in SFS.

For information about handling specific error messages, refer to *VM/ESA: System Messages and Codes*, or use online help.

2. Review receive message log

Command	Explanation
vmfview receive	Command to review the receive message log ($VMFREC $MSGLOG). Correct any problems before you go on. For information about handling specific error messages, refer to *VM/ESA: System Messages and Codes*, or use online help.

Step 4. Apply the New Service

1. Apply New Service

Command	Explanation
vmfapply ppf 5696234E {HLASM|HLASMSFS}	Command to apply the service you received in “Step 3. Receive the New Service” on page 86 The version vector table (VVT) is updated with all service parts and all necessary AUX files are generated on the Alternate Apply disk. Use HLASM if High Level Assembler is installed on minidisk. Use HLASMSFS if High Level Assembler is installed in SFS.

2. Review apply message log

Command	Explanation
vmfview apply	Command to review the apply message log ($VMFAPP $MSGLOG). Correct any problems before going on. For information about handling specific error messages, refer to *VM/ESA: System Messages and Codes*, or use online help.

Step 5. Reapply Local Service

Do this step only if you received the following message VMFAPP2120W during the VMFAPPLY step.

1. Reapply any local modifications *before* building the serviced High Level Assembler. If a local modification exists for ASMADOPT or one is planned refer to Appendix E, "Local Modification Procedures (VM)" on page 211. For further information refer to *VM/ESA Service Guide*.
2. If using the *VM/ESA Service Guide*, follow the steps that are applicable to your local modification. Then return to this *Installation and Customization Guide* to continue with the next step, updating the build status table.
3. For the process in the *VM/ESA Service Guide* make the following substitutions:

   The following substitutions need to be made:

   - `esalcl` should be `5696234E`
   - `esa` should be `5696234E`
   - *compname* should be `HLASM` or `HLASMSFS` (minidisk or SFS)
   - *appid* should be `5696234E`
   - *fm-local* should be the filemode of disk 2C2
   - *fm-applyalt* should be the filemode of disk 2A6
   - *outmode localmod* should be `outmode localsam`
   - substitute your PPF override in all commands requiring the PPF name.

Step 6.Update Build Status Table

Command	Explanation
vmfbld ppf 5696234E {HLASM|HLASMSFS} (status	Command to update the Build Status Table. Use `HLASM` if High Level Assembler is installed on minidisk. Use `HLASMSFS` if High Level Assembler is installed in SFS.

Options:

STATUS Identifies build requirements.

If service has been applied to the source product parameter file (filetype of $PPF) then carry out the instructions below before going onto "Step 7. Rebuild Serviced Parts (Objects)" on page 90.

If the $PPF files have been serviced you will get the following prompt:

```
VMFBLD2185R The following source product parameter files have been
serviced:
VMFBLD2185R 5696234E $PPF
VMFBLD2185R When source product parameter files are serviced, all
            product parameter files built from them must be recompiled
            using VMFPPF before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product
            parameter files built to your A-disk and exit VMFBLD so
            you can recompile your product parameter files with VMFPPF.
VMFBLD2185R Enter one (1) to continue only if you have already
            recompiled your product parameter files with VMFPPF.
```

If you select `0` then the following recompiles the product parameter files (PPF).

If you select `1` you should have previously compiled your product parameter files. If `1` has been selected then go onto "Step 7. Rebuild Serviced Parts (Objects)" on page 90.

1. Indicate $PPF file needs to be compiled

Command	Explanation
`0`	Enter a 0 and continue with the following commands.

   The message

   `VMFBLD2188I Building 5696234E $PPF on 191 (A) from level $PF`*nnnnn*

   is displayed.

2. Compile the product parameter file

Command	Explanation
`vmfppf 5696234E *`	If you have your own PPF override, use your PPF name instead of 5696234E.

3. Copy the product parameter file to software inventory disk

Command	Explanation
`copyfile 5696234E $PPF a = = d (olddate replace`	**Do not** use your own PPF name in place of 5696234E for the COPYFILE commands.

   Options:

   OLDDATE Uses the date and time on each input file as the date and time of the last update of each corresponding output file.

   REPLACE Causes the output file to replace an existing file with the same file identifier.

4. Erase product parameter file from A disk

Command	Explanation
erase 5696234E $PPF a	**Do not** use your own PPF name in place of 5696234E for the ERASE commands.

5. Update build status table

Command	Explanation
vmfbld ppf 5696234E {HLASM|HLASMSFS} (status	VMFBLD updates the build status table. Use HLASM if High Level Assembler is installed on minidisk. Use HLASMSFS if High Level Assembler is installed in SFS.
1	When you receive the *VMFBLD2185R* prompt, enter a 1 to continue.

Options:

STATUS Identifies build requirements.

6. Review the build message log

Command	Explanation
vmfview build	Command to review the build status messages and see which objects need to be built.

Step 7. Rebuild Serviced Parts (Objects)

1. Rebuild serviced parts

Command	Explanation
vmfbld ppf 5696234E {HLASM|HLASMSFS} (serviced	Command to rebuild serviced parts. If you receive message VMFSBR2000I then you need to rebuild the segment holding High Level Assembler code. **Note:** If your Software Inventory disk (51D) is not owned by the MAINT user ID then make sure the VMSESE PROFILE reflects the correct owning user ID. Use HLASM if High Level Assembler is installed on minidisk. Use HLASMSFS if High Level Assembler is installed in SFS.

Options:

SERVICED Identifies build requirements and builds those objects flagged as SERVICED in the service-level build status table.

2. Review build message log

Command	Explanation
vmfview build	Command to review the build message log ($VMFBLD $MSGLOG). Correct any problems before going on. For information about handling specific error messages, refer to *VM/ESA: System Messages and Codes*, or use online help.

Step 8. Verify the Service

After you have applied all the files on the service tape, run the installation verification EXEC to ensure that the product functions properly.

Command	Explanation
V5696234 noseg	Assembles the sample ASMASAMP. The *noseg* parameter will cause High Level Assembler to use its code only from disk and not in any segment.

Step 9. Place the Service into Production

1. Remove the High Level Assembler saved segment. This step is only required if you have a High Level Assembler saved segment in use. If this is not applicable proceed to substep 6 on page 93.

Command	Explanation
q nss all	Command to display list of system data files that contain named saved systems (NSS) and saved segments.

2. Remove system data file.

Command	Explanation
purge nss name asmapseg	Command to remove system data file containing the ASMAPSEG physical saved segment.

3. Re-create the High Level Assembler saved segment.
   a. Prepare to build the segment.
      1) ESA Mode considerations

      If you are placing the segment above 16MB, then prior to the IPL of CMS, use the SET MACHINE command to change the architecture mode to ESA.

Command	Explanation
`set machine esa`	Option ESA designates ESA/390 mode.

2) Set storage and ipl virtual machine

Command	Explanation
`def stor 24M`	Issue this CP command. It defines the virtual machine storage to be greater than where the segment is loaded. The default VM size for this user ID is 8M. If defining a saved segment above 16MB then the virtual machine storage should be above 16MB. In general the storage should be large enough to hold the segment. If you are using SFS in this userid, even if only for the A disk, and you are placing the ASMAPSEG segment below 16Mb at 920-9D0 then use a 'DEF STOR 24M' command. This will allow CMS control blocks not to interfere with the storage area to be used by the High Level Assembler saved segment.
`ipl cms parm clear nosprof instseg no mtseg no` ***** DO NOT press ENTER at the VM READ*****	This command clears your virtual machine. This command bypasses the execution of the system profile (SYSPROF EXEC) and without loading the installation saved segment (CMSINST). If using SFS, append to the **IPL CMS** command **filepool vmsys**. This assumes VMSYS is the default filepool.
`access (noprof`	Bypass the execution of the PROFILE EXEC.

3) Access disks

Command	Explanation
`access 5e5 b`	Access the VMSES/E code.
`link MAINT 51d 51d mr` `access 51d d`	Establish write access the software inventory disk.

b. Issue VMFBLD command to build the High Level Assembler segment.

Command	Explanation
`vmfbld ppf segbld esasegs segblist ASMAPSEG (serviced`	

Options:

SERVICED Identifies build requirements and build those objects flagged as SERVICED in the service-level build status table.

4. Review build message log

Command	Explanation
vmfview build	Command to review the build message log ($VMFBLD $MSGLOG). Correct any problems before going on. For information about handling specific build messages, refer to *VM/ESA: System Messages and Codes*, or use online help.

5. Update the CMS System disk with the SYSTEM SEGID file.

   The system segment identification file must reside on the CMS system disk (usually file mode S) and must be named SYSTEM SEGID so that it is available to CMS at initialization time. This allows CMS to recognize the logical saved segment name.

   The file must also be copied to the test CMS system disk to prevent backlevelling during application of service.

   The following functions can only be performed by authorized user IDs. Contact your system support personnel, if required.

Command	Explanation
rel d (det	Release the software inventory disk.

   Log on to Maint user ID.

Command	Explanation
acc 190 t	Access the production CMS system disk (usually MAINT 190).
acc 490 v	Access the test CMS system disk (usually MAINT 490).
acc 51d d	Access the software inventory disk.
copyfile system segid d system segid t2 (replace olddate	Copy file to production CMS system disk.
copyfile system segid d system segid v2 (replace olddate	Copy file to test CMS system disk.

   Options:

   OLDDATE Uses the date and time on each input file as the date and time of the last update of each corresponding output file.

   REPLACE Causes the output file to replace an existing file with the same file identifier.

6. Move service into production.

You have now built the segment.

a. Log on to MAINT if you plan to put High Level Assembler general-use code on the 'Y' disk (MAINT's 19E disk). Or log on to the user ID of the owner of the disk that will contain the production level of the High Level Assembler code.

b. Link to High Level Assembler code and access MAINT's Y disk.

   Choose which access is required depending on whether you have used a minidisk or SFS.

Command	Explanation
`link P696234E 29e 29e rr`	
`access 29e e`	Access High Level Assembler code on a minidisk.
`access 19e f`	Access CMS's Y disk.

Command	Explanation
`access vmsys:P696234E.HLASM.TBUILD e`	Access High Level Assembler code on a SFS directory.
`access 19e f`	Access CMS's Y disk.

c. Move High Level Assembler to the Y disk.

   If you have created a saved segment for High Level Assembler, then the only modules that should be copied by the VMFCOPY command are:

   - ASMAHL
   - ASMA90
   - ASMADOPT
   - ASMAINFO

   Therefore the VMFCOPY command must be modified to only copy the four modules which do not reside in the saved segment.

   If you have not created a saved segment then all the High Level Assembler modules should be placed onto the Y disk using the VMFCOPY command.

   In both cases (saved segment or no saved segment), the ASMAMAC MACLIB should be copied.

Command	Explanation
`vmfcopy * MODULE e = = f2 (prodid 5696234E%HLASM olddate replace`	Please refer to the previous note about which modules should be copied if you have created a logical saved segment.
`vmfcopy * MACLIB e = = f2 (prodid 5696234E%HLASM olddate replace`	

Options:

OLDDATE Uses the date and time on each input file as the date and time of the last update of each corresponding output file.

REPLACE Causes the output file to replace an existing file with the same file identifier.

If you do not want to use the Y disk for general use code, log on as the owner of the disk where you will put the production level of the High Level Assembler code.

The VMFCOPY command updates the VMSES PARTCAT file on the 19E disk.

d. Migrate from previous Assemblers

If Release 1 of High Level Assembler has been removed then the ASMAHL MODULE must be copied to the Y disk as HLASM MODULE. This will allow VM service to function correctly.

If you want to migrate from previous assemblers, you should consider the following:

- HLASM MODULE in High Level Assembler Release 1 can be replaced by the ASMAHL MODULE in High Level Assembler subsequent releases.
- HASM MODULE in Assembler H can be replaced by the ASMAHL MODULE in High Level Assembler subsequent releases. However, the ASMAHL MODULE does not support the *NUM* and *STMT* options.
- IEV90 MODULE in Assembler H can be replaced by the ASMA90 MODULE in High Level Assembler. If this is done the ASMAHL MODULE should be copied as HASM as well.

You can then copy them to the Y disk.

Command	Explanation
`vmfcopy ASMAHL MODULE e HLASM = f2 (prodid 5696234E%HLASM olddate replace`	
`vmfcopy ASMAHL MODULE e HASM = f2 (prodid 5696234E%HLASM olddate replace`	
`vmfcopy ASMA90 MODULE e IEV90 = f2 (prodid 5696234E%HLASM olddate replace`	

Options:

OLDDATE Uses the date and time on each input file as the date and time of the last update of each corresponding output file.

REPLACE Causes the output file to replace an existing file with the same file identifier.

e. Move User Exits (Optional)

User exits are supplied as sample assembler source on the LOCAL disk. If these exits are to be made generally available, they can be moved (ASMAX... ASSEMBLE) to the MAINT 19E disk, or the LOCAL disk can be made available to the users.

f. Update saved Y disk file directory

This will update the shared Y-STAT (the saved Y disk file directory). If you have created a saved segment, it will also update the S-STAT (the saved S disk file directory) to reflect the updated SYSTEM SEGID file.

1) Define named saved system

This will create a named saved system for CMS.

Command	Explanation
`vmfsetup esa cms`	Allows access to the SAMPNSS EXEC which resides on MAINT's 193 disk.
`sampnss cms`	Create named saved system for CMS.

2) Resave the CMS system

Command	Explanation
`ipl 190 clear parm savesys cms`	Resave CMS saved system to reflect the change to the S-STAT (the saved S disk file directory).

Removing High Level Assembler

Use the VMFINS DELETE command to remove High Level Assembler from your system. The process to do this is covered in the chapter concerning product deletion. Please refer to *VMSES/E Introduction and Reference*.

To Report a Problem with High Level Assembler

To assist with reporting any difficulties reference should be made to the diagnostic process as shown in Chapter 16, “Isolating the Problem” on page 135.

Report any difficulties you have using this product to your IBM Support Center. In the United States, if an APAR is required, submit the data to the location identified in the *Field Engineering Programming System General Information* manual (PSGIM), G229-2228.

Table 37 identifies the component ID (COMP ID) for High Level Assembler.

Table 37. Component IDs

COMP ID	Component Name	REL
569623400	VM HIGH LEVEL ASM	340

Obtaining Service Information

Preventive Service Planning (PSP) information is continually updated as fixes are made available for problems. Check with your IBM Support Center or use IBMLink (ServiceLink) to see whether there is additional PSP information you need. To obtain this information, specify the following UPGRADE and SUBSET values: `HLASM140` and `HLASMVM340`.

On VSE: Planning for, Installing, Customizing, and Maintaining High Level Assembler

Chapter 11. Planning for Installing High Level Assembler on VSE 101
Worksheet: Planning for Installing High Level Assembler on VSE 101
What You Receive with High Level Assembler 102
Distribution Media 102
Basic Material 102
Optional Material 102
Cumulative Service Tape 103
Program Publications and Softcopy 103
Program Source Materials 103
What You Need to Install High Level Assembler 103
Required and Optional Software 103
DASD and Other Storage Required 103
Planning Where to Install High Level Assembler 104
Selecting National Language Support 104
Program Support 105
Program Services 105
Preventive Service Planning 105
Statement of Support Procedures 105
Program and Service Level Information 105
Program Level Information 106
Service Level Information 106
Publications Useful during Installation 106

Chapter 12. Planning for Customizing High Level Assembler on VSE . . . 107
Deciding Whether and What to Customize 107
Planning to Customize High Level Assembler Options 107
Choices to Make Now 107
Planning to Install High Level Assembler into the Shared Virtual Area (SVA) . 109
Why Do It 109
Choices to Make Now 109

Chapter 13. Installing High Level Assembler on VSE 111
Overview of Installation 111
Checklist for Installing High Level Assembler 111
Step 1: Back Up the Original System 111
Step 2: Allocate Space for the Library (Omit if Using the Default Sublibrary) . . 112
Step 3: Install High Level Assembler 113
Method 1: Install High Level Assembler Using the Interactive Interface with Stacked tape 114
Condition Code and Messages 116
Method 2: Install High Level Assembler Using the Interactive Interface with Non-Stacked tape 116
Condition Code and Messages 117
Method 3: Install High Level Assembler Using a Batch Job 117
Step 4: Verify the Installation of High Level Assembler 119

Chapter 14. Customizing High Level Assembler on VSE 123
Changing Default Options 123
Relink High Level Assembler Phase ASMADOPT 124

Placing High Level Assembler into the Shared Virtual Area (SVA) 125

Chapter 15. Maintaining High Level Assembler on VSE 127
Re-installing High Level Assembler 127
Applying Service Updates 127
What You Receive 127
Checklist for Applying Service 128
Step 1. Check Prerequisite APARs or PTFs 128
Step 2. Backup Original System 128
Step 3. Apply Service 129
Method 1: Apply Service Using the Interactive Interface 129
Method 2: Apply Service Using a Batch Job 130
Step 4. Run the installation verification program (IVP) 130
Removing High Level Assembler 130
To Report a Problem with High Level Assembler 131
Obtaining Service Information 131

Chapter 11. Planning for Installing High Level Assembler on VSE

This section contains the following planning information to help you properly install High Level Assembler on VSE:

- Worksheet
- What you receive with High Level Assembler
- Choosing required and optional software
- Verifying that you have enough DASD storage
- Deciding where to install
- Planning which national language support you require
- Checking service updates

Worksheet: Planning for Installing High Level Assembler on VSE

Before you begin the installation you should:

__ 1. Determine which of the following you are installing High Level Assembler from:

   __ A stacked tape (with one or more products besides High Level Assembler)

   __ A tape with just High Level Assembler on it

   See “What You Receive with High Level Assembler” on page 102.

__ 2. Determine the product parts to be installed:

   __ COMPID *569623400*

   __ Feature number ____

   __ Tape label *unlabeled*

   See “Basic Material” on page 102.

__ 3. Verify that required software (and optional software, if appropriate) is at the level needed. See “What You Need to Install High Level Assembler” on page 103.

__ 4. Verify that adequate storage is available. See “DASD and Other Storage Required” on page 103.

__ 5. Determine how you are going to install High Level Assembler:

   __ Using Interactive Interface.

   __ Using a batch installation job.

   See “Planning Where to Install High Level Assembler” on page 104.

__ 6. Determine which of the following you want to install on:

   __ Default library and sublibrary

   __ A different library and sublibrary

   If using a different library and sublibrary, verify that space is sufficient. See “Planning Where to Install High Level Assembler” on page 104.

__ 7. Decide what national language support you want for diagnostic messages. See “Selecting National Language Support” on page 104.

__ 8. Check on latest service updates needed. See “Program Support” on page 105.

What You Receive with High Level Assembler

You receive one of the following when you order High Level Assembler:

COMPIDs	Feature Number	System Name
569623400	5851	VSE/ESA
569623400	5852	VSE/ESA
569623400	5700	VSE/ESA

Distribution Media

High Level Assembler is distributed on one of the following:

- Unlabeled 9-track magnetic tape written at 6250 BPI
- 3480 tape cartridge
- 4mm DAT tape cartridge

The tape or cartridge contains all the programs and data needed for installation.

Basic Material

Table 38 describes the program tape or cartridge. Table 39 describes the file content of the program tape or cartridge. VSE/ESA uses the Maintain System History Program (MSHP) to install this product.

Table 38. Basic Material: Program Tape

Medium	Feature Number	Physical Volume	External Label Identification	VOLSER
6250 tape	5851	1	HLASM V1R4 for VSE	unlabeled
3480 cart.	5852	1	HLASM V1R4 for VSE	unlabeled
4mm DAT	5700	1	HLASM V1R4 for VSE	unlabeled

Table 39. Program Tape: File Content

File	Description
1	Header file containing High Level Assembler copyright statement
2	Backup file ID “HLASM.....1.4.0” followed by a MSHP System History File
3	High Level Assembler library file containing the production sublibrary
4	Tape mark
5	End of backup record
6	Tape mark

Optional Material

There are no optional machine-readable materials for High Level Assembler.

Cumulative Service Tape

You might receive an additional tape containing cumulative service with your order. The PTFs on this tape have not yet been incorporated into this release.

Program Publications and Softcopy

This section identifies the basic and optional publications for High Level Assembler.

One printed copy of the following manuals is included with the product:

- *HLASM Licensed Program Specifications* GC26-4944
- *HLASM Installation and Customization Guide* SC26-3494
- *HLASM Language Reference* SC26-4940
- *HLASM Programmer's Guide* SC26-4941
- *HLASM General Information* GC26-4943

For a list of books for related products, see “High Level Assembler Publications” on page 223.

Program Source Materials

There are no source materials available for High Level Assembler.

What You Need to Install High Level Assembler

The following sections identify the system requirements for installing High Level Assembler.

Required and Optional Software

This section describes the environment required to install and use High Level Assembler.

High Level Assembler runs on VSE with the required licensed programs listed in Table 40. **You should install all licensed programs with the minimum release listed or with any subsequent release**.

Table 40. Required Programs

Required Licensed Program	Minimum Version Supported
VSE/ESA	Version 1 Release 4
VSE/ESA	Version 2 Release 3

If using High Level Assembler within ICCF, then ICCF has a prerequisite of PTF UD49163 (Production) and PTF UD49164 (Generation).

DASD and Other Storage Required

The DASD storage requirements of High Level Assembler must be added to the storage required by other programs having data in the same library. An estimate of required space is the data set's current allocation plus the storage required by High Level Assembler.

Auxiliary storage is needed for the system history file, for the library and for an intermediate work file (IJSYS03) on a direct-access storage device such as 3350, 3375, 3380, 3390, 9345 or FBA.

At a minimum, the target library must have free the number of library blocks shown in Table 41 on page 104. A library block equals 1 kilobyte. For example, 10 library blocks equal 10,240 bytes.

Table 41 also allows for space for service application.

Table 41. Library Requirements

Product Tapefile-id	LIBR BLKS	3350 CYL	3375 CYL	3380 CYL	3390 CYL	9345 CYL	FBA BLKS
HLASM.....1.4.0	4000	9	14	9	9	9	8000

If you plan to have this product in a separate MSHP history file refer to the Table 42 for MSHP DASD space requirements.

Table 42. MSHP Requirements

File	3350 CYL	3375 CYL	3380 CYL	3390 CYL	9345 CYL	FBA BLKS
IJSYSHF	1	2	1	1	1	900

Planning Where to Install High Level Assembler

The High Level Assembler default library is PRD1; the default sublibrary is BASE. All High Level Assembler installation jobs assume you are using sublibrary PRD1.BASE. If you decide to install High Level Assembler in a different library and sublibrary, you need to change some names in the installation jobs.

If you plan to install High Level Assembler to an existing PRD1.BASE sublibrary, make sure there is enough free space to accommodate the additional library blocks.

To check the space, list the directory information of the PRD1 library, using the LISTDIR command of the LIBR program. Make sure there are sufficient library blocks in the free space.

You can use Interactive Interface to install High Level Assembler, or the documented sample batch job.

Selecting National Language Support

When installing High Level Assembler, you can choose which language you want to use for diagnostic messages. Languages available are English, German, Spanish and Japanese. English is also available in mixed case or uppercase.

The following combinations are possible:

English Uppercase Diagnostic messages and listing headings printed in uppercase English.

English Mixed Case Diagnostic messages and listing headings printed in mixed case English.

German Diagnostic messages in German and listing headings printed in mixed case English.

Japanese Diagnostic messages in Kanji and listing headings printed in uppercase English.

Spanish Diagnostic messages in Spanish and listing headings printed in mixed case English.

Program Support

This section describes the IBM support available for High Level Assembler.

Program Services

Contact your IBM representative for specific information about available program services.

Preventive Service Planning

Before installing High Level Assembler, you should review the current Preventive Service Planning (PSP) information.

PSP Buckets are identified by UPGRADEs, which specify product levels, and SUBSETs, which specify the CLCs for a product level. The UPGRADE and SUBSET values for High Level Assembler are:

Table 43. PSP Upgrade and Subset ID

UPGRADE	SUBSET	Description
HLASM140	HLASMVSE489	HLASM VSE

Statement of Support Procedures

Report any difficulties you have using this program to your IBM Support Center. If an APAR is required, the Support Center will provide the address to which any needed documentation can be sent.

Table 13 on page 12 identifies the component IDs (COMPID) for High Level Assembler.

Table 44. Component IDs

CLC	COMPID	Component Name	RETAIN Release
489	569623400	VSE HIGH LEVEL ASM	489

Program and Service Level Information

This section identifies the program and any relevant service levels of High Level Assembler. The program level refers to the APAR fixes incorporated into the program. The service level refers to the PTFs integrated. Information about the cumulative service tape is also provided.

Program Level Information

A list of APAR fixes against previous releases of High Level Assembler that have been incorporated into this release is shown in Appendix C, “High Level Assembler Service” on page 207.

Service Level Information

No PTFs against this release of High Level Assembler have been incorporated into the product tape.

Publications Useful during Installation

The publications listed in Table 45 may be useful during the installation of High Level Assembler. To order copies, contact your IBM representative.

Table 45. Publications Useful During Installation

Publication Title	Form Number
VSE/ESA Administration	SC33-6505
VSE/ESA Guide to System Functions	SC33-6511
VSE/ESA Installation	SC33-6504
VSE/ESA Planning	SC33-6503
VSE/ESA System Control Statements	SC33-6513
VM/ESA System Messages and Codes Reference	SC24-5841

Chapter 12. Planning for Customizing High Level Assembler on VSE

This chapter provides information for planning the customization of High Level Assembler on VSE. It includes:

- Deciding whether and what to customize
- Planning to customize the IBM-supplied, default option values
- Planning to install High Level Assembler into the shared virtual area (SVA)

Deciding Whether and What to Customize

You need to consider whether the IBM-supplied values that come with High Level Assembler suit the needs of your installation. These values control such features as:

- Whether or not to suppress part of the assembly listing
- Whether or not to generate symbol cross-reference information

Make sure that High Level Assembler serves the needs of the application programmers at your installation. Confer with them while you evaluate the customization options for High Level Assembler, particularly those concerning High Level Assembler options that are also available to the application programmers. Doing so will ensure that the modifications you make best support the application programs being developed at your installation.

The information in this chapter helps you plan your customization. See Chapter 14, “Customizing High Level Assembler on VSE” on page 123 for the actual customization procedure.

Planning to Customize High Level Assembler Options

The High Level Assembler options should be reviewed to assess the required defaults for your site. A worksheet is provided for planning purposes.

Choices to Make Now

The following worksheet helps you plan and code the options appropriate for your installation. Appendix A, “High Level Assembler Options” on page 171 lists all the assembler options, the values that may be specified, and the IBM-supplied default value for each option. Review these default options and complete the worksheet by filling in the **Enter Selection** column.

Table 46 (Page 1 of 3). Worksheet: Options

Option	Enter Selection	IBM-Supplied Default	Description
ADATA	_____	NO	page 171
ADEXIT	_____	no exit specified	page 171
ALIGN	_____	YES	page 172
ALIGNWARN	_____	YES	page 172

Table 46 (Page 2 of 3). Worksheet: Options

Option	Enter Selection	IBM-Supplied Default	Description
BATCH	_____	YES	page 173
CODEPAGE	_____	047C	page 173
COMPAT	_____	NO	page 174
CONTWARN	_____	YES	page 175
DBCS	_____	NO	page 176
DECK	_____	NO	page 176
DELETE	_____	**No options deleted**	page 176
DSECT	_____	NO	page 177
DXREF	_____	YES	page 178
ESD	_____	YES	page 178
FLAG	_____	0	page 178
FOLD	_____	NO	page 179
GOFF	_____	NO	page 179
GOFFADATA	_____	NO	page 180
IMPLENWARN	_____	NO	page 180
INEXIT	_____	no exit specified	page 180
INFO	_____	NO	page 181
LANGUAGE	_____	EN	page 181
LIBEXIT	_____	no exit specified	page 182
LIBMAC	_____	NO	page 182
LIMIT	_____	NO	page 183
LINECOUNT	_____	60	page 183
LIST	_____	YES	page 183
MAP	_____	YES	page 184
MXREF	_____	SOURCE	page 185
OBJECT	_____	YES	page 185
OBJEXIT	_____	no exit specified	page 186
OPTABLE	_____	UNI	page 186
PAGE0WARN	_____	NO	page 187
PCONTROL	_____	NO	page 188
PESTOP	_____	NO	page 190
PROFILE	_____	NO	page 190
PROFMEM	_____	ASMAPROF	page 190
PRTEXIT	_____	no exit specified	page 191

Table 46 (Page 3 of 3). Worksheet: Options

Option	Enter Selection	IBM-Supplied Default	Description
PUSHWARN	_____	YES	page 191
RA2	_____	NO	page 192
RECORDINFO	_____	YES	page 192
RENT	_____	NO	page 193
RLD	_____	YES	page 193
RXREF	_____	YES	page 193
SIZE	_____	MAX	page 194
STORAGE	_____	BELOW	page 195
SUBSTRWARN	_____	YES	page 196
SYSPARMV	_____	none specified	page 196
TERM	_____	NO	page 196
TEST	_____	NO	page 197
THREAD	_____	YES	page 197
TRANSLATE	_____	NO	page 198
TRMEXIT	_____	no exit specified	page 198
USING0WARN	_____	YES	page 198
WARN	_____	15	page 199
XREF	_____	(SHORT,UNREFS)	page 201

Planning to Install High Level Assembler into the Shared Virtual Area (SVA)

High Level Assembler can utilize the benefits to be gained by placing it in the SVA.

Why Do It

Installing frequently used data (such as licensed programs) in the SVA has several advantages:

- Because several users can access the same physical storage, use of real storage is minimized.
- Using the SVA decreases the input/output rate and the DASD paging requirements, thereby improving the performance of the virtual machine.
- The elapsed time for jobs is shorter.

Choices to Make Now

Table 47 on page 110 gives the names of the High Level Assembler phases, their approximate sizes, and whether or not they are eligible for the SVA.

Table 47. High Level Assembler Phases

Member Name	Member Type	SVA Eligible	Approximate Size
ASMA90	PHASE	YES	10K
ASMA93	PHASE	YES	370K
ASMADOPT	PHASE	YES	2K
ASMAMUE	PHASE	YES	21K
ASMAOESA	PHASE	YES	32K
ASMAOXA	PHASE	YES	24K
ASMAO370	PHASE	YES	24K
ASMAODOS	PHASE	YES	14K
ASMALTAS	PHASE	YES	1K
$SVAASMA	PHASE	NO	1K
ASMAMDE	PHASE	YES	24K
ASMAMES	PHASE	YES	24K
ASMAMJP	PHASE	YES	23K
ASMAINFO	PHASE	YES	19K
ASMAOEME	PHASE	YES	42K
ASMA0474	PHASE	YES	2K
ASMA0475	PHASE	YES	2K
ASMA0476	PHASE	YES	2K
ASMA0477	PHASE	YES	2K
ASMA0478	PHASE	YES	2K
ASMA0479	PHASE	YES	2K
ASMA047A	PHASE	YES	2K
ASMA047B	PHASE	YES	2K
ASMA047C	PHASE	YES	2K
ASMADOP	PHASE	YES	34K

Phases are placed in the SVA using the SET SDL command. If you have not previously placed High Level Assembler in the SVA, you need to make sure that you have enough free space to accommodate the additional phases.

The maximum number of system directory list (SDL) entries is specified during IPL using the SVA command. Check this number to see if the addition of High Level Assembler phases causes it to be exceeded.

Chapter 13. Installing High Level Assembler on VSE

This chapter describes the installation method and the step-by-step procedures you use to install and activate the functions of High Level Assembler.

Overview of Installation

You install this release of High Level Assembler by using the Maintain System History Program (MSHP).

Checklist for Installing High Level Assembler

Table 48 lists the steps and associated jobs for installing High Level Assembler. The remaining sections in this chapter describe each step. You can use Table 48 as a checklist.

Table 48. Summary of Steps for Installing High Level Assembler

Step	Description	Installation Job	page
__ 1	Back up the original system.	—	111
__ 2	Allocate space for the library. (Omit if using the default sublibrary.)	ASMADEF	112
__ 3	Install High Level Assembler.		113
	Method 1. Install High Level Assembler using the Interactive Interface with High Level Assembler on an Optional Products (stacked) tape.	—	114
	Method 2. Install High Level Assembler using the Interactive Interface with High Level Assembler as the only product on the (non-stacked) tape.	—	116
	Method 3. Install High Level Assembler using a batch job.	ASMAINST	117
__ 4	Verify the installation of High Level Assembler.	ASMAIVPS	119

Step 1: Back Up the Original System

Make a backup copy of your current High Level Assembler library or the library you intend to install High Level Assembler into, and the system history file.

For information about backing up libraries and the system history file, see *VSE/ESA System Control Statements*.

Step 2: Allocate Space for the Library (Omit if Using the Default Sublibrary)

By default, High Level Assembler is installed into the PRD1.BASE sublibrary for VSE/ESA. If you decide to install High Level Assembler into a sublibrary other than PRD1.BASE then proceed with this step.

Decide where to allocate space for the High Level Assembler sublibrary. Identify, on the disk volume (or volumes) to be used for the library, suitable areas of free space. To do this, list the volume table of contents (VTOC) of the disk or disks to be used.

Choose one of the following jobs to list the VTOC:

1. Use the LVTOC utility program The sample job shown in Figure 18 shows the JCL needed to list the VTOC for the volume with serial number `SYSWK1`.

```
// JOB ASMAVTOC    LIST VOLUME TABLE OF CONTENTS
// ASSGN SYS004,DISK,TEMP,VOL=SYSWK1,SHR
// ASSGN SYS005,SYSLST
// EXEC LVTOC
/*
/&
```

Figure 18. Job to List the Contents of a DASD Volume

2. Use the DITTO utility program

   As an alternative to using the system utility LVTOC, DITTO's Display VTOC (DVT) may be used. Figure 19 shows a sample job.

```
// JOB ASMADITT    LIST VOLUME TABLE OF CONTENTS
// UPSI 1
// ASSGN SYS001,cuu
// EXEC DITTO,SIZE=512K
$$DITTO DVT INPUT=SYS001,SORTBY=EXTENT
$$DITTO EOJ
/*
/&
```

Figure 19. Job to List the Contents of a DASD Volume

Use the disk space selected for High Level Assembler in the LIBR installation job to allocate the VSE Librarian library in the sample job shown in Figure 20 on page 113.

```
// JOB ASMADEF
*  CREATE A LIBRARY FOR THE High Level Assembler
// OPTION LOG
*  Label for the High Level Assembler
*  Library                                          1
// DLBL HLASM,'HLASM.LIBRARY',99/365,SD
// EXTENT SYS002,SYSWK1,,,rtrk,ntrk
// ASSGN SYS002,DISK,VOL=SYSWK1,SHR
* ----------------------------------------
*  Define the High Level Assembler Library         2
* ----------------------------------------
// EXEC LIBR
 DELETE LIB=HLASM
 DEFINE LIB=HLASM
/*
/&
```

Figure 20. Job to Allocate the High Level Assembler library space.

In area **1** change the *filename* (HLASM in the example) and *file-id* (HLASM.LIBRARY in the example) of High Level Assembler to suit your installation. Points to consider are:

- The variable *ntrk* indicates the number of tracks required; this is the size of the extent needed in Figure 20.
- If you are using an FBA device, such as an IBM 3370, or a CKD device such as an IBM 3380, see the information in “DASD and Other Storage Required” on page 103 for the number of blocks/tracks required.
- Change *rtrk* operand to a value to suit the type of device being used for the current installation.
- The variable *rtrk* represents the start position of the extent.

The Librarian job step in area **2** includes a DELETE statement before the DEFINE statement so the job can be rerun. This means the following messages are issued when the job runs for the first time; however these may be treated as informational and ignored. The job continues to allocate the library.

The messages are:

```
L101I  LIBRARY HLASM DOES NOT EXIST
L027I  ABNORMAL END DURING DELETE COMMAND
       PROCESSING
L113I  RETURN CODE OF DELETE IS 8
```

Step 3: Install High Level Assembler

You can install High Level Assembler using either the Interactive Interface of VSE or a batch installation job.

Method 1: Install High Level Assembler Using the Interactive Interface with Stacked tape

Carry out the following tasks:

1. Mount the High Level Assembler tape on an available tape drive.
2. Library Labels

   If the product is not being installed in the default library then the library to contain High Level Assembler must have its disk label in the label information area.
3. Log on to the Interactive Interface

   To install High Level Assembler using the Interactive Interface, log on to the VSE/ESA Interactive Interface as the system administrator. (If you would like information about the functions of the Interactive Interface, refer to *VSE/ESA Administration*.)

   In the following menus, enter the highlighted items that appear after the ===> symbol.
4. Mount the High Level Assembler tape on an available tape drive.
5. In the **VSE/ESA FUNCTION SELECTION** menu, select:
   ===> ***1*** (Installation)
6. In the **INSTALLATION** menu:

   You received High Level Assembler as a stacked tape, which contains one or more optional products, select:
   ===> ***1*** (Install Programs - Stacked V2 Format)
7. In the **INSTALL PROGRAMS - STACKED V2 FORMAT** menu, select:
   ===> ***1*** (Prepare for Installation (Stacked Tapes Only))
8. In the **PREPARE FOR INSTALLATION (STACKED TAPES ONLY)** menu, select:
   ===> ***cuu***
   (the address of the tape drive where you mounted the distribution tape)
9. In the **JOB DISPOSITION** menu:

   Make any changes required and press Enter to submit the job.
10. Respond to console messages

    When the job starts it asks if the tape is ready. After the tape is scanned it asks if further tapes are to be read. Please refer to Figure 21 on page 115 for examples of the console messages.

```
01 BG 000 IESI0091I PLEASE MOUNT TAPE LABELLED "VSE OPTIONAL TAPE NUMBER
02*BG 000 IESI0092A MOUNT ON TAPE DRIVE 580 . WHEN READY, REPLY "END/ENTER"
03*BG-000
04 0
05*BG 000 IESI0090A ARE THERE ANY MORE OPTIONAL PROGRAM TAPES? YES/NO
06*BG-000
10 0 no
11 BG 000 EOJ INSPRE    MAX.RETURN CODE=0000
```

Figure 21. Console Messages

The output listing from this job gives a list of the optional programs on the distribution tape with program identifiers and recommended library sizes. The tapefile identifier for High Level Assembler is `HLASM.....1.4.0`.

The program identifiers of the optional programs on the distribution tape are also automatically entered on the **INSTALL PRODUCT(S) FROM TAPE** menu.

11. To return to the **VSE/ESA FUNCTION SELECTION** menu, enter:
    ===> ***1*** (Installation)
12. In the **INSTALLATION** menu, select:
    ===> ***1*** (Install Programs - Stacked V2 Format)
13. In the **INSTALL PROGRAMS - STACKED V2 FORMAT** menu, select:
    ===> ***2*** (Install Product(s) from Tape)
14. In the **INSTALL PRODUCT(S) FROM TAPE** menu:

    Enter **1** (install) in the OPT field against the tapefile identifier
    HLASM.....1.4.0 (High Level Assembler)
    and **2** (skip installation) against any other optional products you do not intend to install at this time.

    If you are running on a VSE/ESA Version 1 Release 4, ensure you select a partition (dynamic or static) which is less than 16MB in size.

    If you did not use the default library PRD1.BASE, enter the name of your library and sublibrary on this screen. The DLBL and EXTENT information for this library should already be in the label information area.

    Press PF5 to generate the installation job.
15. Retain products list

    Decide if you want to keep the product list previously generated from LIBR utility scan of the product tape.
16. In the**VSE/ESA INSTALL PRODUCT(S) TAPE SPECIFICATION** menu, enter:

    ===> ***cuu***
    (the address of the tape drive where you mounted the
    High Level Assembler tape)

17. In the **JOB DISPOSITION** menu:

    Make any changes required and press ENTER to submit the job to install High Level Assembler.

18. Respond to console messages

    Confirm prompt when asking for tape to install product.

Condition Code and Messages

If you do not receive a condition code of 0:

1. Check the list output for error conditions.
2. See *VSE/ESA System Control Statements* for corrective action.
3. Correct the error.
4. Rerun the job.
5. Recheck the condition code.

Method 2: Install High Level Assembler Using the Interactive Interface with Non-Stacked tape

Carry out the following tasks:

1. Log on to the Interactive Interface

   To install High Level Assembler using the Interactive Interface, log on to the VSE/ESA Interactive Interface as the system administrator. (If you would like information about the functions of the Interactive Interface, refer to *VSE/ESA Administration*.)

   In the following menus enter the highlighted items that appear after the ===> symbol.

2. Mount the High Level Assembler tape on an available tape drive.
3. In the **VSE/ESA FUNCTION SELECTION** menu, select:
   ===> ***1*** (Installation)

4. In the **INSTALLATION** menu:

   You received High Level Assembler as a stacked tape, which contains one or more optional products, select:
   ===> ***2*** (Install Programs - Non-stacked V2 Format or V1 Format)

5. In the **INSTALL PROGRAMS - NON-STACKED V2 OR V1 FORMAT** menu select:
   ===> ***HLASM... ...1.4.0*** (Tapefile-id)

   If you are running on a VSE/ESA Version 1 Release 4, ensure you select a partition (dynamic or static) which is less than 16MB in size.

   If you did not use the default library PRD1.BASE, enter the name of your library and sublibrary on this screen. The DLBL and EXTENT information for this library should already be in the label information area.

   Press PF5 to generate the installation job.

6. In the **VSE/ESA INSTALL PRODUCT(S) TAPE SPECIFICATION** menu, enter:

   ===> ***cuu***
   (the address of the tape drive where you mounted the
   High Level Assembler tape)

7. In the **JOB DISPOSITION** menu:

   Make any changes required and press Enter to submit the job to install High Level Assembler.

8. Respond to console messages

   Confirm prompt when asking for tape to install product.

Condition Code and Messages

If you do not receive a condition code of 0:

1. Check the list output for error conditions.
2. See *VM/ESA System Messages and Codes Reference* for corrective action.
3. Correct the error.
4. Rerun the job.
5. Recheck the condition code.

Method 3: Install High Level Assembler Using a Batch Job

The batch installation job stream for installing High Level Assembler uses the `MSHP` system history file that already exists as part of the VSE system. This system history file might already be defined in the system standard labels; if not, make sure that `DLBL` and `EXTENT` statements, with the necessary information for the system history file, are included in the job stream.

Depending on how you request the High Level Assembler product you might receive different installation tapes. One could contain only the High Level Assembler product, the other might be a stacked tape containing one or more optional program products. The job shown in Figure 22 on page 118 handles both types of tape (stacked and non-stacked).

Create and tailor the following job stream, mount the distribution tape, and run the installation job.

Figure 22 on page 118 provides the JCL required to install High Level Assembler. Tailor this JCL to suit the requirements of your installation.

As many as five modifications might be required to tailor the JCL. The keys within Figure 22 on page 118 are explained individually and refer to the sections that accompany the JCL description.

```
// JOB ASMAINST
*  INSTALL THE High Level Assembler LIBRARY
// OPTION LOG
*  Label for High Level Assembler Library   1
*  Assign  install tape as SYS006           2
// ASSGN SYS006,cuu
// MTC REW,SYS006
* ---------------------------------------
*  This step installs High Level Assembler
*  from the distribution tape
*  using the VSE system history file        3
* ---------------------------------------
// EXEC MSHP,SIZE=900K,PARM='PIDSTACKED'
 INSTALL PROD FROMTAPE ID='HLASM......1.4.0' -
    PROD INTO=PRD1.BASE
/*
* ---------------------------------------
*  List the High Level Assembler Library     4
* -------------------------------------
// EXEC LIBR
 LISTDIR SUBLIB=PRD1.BASE  -
     OUTPUT=NORMAL -
     UNIT=SYSLST
/*
* ----------------------------------------
*  Retrace the High Level Assembler product  5
* ----------------------------------------
// EXEC MSHP,SIZE=900K
RETRACE COMPONENT IDENTIFIER=5696-234-00
/*
// MTC RUN,SYS006
/*
/&
```

Figure 22. Job to Install High Level Assembler.

1. Specify the Label Information

   In area **1**, if you are installing High Level Assembler into a sublibrary other than the default then insert `DLBL, EXTENT and ASSGN` information as specified in Figure 20 on page 113. The library name must match the name used in the allocation job in Figure 20 on page 113.

   There is no `DLBL` statement for the system history file. Typically it would have a permanent system standard label for this, with `IJSYSHF` as the file name. (`IJSYSHF` is the default file name that `MSHP` looks for in a label statement.)

2. Assign the Distribution Tape

   Assign the distribution tape in area **2** to logical unit `SYS006`. Replace *cuu* with the address of the tape drive on which to mount the distribution tape. Alternatively you may use the generic tape assignment:

   ```
   // ASSGN SYS006,TAPE
   ```

3. Install High Level Assembler

   Area **3** of the job calls `MSHP` to install High Level Assembler into the sublibrary identified on the `INTO` operand of the `INSTALL` statement. If you are installing

High Level Assembler into a sublibrary other than the default, then change the name of the sublibrary on the `INTO` operand of the INSTALL statement to reflect this sublibrary. For more information about the install options, see "Maintain System History Program (MSHP)" in *VSE/ESA System Control Statements*.

4. List the Directory Entries

   The step in area **4** of the job lists the directory entries of the sublibrary where High Level Assembler was installed. Remove this step if a directory list is not required. If you have installed High Level Assembler into a sublibrary other than the default, then the name of the sublibrary must be changed to reflect that used in Figure 20 on page 113.

   Entries for High Level Assembler have a four character prefix of ASMA to distinguish them from other products; there are three exceptions to this rule:

   - `HD234489.Z`
   - `$SVAASMA.PHASE`
   - `$SVAASMA.OBJ`

5. Retrace the High Level Assembler product in the system history file.

   The final step in area **5** of the job prints the component records from the system history file for High Level Assembler. Remove this step if a retrace listing is not required.

   If this job has to be run again, remember first to restore the system history file, which should have been backed up before running this install job, and second to run the library allocation step again, if applicable.

Step 4: Verify the Installation of High Level Assembler

The installation verification programs, `ASMASAM1.A` and `ASMASAM2.A` are sample programs provided on the distribution tape. It lets you check that your installation is successful by exercising representative features of High Level Assembler.

To run the installation verification program:

1. Convert edited macros to source format

   If you have a VSE/ESA Version 2 system bypass this substep and proceed to substep 2 on page 120.

   If you have a VSE/ESA Version 1 Release 3 system and you have previously setup the library exit for processing E-decks(refer to *VSE/ESA Guide to System Functions*), bypass this substep and proceed to substep 2 on page 120.

   Because the High Level Assembler cannot process an edited macro, you must de-edit certain macros before running the sample program. You can use the ESERV program to convert the edited macro back to source format.

   The macros that are used by the sample program are:

   - NOTE
   - WRITE
   - WTO

   The job shown in Figure 23 on page 120 converts these macros. The job uses the ESERV program to convert the macros to a punch file on disk. This

file is input to the LIBR program and the macros are cataloged as A source books.

```
// JOB ASMAMACR
*
*  SAMPLE JCL TO DE-EDIT & CATALOG REQUIRED MACROS FOR IVP
*
// SETPARM VOLUME='SYSWK2'               * VOLUME TO USE
// SETPARM START='????'                  * STARTING EXTENT TRK/BLK 1
// SETPARM LENGTH='0060'                 * LENGTH OF WORK FILE
*
*  DE-EDIT MACROS FOR SAMPLE ASSEMBLER PROGRAM
*
// LIBDEF *,SEARCH=IJSYSRS.SYSLIB                                    2
// DLBL IJSYSPH,'MACRO.PUNCH.OUT',0
// EXTENT SYSPCH,&VOLUME,1,0,&START,&LENGTH
ASSGN SYSPCH,DISK,VOL=&VOLUME,SHR
// OPTION DECK
// EXEC ESERV
   PUNCH E.NOTE,E.WTO,E.WRITE
/*
CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'MACRO.PUNCH.OUT',0
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=&VOLUME,SHR
// EXEC LIBR,PARM='ACCESS SUBLIB=lib.sublib'                         3
/*
CLOSE SYSIPT,SYSRDR
/*
/&
// JOB RESET
ASSGN SYSIPT,SYSRDR    IF 1A93D, CLOSE SYSIPT,SYSRDR
ASSGN SYSPCH,PUNCH     IF 1A93D, CLOSE SYSPCH,PUNCH
/*
/&
```

Figure 23. Job to Create Macros as A Source Books

a. In area 1 change the SETPARM values to those applicable to your installation.

b. In area 2 the macros are found in the IJSYSRS library.

c. In area 3 direct the macros (now in source format) to a user sublibrary. Do not place them in the IJSYSRS library, as this might cause problems with MSHP service on the edited macros.

2. Run the Sample Program

Figure 24 on page 121 shows the job ASMAIVPS.Z provided in the installed sublibrary. It runs the sample program from the same sublibrary. If you installed High Level Assembler in a sublibrary other than the default, modify the IBM-supplied JCL that runs the verification program.

Run the job listed in Figure 24 on page 121 to verify the success of your installation.

```
// JOB ASMAIVPS
*
*  SAMPLE JCL TO VERIFY INSTALLATION OF THE
*  High Level Assembler.
*
// LIBDEF *,SEARCH=(lib.sublib,PRD1.BASE)    1
// EXEC ASMA90,SIZE=ASMA90
     COPY ASMASAM1
     COPY ASMASAM2
/*
/&
```

Figure 24. Job to Verify the Success of Your Installation

In area **1**, specify the user library where the source format macros have been placed, followed by the library where the High Level Assembler resides.

If you are using the library exit for processing E-decks then modify the EXEC statement in Figure 24 to:

```
// EXEC ASMA90,SIZE=(ASMA90,128K),PARM='EXIT(LIBEXIT(EDECKXIT))'
```

A return code of 0 (zero) for the job indicates that the sample program completed successfully. The assembly of the sample program is the actual verification that the product is installed and functions correctly.

The program being assembled is not intended to be run; however, if it is run, it sets a return code of zero and returns to the caller.

Chapter 14. Customizing High Level Assembler on VSE

You can customize, or modify, High Level Assembler only after installing the product (as described in Chapter 13, “Installing High Level Assembler on VSE” on page 111). This chapter will cover:

1. Changing default options
2. Placing High Level Assembler into the Shared Virtual Area

Changing Default Options

If you decided to change any of the default High Level Assembler options when planning to customize High Level Assembler (see Chapter 12, “Planning for Customizing High Level Assembler on VSE” on page 107) you can change these options now.

A sample job `ASMAOPTV.Z` is provided to help you modify and assemble the IBM-supplied, default assembler options. It is shown in Figure 25 on page 124.

The ASMAOPT macro in `ASMAOPTV.Z` is used to specify the options you want to set as the defaults. Refer to “General Rules for coding the ASMADOPT ASSEMBLE file” on page 81 for more information on how to enter the options into the sample job. For more detailed information on these options see *HLASM Programmer's Guide*. To determine how to set the options, see Appendix A, “High Level Assembler Options” on page 171.

When the changes have been made, submit the job for assembly on your VSE system. The ASMAOPTV job assembles and catalogs a new ASMADOPT.OBJ module in the user sublibrary. When this job has completed successfully, continue with the section on re-linking High Level Assembler phase `ASMADOPT`, which completes the customization of the High Level Assembler options.

```
// JOB ASMAOPTV ASSEMBLE DEFAULT OPTIONS
*
// SETPARM VOLUME='SYSWK2'               * VOLUME TO USE           |
// SETPARM START='????'                  * STARTING EXTENT TRK/BLK | 1
// SETPARM LENGTH='0020'                 * LENGTH OF WORK FILE     |
*
*  Assemble High Level Assembler Default Options
*
// LIBDEF *,SEARCH=(PRD1.BASE)                                      2
// DLBL IJSYSPH,'ASSEMBLE.OUTPUT',0
// EXTENT SYSPCH,&VOLUME,1,0,&START,&LENGTH
ASSGN SYSPCH,DISK,VOL=&VOLUME,SHR
// OPTION DECK
// EXEC ASMA90
         PUNCH 'CATALOG ASMADOPT.OBJ REPLACE=YES'
         PRINT ON,GEN
         ASMAOPT
         END
/*
CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'ASSEMBLE.OUTPUT',0
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=&VOLUME,SHR
// EXEC LIBR,PARM='MSHP;ACCESS SUBLIB=lib.sublib'                   3
/*
CLOSE SYSIPT,SYSRDR
/*
/&
// JOB RESET
ASSGN SYSIPT,SYSRDR    IF 1A93D, CLOSE SYSIPT,SYSRDR
ASSGN SYSPCH,PUNCH     IF 1A93D, CLOSE SYSPCH,PUNCH
/*
/&
```

Figure 25. Sample JCL to Assemble the Default Options

In area **1**, change the SETPARM values to those applicable to your installation.

In area **2**, if High Level Assembler has been installed in a different sublibrary from the default, change this to reflect your sublibrary.

In area **3**, change the sublibrary to reflect a user sublibrary in which changes to the High Level Assembler default options will be cataloged. This should be in a user library to ensure that the original IBM-supplied defaults are not overwritten.

Relink High Level Assembler Phase ASMADOPT

The High Level Assembler phase ASMADOPT needs to be relinked to successfully change the default options. The sample JCL member ASMAOPTL.Z is provided, which relinks the High Level Assembler phase ASMADOPT (refer to Figure 26 on page 125).

```
// JOB ASMAOPTL   Linkedit Default Options Phase
*
*  link edit ASMADOPT
*
// LIBDEF *,SEARCH=(lib.sublib,PRD2.CONFIG,PRD1.BASE)  1
// LIBDEF PHASE,CATALOG=lib.sublib                     2
// OPTION CATAL
 INCLUDE ASMADOLK                                     3
// EXEC LNKEDT,PARM='MSHP'
/*
/&
```

Figure 26. Sample JCL to Linkedit the Default Options Phase

In area **1** the library-search chain points first to the sublibrary where ASMADOPT.OBJ was catalogued when you assembled the changed options. It then points to the sublibrary where High Level Assembler was installed, and where the link book ASMADOLK.OBJ resides.

In area **2** , change the library and sublibrary to reflect the user sublibrary where ASMADOPT.PHASE will reside.

In area **3** a link book supplies the phase statement and include values.

Placing High Level Assembler into the Shared Virtual Area (SVA)

This customization must be done by a system administrator.

Table 47 on page 110 gives a list of phase names, their link-edit attributes and approximate sizes. It shows which phases are eligible for sharing storage, and the approximate size of each phase.

To assist you in placing the High Level Assembler phases into the SVA, a sample member named `ASMASVA2.Z` is provided. This sample job includes all the High Level Assembler phases eligible for the SVA.

The *VSE/ESA System Control Statements* provides specific instructions about placing phases into the SVA.

The phases are loaded using a load list. The $SVAASMA load list is provided. This load list can also be placed in the BG startup procedure. If this is done then the library containing the High Level Assembler must be in the LIBDEF search chain within the LIBSDL procedure.

```
// JOB ASMASVA2 LOAD SVA-ELIGIBLE PHASES
*
*  Load SVA using a load list
*
// LIBDEF *,SEARCH=(lib.sublib,PRD1.BASE)
SET SDL
LIST=$SVAASMA
/*
/&
```

Figure 27. Sample JCL to Load SVA-eligible Phases using Load List

Chapter 15. Maintaining High Level Assembler on VSE

This chapter describes how to re-install or remove High Level Assembler and how to apply service updates to High Level Assembler. To effectively use the maintenance procedures, you must have already installed High Level Assembler and any required products.

In addition, this chapter describes how to remove High Level Assembler.

Re-installing High Level Assembler

You do not need to perform all the planning and installation procedures to re-install High Level Assembler. For example, you might not need to reconsider your storage needs if High Level Assembler replaces the existing High Level Assembler data sets.

You do not need to remove High Level Assembler from your system before re-installing High Level Assembler, unless you intend to re-install the product in a different sublibrary from the previous installation. In this case you must remove High Level Assembler from the system history file before you can re-install it.

To re-install High Level Assembler, you follow the same steps as for installing High Level Assembler. See Chapter 13, “Installing High Level Assembler on VSE” on page 111.

Applying Service Updates

You might need to apply maintenance or service updates to High Level Assembler periodically. This section details these procedures.

What You Receive

If you report a problem with High Level Assembler to your IBM Support Center, you will receive a tape containing one or more APARs or PTFs to solve your problem.

You might also receive a list of pre-requisite APARs or PTFs, which should have been applied to your system before applying the current service. These pre-requisite APARs or PTFs might relate to High Level Assembler or any other licensed product you have installed, including VSE/ESA.

You apply service to High Level Assembler using either the VSE/ESA Interactive Interface or a batch job.

The following checklist provides a summary of steps you should use to apply service to High Level Assembler.

Checklist for Applying Service

Table 49 lists the steps for installing corrective service on High Level Assembler. You can use Table 49 as a checklist.

Table 49. Summary of Steps for Installing Service on High Level Assembler

Step	Description	MSHP Command or Jobname	page
__ 1	Ensure prerequisite APARs or PTFs are applied.	RETRACE	128
__ 2	Backup existing system	___	128
__ 3	Apply service	INSTALL	129
__ 4	Run the installation verification program	ASMAIVPS	130

Step 1. Check Prerequisite APARs or PTFs

Prerequisite APARs or PTFs need to be applied to your system before you can apply the current maintenance. These APARs or PTFs might apply to High Level Assembler or any licensed program you have installed at your installation.

Your IBM Support Center has given you a list of any relevant prerequisite APARs or PTFs. Probably most have been applied to your system. You can verify this by retracing the APARs and PTFs in your system history file. The job shown in Figure 28 shows how to retrace APARs and PTFs in the system history file. This job is supplied as ASMARETR.Z

Use this listing to check that you have already applied any pre-requisite APARs or PTFs. If you have not, arrange for your IBM Support Center to send them to you and apply them before applying other service.

```
// JOB ASMARETR Retrace APARs and PTFs
// EXEC MSHP,SIZE=700K
RETRACE APARS
RETRACE PTFS
/*
/&
```

Figure 28. Job to Retrace APARs and PTFs

Step 2. Backup Original System

Make a backup copy of your current High Level Assembler library and the system history file. For information about backing up libraries and the system history file, see *VSE/ESA System Control Statements*.

Step 3. Apply Service

You can apply service to High Level Assembler from the provided service tape using either the Interactive Interface or a batch job.

Method 1: Apply Service Using the Interactive Interface

To apply service to High Level Assembler using the Interactive Interface, log on to the VSE/ESA Interactive Interface as the system administrator. (If you would like information about the functions of the Interactive Interface, refer to *VSE/ESA Administration*.)

Mount the service tape on an available tape drive.

In the following menus specify the highlighted items that appear after the ===> symbol.

1. **VSE/ESA FUNCTION SELECTION** menu:
   ===> ***1*** (Installation)

2. **INSTALLATION** menu:

   ===> ***4*** (IBM Service)

   a. **IBM SERVICE** menu:
      ===> ***2*** (PTF Handling)

   b. **PTF HANDLING** menu:

   - If you want to print the documentation about the supplied PTFs before applying the service, select:
     ===> ***1*** (Print Service Document)

     PRINT SERVICE DOCUMENT menu:
     ===> ***cuu***
     (the address of the tape drive where you mounted the service tape)

   - If you want to apply the service directly, select:
     ===> ***3*** (Apply PTFs from Service Tape)

     APPLY PTF menu:
     ===> ***cuu***
     (the address of the tape drive where you mounted the service tape). Press Enter.

     ===> ***tapename***
     (allocate a unique name for the service tape)
     Press Enter.

   c. **JOB DISPOSITION** menu:

   Make any changes required and press Enter to submit the job and apply the service.

   d. Respond to console messages

Confirm prompt when asking for service tape.

Method 2: Apply Service Using a Batch Job

The batch job to apply service to High Level Assembler uses the MSHP system history file where High Level Assembler was installed.

A sample job to apply service using MSHP is shown in Figure 29. For more information on MSHP see *VSE/ESA System Control Statements*.

```
// JOB ASMAAPP Apply Service
// ASSGN SYS006,cuu            1
// EXEC MSHP,SIZE=700K
INSTALL SERVICE FROMTAPE        2
/*
/&
```

Figure 29. Job to Retrace APARs and PTFs

In area **1**, change *cuu* to the address of the tape drive where you have mounted the service tape.

Area **2** shows the MSHP statement to install service from a tape. The information in the system history file directs MSHP to apply the service to the sublibrary in which High Level Assembler is installed. You do not need to supply this information. This job is supplied as member ASMAAPP.Z

Step 4. Run the installation verification program (IVP)

After you have applied all the files on the service tape, run the installation verification program ASMAIVPS to ensure that High Level Assembler functions properly. See the description of ASMAIVPS in Figure 24 on page 121.

Removing High Level Assembler

You do not have to remove High Level Assembler from your system before installing a new version or release.

If you do have to remove High Level Assembler for any reason, you must delete all the High Level Assembler entries from your sublibrary and remove High Level Assembler from the system history file.

A sample job ASMADELV.Z will

- delete all High Level Assembler members from the sublibrary where it is installed
- remove High Level Assembler from System History File
- update DTRIHIST.Z in the system library to remove High Level Assembler from the list of installed products

  Consult the instructions in the sample job ASMADELV.Z for more information.

To Report a Problem with High Level Assembler

To assist with reporting any difficulties refer to the diagnostic process as shown in Chapter 16, “Isolating the Problem” on page 135.

Report any difficulties you have using this product to your IBM Support Center. In the United States, if an APAR is required, the Support Center will provide the address to which any needed documentation can be sent.

Table 50 identifies the component ID (COMP ID) for High Level Assembler.

Table 50. Component IDs

COMP ID	Component Name	REL
569623400	VSE HIGH LEVEL ASM	489

Obtaining Service Information

Preventive Service Planning (PSP) information is continually updated as fixes are made available for problems. Check with your IBM Support Center or use either Information/Access or SoftwareXcel Extended to see whether there is additional PSP information that you need. To obtain this information, specify the following UPGRADE and SUBSET values: HLASM140 and HLASMVSE489.

Diagnosing Problems with High Level Assembler

Chapter 16. Isolating the Problem . . . 135
Diagnosing the Problem . . . 135
Diagnostic Aids . . . 136
Installation Problems . . . 137

Chapter 17. Building a Keyword String . . . 139
Keyword Usage . . . 139
Using the Problem Identification Worksheet . . . 140
Step 1. Component Identification Keyword Procedure . . . 141
Step 2. Release Level Keyword Procedure . . . 141
  Release Level Procedure . . . 141
Step 3. Type-of-Failure Keyword Procedure . . . 142
  Abnormal Termination Problem Procedures . . . 142
    ABENDxxxx Procedure . . . 142
  Message Problem Procedures . . . 144
    MSGx Procedure . . . 145
  Loop or No Response Problem Procedures . . . 145
  Output Problem Procedures . . . 146
    INCORROUT Procedure . . . 146
  Performance Problem Procedures . . . 147
    PERFM Procedure . . . 147
  Documentation Problem Procedures . . . 148
    DOC Procedure . . . 148
Step 4. System-Type Keyword . . . 149
  System-Type Procedure . . . 149

Chapter 18. Using the Keyword String as a Search Argument . . . 151
How to Use the Keyword String . . . 151
Search Argument Procedure . . . 151

Chapter 19. Preparing an APAR . . . 153
Initiating an APAR . . . 153
    Machine-Readable Source . . . 155
    Job Control Statements . . . 155
    Interactive Environment . . . 155
    Job Control Statements . . . 156
    SYSADATA (MVS and CMS) or SYSADAT (VSE) Output . . . 156
Submitting the APAR Documentation . . . 156

Chapter 20. Modifier Keywords . . . 159
High Level Assembler Language Elements . . . 160
Assembler Options . . . 162

Chapter 21. Problem Identification Worksheet . . . 165

Chapter 22. Internal Trace Facility . . . 167
How to Invoke the Internal Trace Facility . . . 167
Specifying the Trace Data set . . . 167
ITF Assembler Option . . . 168

Chapter 16. Isolating the Problem

This chapter covers the diagnostic process, diagnostic aids that are provided and overcoming installation problems.

If a problem occurs while you are using High Level Assembler, its cause might not be obvious. It might be an error in your program, in the assembler itself, or in some component of the assembler's operating environment. To help you identify the failure, use the following procedure.

Diagnosing the Problem

This procedure gathers the diagnostic information required for developing a keyword string to search the software support database. It describes options that supply all available diagnostic information. You will need this information to discuss the problem with your IBM support representative if a search against the database fails to locate a fix for your problem.

1. Determine if the program has been changed since it was last assembled successfully. If it has, examine the changes. If the error is occurring in the changed code and cannot be corrected, note the change that caused the error. If possible, retain copies of both the original and the changed programs to submit with an Authorized Program Analysis Report (APAR) if it is required.
2. Ensure that you are assembling the correct version of the source code. You might have incorrectly identified the location of your source file. For example, check your data set names.
3. Determine if the problem looks like a wait or a loop. If it does, it might be a system problem. You should follow your installation's procedures for resolving such problems.
4. Follow the basic diagnostic procedures discussed in *docid=asma100.HLASM Programmer's Guide*. If you receive an assembler error diagnostic message, you should verify the correct syntax and usage of the code that produced the error.
5. Correct all problems diagnosed by diagnostic messages, and make sure that previous diagnostic messages are not the cause of the current problem. Pay attention to warning messages (W-level messages). Identify the message by the following convention:
   - High Level Assembler messages are prefixed by the characters ASMA.
   - For messages with prefixes other than ASMA, the prefix determines which system/subsystem issued the message. Consult the applicable system/subsystem messages manual.
6. Your installation may have received an IBM Program Temporary Fix (PTF) for the problem. Make sure that all PTFs have been applied, so that your installation is at the latest maintenance level.
7. The Preventive Service Plan (PSP bucket), an online database available to IBM customers, gives information about product installation problems and other problems. Refer to the “Planning for Installation” chapters on page 11 for MVS, page 56 for VM, and page 105 for VSE for more information.

8. After the failure has been identified, consider writing a small test case that reproduces the problem. This test case should help you to:
   - Isolate the problem.
   - Distinguish between an error in the application program and an error in High Level Assembler.
   - Choose keywords that best describe the error.
9. Specify the following assembler options, in addition to the options originally specified, and reassemble the program. These options produce maximum diagnostic information that help you diagnose product errors. See *HLASM Programmer's Guide* for more information on how to use these options.

   ```
   DXREF
   ESD
   FLAG(0,ALIGN,CONT,RECORD,SUBSTR)
   LIST(121)
   MXREF(FULL)
   PCONTROL(DATA,GEN,ON,MCALL,MSOURCE,UHEAD)
   RLD
   RXREF
   USING(MAP,WARN(15))
   XREF(FULL)
   ```

10. If the error symptoms change, return to step 4 on page 135.
11. Record the sequence of events that led to the error condition. You might be able to use this information in developing a keyword string, and will need it if an APAR is required.
12. Begin developing the keyword string, using the procedure in Chapter 17, “Building a Keyword String” on page 139.

Diagnostic Aids

In certain situations when an assembly cannot be completed, High Level Assembler requests its internal abnormal termination routine to produce a specially formatted dump. Diagnostic information in the dump can be useful if an APAR is needed. The types of data that can be extracted from the dump include:

- Assembly abnormal-termination messages
- Register contents when the abnormal termination was requested
- The assembler common-storage area
- The statement that was being processed when the abnormal termination was requested

The contents of the dump depend upon when the abnormal termination was requested during processing of the assembly.

This information should be retained and provided when requested by IBM.

Installation Problems

You can avoid or solve most installation problems if you follow these steps:

1. Consult the PSP bucket (see step 7 on page 135).
2. Read any material that accompanies the installation tape.
3. Review the step-by-step installation procedure before installing High Level Assembler.

If you still cannot solve the problem, develop a keyword string based on the symptoms of the problem as described in Chapter 17, “Building a Keyword String” on page 139.

Chapter 17. Building a Keyword String

Failures in High Level Assembler can be described through the use of *keywords*. A keyword is a word or abbreviation assigned to describe one aspect of a product failure. A set of keywords, called a keyword string, can be used to describe the failure in detail. The procedures in this section will help you construct a keyword string that describes what you know about the product failure.

Information/Access, a feature of IBM Information/System Licensed Program, 5735-0ZS, can give you access to a computer-based abstract of the information in the software support database. This feature allows you to do your own search for previously recorded product failures before calling the IBM Support Center.

After it is constructed, the keyword string is used as a search argument to search against an IBM software support database, such as the Software Support Facility (SSF). The database contains keyword and text information describing all current problems, reported through APARs, and associated PTFs. IBM Support Center personnel have access to the software support database and are responsible for storing and retrieving the information. They use the keyword string to search the database and retrieve records that describe similar known problems.

If the keyword string produces a match in the software support database, the search might yield a fuller description of the problem and possibly identify a correction or circumvention. Such a search might yield several matches with previously reported problems. Review each error description carefully to determine if the problem description in the database matches your problem.

If a match is not found, use the keyword string you have constructed to describe the failure when contacting the IBM Support Center for assistance and when submitting an APAR. Keywords ensure that identical program errors are described with identical keyword strings. Spelling the keywords exactly as they are presented in this book is especially important for a successful match. For additional information on keywords and APAR preparation, see *Field Engineering Programming System General Information*.

Keyword Usage

The first keyword in a keyword string identifies the failing component. The component identification for High Level Assembler is the product identifier (569623400). A search of the software support database with this single keyword would locate all problems reported for the whole assembler. Each additional keyword added to the keyword string narrows the scope of the search argument and helps to eliminate unnecessary examination of problem descriptions that have similar, but not matching, characteristics. In some cases, a correction for a product failure might be located with less than a full set of keywords. If you cannot follow the instructions for selecting a particular keyword, omit that keyword to avoid incorrectly identifying the problem. In general, if you contact IBM, you will be asked to identify your problem with a full set of keywords, as described here.

Figure 30 on page 140 shows the process of creating a keyword string. The keywords are indicated as you proceed through the diagram. A full set of keywords for High Level Assembler contains:

- The component identification
- The release level
- The type of failure
- One or more modifier keywords, depending on the type of failure, if applicable

Follow the steps in the keyword procedures until you are directed to the search argument procedure.

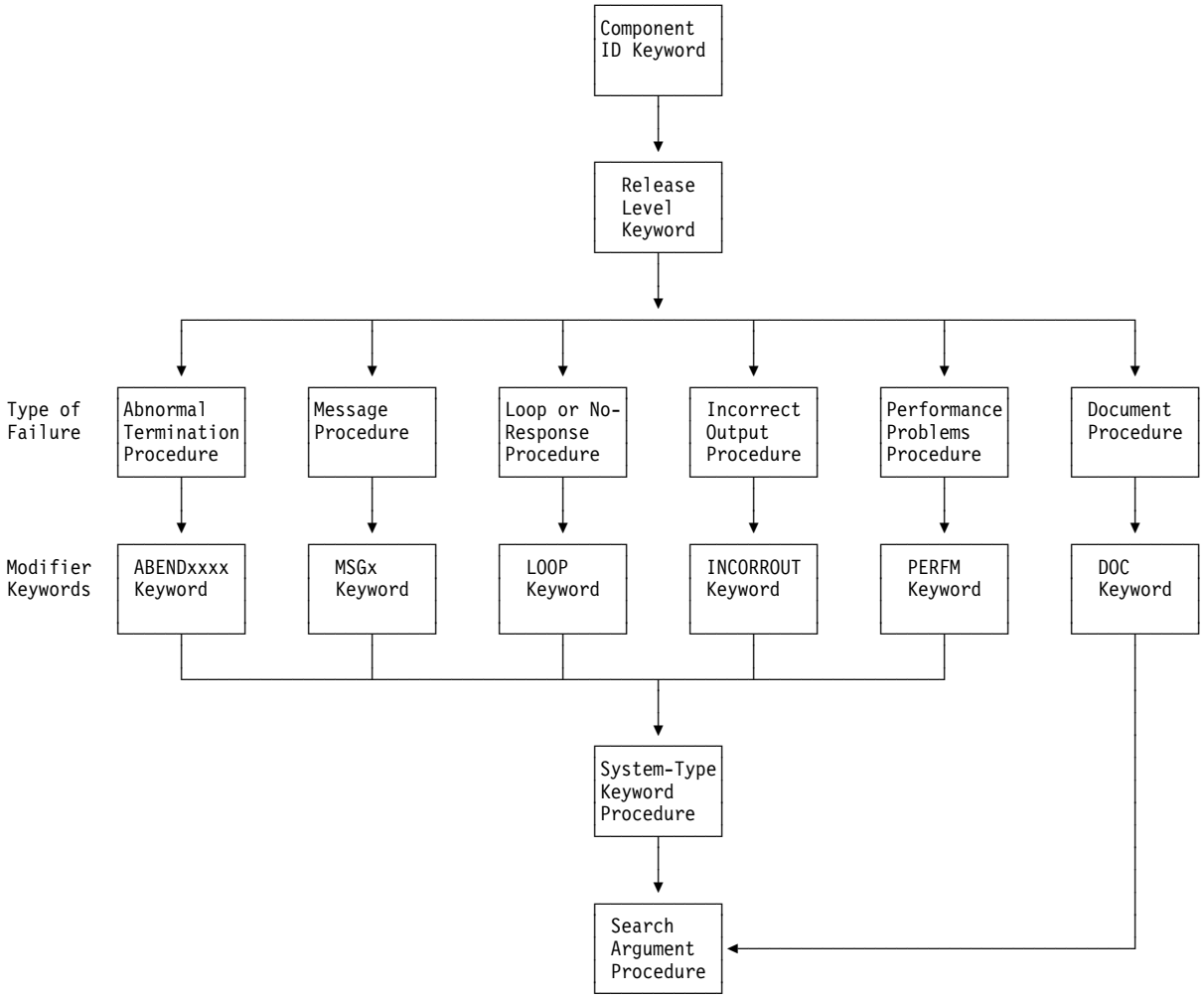


Figure 30. High Level Assembler–Problem Identification Using Keywords

Using the Problem Identification Worksheet

You can use the worksheet described in Chapter 21, “Problem Identification Worksheet” on page 165 to help you construct and record a keyword string. As you identify the keywords associated with your software problem, record them in the spaces provided.

Step 1. Component Identification Keyword Procedure

This procedure shows what to specify in the component identification keyword. The component identification keyword is always the first keyword placed in the search argument string. It is derived from the High Level Assembler program number and identifies the area within the software support database that contains APARs for High Level Assembler.

Use the component identification keyword with at least a type-of-failure keyword to search the software support database. If you use the component identifier keyword without additional keywords, a full listing of all APARs affecting High Level Assembler are produced.

Component Identification Procedure

1. Use `569623400` as the component identification keyword. This number is the IBM High Level Assembler for MVS & VM & VSE product identifier, 5696234, with a two-digit component number appended. The component number in this case is `00`.
2. If service updates have been applied to the licensed program, the update level of the last service update applied should be noted. See your system programmer for the current service level of your High Level Assembler. Although the service update level is not used as a keyword search argument, it is useful when reviewing APARs selected during the keyword search. The *Diagnostic Cross Reference and Assembler Summary* section of the assembler listing shows the last PTF applied to the High Level Assembler.

Step 2. Release Level Keyword Procedure

Use the following procedure to identify the specific release level of High Level Assembler under which you were operating when the failure occurred.

Release Level Procedure

1. Locate the release and modification level (denoted by `HLASM R`*`r.m`*) line at the top of the first page of your latest assembly output listing for the failing program. The release level line contains the current product identification data in the following format:

   ```
   HLASM Rr.m  yyyy/mm/dd hh.mm
   ```

   where *r.m* specifies the current release and modification number. The release level line for Release 4.0 of High Level Assembler is:

   ```
   HLASM R4.0  yyyy/mm/dd hh.mm
   ```

2. Specify the release level keyword, using the following format: *Rrm0*

   where *r* is the release level, and *m* is the modification level. The last character of this keyword is always zero.

   If Release 4.0 is the level you found in your listing, the release level keyword is R400.

The following is an example of a set of keywords, consisting of the component identification and release level keywords:

```
Component Identification:     569623400

Release Level:                R400
```

Step 3. Type-of-Failure Keyword Procedure

Various types of failures might occur in the High Level Assembler licensed program. Read the following table and select the type of failure that best describes the problem. Then go to the associated keyword procedure listed in this table for instructions on how to complete the keywords for that type of failure. If more than one keyword describes the problem you are experiencing, use the one that appears first in the table.

Table 51. Types of High Level Assembler Failures

Type-of-Failure	Symptom	Procedure
Abnormal Termination	The assembler has ended abnormally without a message, or with a completion code that indicates a system abend has occurred.	See "Abnormal Termination Problem Procedures."
Message Problems	A message indicates an assembler error, or there seems to be an error with the message itself.	See "Message Problem Procedures" on page 144.
No Response Problems	The assembler seems not to be doing anything, or is doing something repetitively.	See "Loop or No Response Problem Procedures" on page 145.
Output Problems	The output from the assembler is missing or incorrect.	See "Output Problem Procedures" on page 146.
Performance Problems	The performance of the assembler is degraded.	See "Performance Problem Procedures" on page 147.
Documentation Problems	Information in one of the High Level Assembler publications is incorrect or missing.	See "Documentation Problem Procedures" on page 148.

Abnormal Termination Problem Procedures

Use this procedure if High Level Assembler ends abnormally with a system message or a system abend code. Do not use this procedure if termination was accompanied by a message with the prefix ASMA. For those situations, see "Message Problem Procedures" on page 144. On MVS, if High Level Assembler ends abnormally with a system abend code 322, 522, or 722, you should use the procedure for "Loop or No Response Problem Procedures" on page 145.

ABENDxxxx Procedure

1. Determine with which assembler options the failure occurs. If the failure occurs only when using certain options, indicate those options in the keyword string. Select the applicable modifier keyword from the list shown in "Assembler Options" on page 162.

2. On MVS, if a system abend code is available, replace the *xxxx* of ABENDxxxx with the system abend code, prefixed by a zero. For example, if the code was a system abend 0C4, specify `ABEND00C4` as your keyword. If the failure occurred with a system abend 0C1, your set of keywords, so far, consists of:

   ```
   Component Identification:     569623400

   Release Level:                R400

   Type of Failure:              ABEND00C1
   ```

3. If you receive the following message on CMS, use `OPERATION` and `EXCEPTION` as keywords.

   ```
   DMSITP141T Operation exception occurred at 6BB40C in routine ASMAHL
   ```

   Your set of keywords would then consist of:

   ```
   Component Identification:     569623400

   Release Level:                R400

   Type of Failure:              OPERATION EXCEPTION
   ```

4. On VSE, ABENDs will be accompanied by a message which describes the cause of the error. Use words from the message as keywords. For example, if you receive the following message, you could use `PROTECTION` and `EXCEPTION` as keywords.

   ```
   0S03I PROGRAM CHECK INTERRUPTION - HEX LOCATION 0006D2D0 - CONDITION CODE 0
   - PROTECTION EXCEPTION
   ```

   Your set of keywords would then consist of:

   ```
   Component Identification:     569623400

   Release Level:                R400

   Type of Failure:              PROTECTION EXCEPTION
   ```

5. On CMS or VSE, if using words from the message as your type-of-failure keyword does not produce a match in the software support database, use the equivalent MVS system abend code as the type-of-failure keyword.

   The most common errors, with their MVS system abend codes, are:

Error	**System Abend Code**
Operation exception	0C1
Privileged operation exception	0C2
Execute exception	0C3
Protection exception	0C4
Addressing exception	0C5
Specification exception	0C6
Data exception	0C7
Fixed-point overflow exception	0C8
Fixed-point divide exception	0C9

6. Determine with which assembler options the failure occurs. If the failure occurs only when using certain options, indicate those options in the keyword string. Select the applicable modifier keyword from the list shown in “Assembler Options” on page 162.

Message Problem Procedures

The following message types are issued by High Level Assembler:

- *ASMAHL* command errors (CMS)
- Assembler-error diagnostic messages
- Assembly abnormal-termination messages

High Level Assembler messages are identified by the prefix ASMA.

On CMS, messages issued by the *ASMAHL* command have a prefix of ASMACMS.

The format of the message identifier is ASMA*nnnc*, where:

ASMA is the message prefix identifying all High Level Assembler messages.

nnn is the message number.

c is one of the following message severity characters:

- I for informational messages
- W for warning messages
- E for normal error messages
- S for severe error condition messages
- C for critical error condition messages
- U for unrecoverable error condition messages

Assembler error diagnostic messages are numbered `ASMA001` to `ASMA899`. Assembly abnormal termination messages are numbered `ASMA900` to `ASMA999`. Messages with other prefixes are issued by operating systems, subsystems, and access methods. They should not be addressed as High Level Assembler product problems. See the messages manuals for the applicable components.

Use the MSGx keyword procedure for any one of the following conditions:

- A message is issued under a set of conditions that should not have caused it to be issued.
- A message contains incorrect data or is missing data.
- A message indicates an internal assembler error (for example, `ASMA951`).

Do not use this procedure if the assembler ended with a system abend code, or a system message indicating an abnormal-termination. In these cases, continue with “Abnormal Termination Problem Procedures” on page 142.

MSGx Procedure

1. Replace the *x* of MSGx with the complete message identifier, but do not include the severity character (if any). For example, if assembly abnormal termination message `ASMA950U` is issued, the MSGx keyword would be `MSGASMA950`. Your set of keywords, so far, would consist of:

   ```
   Component Identification:     569623400

   Release Level:                R400

   Type of Failure:              MSGASMA950
   ```

2. Determine with which assembler options the failure occurs. If the failure occurs only when using certain options, indicate those options in the keyword string. Select the applicable modifier keyword from the list shown in “Assembler Options” on page 162.
3. Continue with “Step 4. System-Type Keyword” on page 149.

Loop or No Response Problem Procedures

Use the LOOP keyword procedure for any of the following conditions:

- A program seems to be doing nothing or is doing something repetitively.
- A job does not reach completion.
- On MVS, the system abend code is 322, 522, or 722, which means that your program has timed out or exceeded its output limits.

If the program appears to be in a WAIT state, follow your local procedures for resolution.

LOOP Procedure

1. Determine with which assembler options the failure occurs. If the failure occurs only when using certain options, indicate those options in the keyword string. Select the applicable modifier keyword from the list shown in “Assembler Options” on page 162.
2. If you are running on MVS and the error is a system abend, with a system abend code indicating not enough time, or inadequate output limits, increase the allotment and rerun your program. If the problem is still unresolved, your set of keywords, so far, would consist of:

   ```
   Component Identification:     569623400

   Release Level:                R400

   Type of Failure:              LOOP
   ```

3. Continue with “Step 4. System-Type Keyword” on page 149.

Output Problem Procedures

Use this procedure when the output appears to be incorrect or missing, but the program otherwise ended normally.

INCORROUT Procedure

1. If the data or records were repeated endlessly, use the “LOOP Procedure” on page 145 instead of the INCORROUT procedure to create your keyword string.
2. Use INCORROUT as your type-of-failure keyword.
3. If the error was detected because of incorrect or missing output from an assembly that otherwise completed successfully, select a modifier keyword from the following table to describe the type of error in the output.

Modifier Keyword	Type of Incorrect Output
DUPLICATE	Some data or records were duplicated, but were not repeated endlessly.
INVALID	The output that appeared was incorrect or not as expected.
MISSING	Some expected output was missing.

4. Select another modifier keyword from the following table to describe the portion of the output in which the error occurred.

Modifier Keyword	Portion of Output in Error
ADATA	Associated data file.
DXREF	DSECT cross reference.
ESD	External symbol dictionary listing. If the external symbol dictionary part of the object program is in error, use the OBJECT keyword followed by the ESD keyword.
GOFF (MVS and CMS)	Machine-language generalized object program.
MESSAGE	Diagnostic message.
MXREF	Macro and copy code source summary, and macro and copy code cross reference.
OBJECT	Machine-language object program.
RLD	Relocation Dictionary listing. If the relocation dictionary part of the object program is in error, use the OBJECT keyword followed by the RLD keyword.
RXREF	General Purpose Register Cross Reference
SOURCE	Source listing.
STAT	Statistics and error listing.
TERM	Progress and diagnostic messages on SYSTERM data set for MVS, on terminal for CMS or on SYSLOG for VSE.
UMAP	USING map.
XREF	Ordinary Symbol and Literal Cross Reference listing.

5. Determine with which assembler options the failure occurs. If the failure occurs only when using certain options, indicate those options in the keyword string. Select the applicable modifier keyword from the list shown in “Assembler Options” on page 162.

   For example, if you think that the assembler has given an incorrect Ordinary Symbol and Literal Cross Reference section of the listing when assembling with the XREF(FULL) option, your set of keywords, so far, would consist of:

   ```
   Component Identification:     569623400

   Release Level:                R400

   Type of Failure:              INCORROUT

   Modifiers:                    INVALID
                                 XREF
                                 FULL
   ```

6. Continue with “Step 4. System-Type Keyword” on page 149.

Performance Problem Procedures

Most performance problems can be related to system tuning and should be handled by system engineers and system programmers. Use the PERFM keyword when the performance problem cannot be corrected by system tuning and performance is below expectations as documented in an IBM product publication.

PERFM Procedure

1. Record the actual and expected performance measurements for your system configuration. Note the order number and page of the IBM document that is the source of your performance expectations. You will be asked for this information if you contact the IBM Support Center. If you prepare materials for an APAR, you should also include this information in the error description.
2. Determine with which assembler options the failure occurs. If the failure occurs only when using certain options, indicate those options in the keyword string. Select the applicable modifier keyword from the list shown in “Assembler Options” on page 162.
3. Use PERFM as your type-of-failure keyword. For example, your set of keywords for performance problems would consist of:

   ```
   Component Identification:     569623400

   Release Level:                R400

   Type of Failure:              PERFM
   ```

4. Continue with “Step 4. System-Type Keyword” on page 149.

Documentation Problem Procedures

Use the DOC keyword procedure when you notice a problem caused by incorrect or missing information in one of the published High Level Assembler documents.

DOC Procedure

1. Locate the page or pages in the document where the problem occurs, and prepare a description of the error and the problem it caused. This information is required for APAR preparation if no similar problem is found in the software support database.
2. Decide whether this documentation problem is severe enough to cause lost time for other users.

   If the problem is not severe, fill out the Reader's Comment Form attached to the back of the publication in question. Include the problem description you have developed, along with your name and return address, so that IBM can respond to your comments.

   If the problem is severe enough to cause lost time for other users, continue creating your keyword string to determine whether IBM has a record of the problem. If this is a new problem, you will be asked to submit a severity-3 or severity-4 documentation (DOC) APAR.
3. Use the order number on the cover of the document together with the DOC keyword as your type-of-failure keyword, but omit the hyphens. Leave a single space between DOC and the document number. The number following the last hyphen in the document number indicates the document release level. If the document release level number has only one digit, it must be preceded by a zero. For example if the order number is SC26-4940-03 (*HLASM Language Reference*), use SC26-494003. Your set of keywords consists of:

   ```
   Component Identification:     569623400

   Release Level:                R400

   Type of Failure:              DOC SC26-494003
   ```
4. To determine if this documentation problem has already been reported, turn to Chapter 18, “Using the Keyword String as a Search Argument” on page 151.

   If, after searching the IBM software support database, you do not find a matching description, **return here to continue.**
5. Before discontinuing your search, you may want to search again, using the following format:

   ```
   Component Identification:     569623400

   Release Level:                R400

   Type of Failure:              DOC SC26-4940**
   ```

   The two asterisks appended to the document number cause a search for all problems reported for the document rather than only those for a specific release of the document.
6. Continue with “Step 4. System-Type Keyword” on page 149.

Step 4. System-Type Keyword

Use this procedure to indicate which system you were operating on when High Level Assembler failed.

System-Type Procedure

1. If the failure occurred while your program was assembling:
   - On MVS, use MVS as your system-type keyword.
   - On CMS, use CMS as your system-type keyword.
   - On VSE, use VSE as your system-type keyword.
2. Use ESA as your next system-type keyword.

   For example, the keywords you use when a message problem occurs, would consist of:

```
Component Identification:     569623400

Release Level:                R400

Type of Failure:              MSGASMAxxx

System Type:                  MVS ESA
```

Chapter 18. Using the Keyword String as a Search Argument

This chapter explains how to use the keyword string you have developed to search the software support database. You can conduct the search yourself if you have access to the correct database, or you can request that IBM conduct the search.

How to Use the Keyword String

Searches against a software support database will be most successful if you follow these rules:

- Use only the keywords given in this book.
- Spell keywords the way they are spelled in this book. Any variation in spelling may result in an unsuccessful search.
- Include all the applicable keywords in any discussion with IBM support personnel or in an APAR.

Search Argument Procedure

1. Search the software support database, using the full set of keywords you have developed. For example given the following list:

   ```
   Component Identification:    569623400

   Release Level:               R400

   Type of Failure:             ABEND00C4

   System Type:                 MVS ESA

   Modifiers:                   MXREF
   ```

   your keyword string would consist of:

   ```
   569623400 R400 ABEND00C4 MVS ESA MXREF
   ```

2. If the search produces a list of APARs, continue with step 3, otherwise go to step 6 on page 152.
3. When your search is complete, eliminate from the list of possible APAR fixes those that have already been applied to your system.
4. Compare each of the remaining closed APAR descriptions with the current failure symptoms.
5. If a match is found, find out if there is a corresponding PTF. You can order the PTF from the IBM Support Center. You might already have the PTF at your installation, and only need to install it from the correct Program Update Tape (PUT) or Extended Service Option (ESO). If there is a PTF, apply it to your system and exit this procedure.

   For information about how to apply a PTF, refer to the cover letter for the PTF to be applied.

6. If the search **did not** produce a list of APARs, or an APAR description matching the current failure is not found, expand the search by using the following techniques:
   a. Omit the release level keyword (for example, `R400`) from the search. This expands the search to include similar failures on other release levels.
   b. Drop one keyword from the right end of the search argument string. The diagnostic procedures directed you to construct the keyword string with the most significant keywords listed first. By dropping a keyword from the right, you eliminate the least significant keyword, thereby expanding your search while maintaining the relevancy of your search argument string. Perform the search against the software support data base using your shortened search argument string. Repeat this step as necessary.
7. If a match is not found using the preceding techniques, go to Chapter 19, “Preparing an APAR” on page 153.

Chapter 19. Preparing an APAR

This chapter explains how to prepare an Authorized Program Analysis Report (APAR) if you are asked to do so by IBM support personnel.

You may be asked to prepare an APAR if:

- You have eliminated user errors as a possible cause of the problem.
- You have followed the diagnostic procedures presented in this book.
- The keyword search has proved unsuccessful.

Initiating an APAR

1. Contact the IBM Support Center for assistance. Tell the support personnel that you have used this manual to create a keyword string. Be prepared to supply the following information:
   - Customer number
   - Operating system
   - Operating system release level
   - Current High Level Assembler maintenance level (PTF list and list of APAR fixes applied)
   - The various keyword strings used to search the software support data base
   - Processor serial and model number
2. From the following list, you might be asked to include the applicable High Level Assembler environmental information with your APAR:
   - Job control statements
   - Any spooled CMS consoles or special EXECs
   - The following assembler listings:
     - High Level Assembler Options Summary
     - External Symbol Dictionary
     - Source and Object
     - Relocation Dictionary
     - Ordinary Symbol and Literal Cross Reference
     - Unreferenced Symbols in a CSECT
     - DSECT Cross Reference
     - Macro and Copy Code Source Summary
     - Macro and Copy Code Cross Reference
     - General Purpose Register Cross Reference
     - Diagnostic Cross Reference and Assembler Summary
     - USING map
   - Machine-readable copy of the program causing the problem, including all macros and copy members required by the program
   - Hard copy of job control language or CMS procedure for unloading the submitted machine-readable information
   - SYSADATA output

In addition, the console log might be helpful in reproducing the error. Any listings supplied must be from the High Level Assembler assembly that failed.

You may also be asked to provide trace information from the assembly that failed.

Table 52 provides the options or methods used to produce the documentation required. Many of these materials may have been previously produced in the required format during the development of the keyword string. (See "Diagnosing the Problem" on page 135.) Any additional requirements are explained after the table.

Table 52 (Page 1 of 2). Problem Resolution Documentation Descriptions

Item	Materials Required	How to Obtain Materials
1	Machine-readable source program	See "Machine-Readable Source" on page 155.
2	Assembly listings:	
	High Level Assembler Options Summary	by default
	External Symbol Dictionary	ESD option
	Source and Object	LIST option
	Relocation Dictionary	RLD option
	Ordinary Symbol and Literal Cross Reference	XREF(SHORT) or XREF(FULL)
	Unreferenced Symbols Defined in a CSECT	XREF(UNREFS)
	DSECT Cross Reference	DXREF option
	Macro and Copy Code Source Summary	MXREF(FULL) option or MXREF(SOURCE)
	Macro and Copy Code Cross Reference	MXREF(FULL) option or MXREF(XREF)
	General Purpose Register Cross Reference	RXREF option
	USING map listing	USING(MAP) option
	Diagnostic Cross Reference and Assembler Summary	by default
	Assembler abnormal termination dump	by default

Table 52 (Page 2 of 2). Problem Resolution Documentation Descriptions

Item	Materials Required	How to Obtain Materials
3	Assembler system ABEND dump	On MVS: SYSUDUMP DD statement (as directed by IBM support personnel). On CMS: VM DUMP command (as directed by IBM support personnel). On VSE JCL OPTION DUMP or PARTDUMP (as directed by IBM support personnel).
4	Partition/region size/virtual storage size	JCL or system programmer.
5	List of applied PTFs	System programmer.
6	MVS job control statements with MGGLEVEL(1,1), TSO ALLOCATE statements, CMS interactive session listing, or VSE job control statements with OPTION LOG.	See "Job Control Statements" on page 155, "Interactive Environment" on page 155 or "Submitting the APAR Documentation" on page 156.
7	SYSADATA (MVS or CMS) or SYSADAT (VSE) output	See "SYSADATA (MVS and CMS) or SYSADAT (VSE) Output" on page 156.
8	Trace Output (ON REQUEST)	See Chapter 22, "Internal Trace Facility" on page 167.

Machine-Readable Source

- The source program must be supplied in machine-readable form, using an IBM-supplied system utility program. The source program should be reduced to the smallest, least complex form that still produces the error.
- If the COPY statement was used, or library macros were called, the libraries used should be supplied in machine-readable form, preferably on unlabeled or standard labeled magnetic tape.

MVS

Job Control Statements

- Supply the JCL listings used to run the assembly, including an expanded list of the cataloged procedures used.
- If there is a large amount of JCL, it must be supplied in machine-readable form, preferably on unlabeled magnetic tape. MVS

CMS

Interactive Environment

- The listing supplied must include all parts of the interactive session that dealt with this problem.
- Supply full details of the interactive environment immediately before you invoked the assembler.

Use the CP SPOOL command to spool your console for printing, then issue the following commands:

```
QUERY DISK *
QUERY FILEDEF
QUERY INPUT
QUERY LIBRARY
QUERY MACLIB
QUERY OUTPUT
QUERY SEARCH
QUERY SET
QUERY TERMINAL
QUERY VIRTUAL
```

This provides the necessary details of your interactive environment. CMS

VSE

Job Control Statements

- Supply the JCL listings used to run the assembly, including an expanded list of the cataloged procedures used.
- Run LSERV to list the system standard labels and partition standard labels in effect.
- If there is a large amount of JCL, it must be supplied in machine-readable form, preferably on unlabeled magnetic tape.
- Issue a LISTIO for the partition to list the system logical unit assignments active.
- Issue a MAP command on the VSE console to show the partition size and GETVIS storage allocation.
- Run LIBR with the command LISTDIR SDL to list the contents of the system directory list (phases that have been placed in the SVA). VSE

SYSADATA (MVS and CMS) or SYSADAT (VSE) Output

SYSADATA output data must be supplied in machine-readable form, preferably on unlabeled magnetic tape, using an IBM-supplied system utility program or command.

Submitting the APAR Documentation

When submitting material for an APAR to IBM, be sure that the media containing source programs, job stream data, interactive environment information, data sets, or libraries are carefully packed and clearly identified.

Each tape submitted must have the following information attached and visible:

1. The APAR number assigned by IBM
2. A list of data sets on the tape (source program, JCL or interactive environment information, data, etc.)

3. A description of how the tape was made, giving the following information:
   a. A full listing of JCL or interactive environment information used to produce the machine-readable source. Include the block size, LRECL, and format of each file.

      MVS If the file was unloaded from a partitioned data set, include the block size, LRECL, and number of directory blocks in the original data set. MVS

   b. Labeling information used for the volume and its data sets.
   c. The recording mode and density.
   d. The name of the utility program that created each data set.
   e. The record format and block size used for each data set.

Chapter 20. Modifier Keywords

One or more modifier keywords may be used in the same keyword string to define the problem. Additional modifiers help make the search argument more specific. Use the capitalized spelling of the modifier in the keyword string. The various types of modifier keywords listed below are:

- Assembler Language
  - Ordinary assembler instructions
  - Conditional assembly instructions
  - Macro processing instructions
  - System variable symbols
  - Machine instructions

  See “High Level Assembler Language Elements” on page 160 for a list of the assembler language elements and the associated keywords to be used in the keyword string.

- Assembler Options

  Select from your assembly listing those assembler options that you consider significant to the type of failure. See “Assembler Options” on page 162 for a list of the assembler options. The option name itself (and suboption if applicable) is the keyword.

High Level Assembler Language Elements

LANGUAGE ELEMENT	KEYWORD
ACONTROL	ACONTROL
ACTR	ACTR
ADATA	ADATA
AGO	AGO
AEJECT	AEJECT
AIF	AIF
AINSERT	AINSERT
ALIAS	ALIAS
AMODE	AMODE
ANOP	ANOP
AREAD	AREAD
ASPACE	ASPACE
Built-in Function[1]	
CCW	CCW
CCW0	CCW0
CCW1	CCW1
CEJECT	CEJECT
CATTR	CATTR(MVS and CMS)
CNOP	CNOP
COM	COM
COPY	COPY
CSECT	CSECT
CXD	CXD
DC	DC
DROP	DROP
DS	DS
DSECT	DSECT
DXD	DXD
EJECT	EJECT
END	END
ENTRY	ENTRY
EQU	EQU
EXITCTL	EXITCTL
EXTRN	EXTRN
GBLA	GBLA
GBLB	GBLB
GBLC	GBLC
ICTL	ICTL
LCLA	LCLA
LCLB	LCLB
LCLC	LCLC
LOCTR	LOCTR
LTORG	LTORG
Machine instruction[2]	
MEXIT	MEXIT
MHELP	MHELP
MNOTE	MNOTE
OPSYN	OPSYN
ORG	ORG
POP	POP
PRINT	PRINT
PUSH	PUSH
REPRO	REPRO
RMODE	RMODE
RSECT	RSECT
SETA	SETA
SETAF	SETAF
SETB	SETB
SETC	SETC
SETCF	SETCF
SPACE	SPACE
START	START
&SYSADATA_DSN	SYSADATA DSN
&SYSADATA_MEMBER	SYSADATA MEMBER
&SYSADATA_VOLUME	SYSADATA VOLUME
&SYSASM	SYSASM
&SYSCLOCK	SYSCLOCK
&SYSDATC	SYSDATC
&SYSDATE	SYSDATE
&SYSECT	SYSECT
&SYSIN_DSN	SYSIN DSN
&SYSIN_MEMBER	SYSIN MEMBER
&SYSIN_VOLUME	SYSIN VOLUME
&SYSJOB	SYSJOB
&SYSLIB_DSN	SYSLIB DSN
&SYSLIB_MEMBER	SYSLIB MEMBER
&SYSLIB_VOLUME	SYSLIB VOLUME
&SYSLIN_DSN	SYSLIN DSN
&SYSLIN_MEMBER	SYSLIN MEMBER
&SYSLIN_VOLUME	SYSLIN VOLUME
&SYSLIST	SYSLIST
&SYSLOC	SYSLOC
&SYSMAC	SYSMAC
&SYSM_HSEV	SYSM HSEV
&SYSM_SEV	SYSM SEV
&SYSNEST	SYSNEST
&SYSNDX	SYSNDX
&SYSOPT_DBCS	SYSOPT DBCS
&SYSOPT_OPTABLE	SYSOPT OPTABLE
&SYSOPT_RENT	SYSOPT RENT
&SYSOPT_XOBJECT	SYSOPT XOBJECT
&SYSPARM	SYSPARM
&SYSPRINT_DSN	SYSPRINT DSN
&SYSPRINT_MEMBER	SYSPRINT MEMBER
&SYSPRINT_VOLUME	SYSPRINT VOLUME
&SYSPUNCH_DSN	SYSPUNCH DSN
&SYSPUNCH_MEMBER	SYSPUNCH MEMBER
&SYSPUNCH_VOLUME	SYSPUNCH VOLUME
&SYSSEQF	SYSSEQF
&SYSSTEP	SYSSTEP

[1] For a conditional assembly language built-in function you should use the two keywords “BIF” and the built-in function name.

[2] For a machine instruction language element, you should use the machine instruction mnemonic as the keyword.

LANGUAGE ELEMENT	KEYWORD
&SYSSTMT	SYSSTMT
&SYSTEM_ID	SYSTEM ID
&SYSTERM_DSN	SYSTERM DSN
&SYSTERM_MEMBER	SYSTERM MEMBER
&SYSTERM_VOLUME	SYSTERM VOLUME
&SYSTIME	SYSTIME
&SYSVER	SYSVER
TITLE	TITLE
USING	USING
WXTRN	WXTRN
XATTR	XATTR(MVS and CMS)

Assembler Options

The option name (and suboption if applicable) is the keyword.

ASSEMBLER OPTION

ABOVE
ADATA
ALIGN
ASA
BATCH
BELOW
| CODEPAGE
COMPAT
COMPAT CASE
COMPAT LITTYPE
COMPAT MACROCASE
COMPAT SYSLIST
COMPAT NOCASE
COMPAT NOLITTYPE
COMPAT NOMACROCASE
COMPAT NOSYSLIST
DBCS
DECK
DELETE
DSECT
DISK (CMS)
DXREF
ERASE (CMS)
EXIT
EXIT ADEXIT
EXIT INEXIT
EXIT TRMEXIT
EXIT LIBEXIT
EXIT PRTEXIT
EXIT OBJEXIT
EXIT NOADEXIT
EXIT NOINEXIT
EXIT NOTRMEXIT
EXIT NOLIBEXIT
EXIT NOPRTEXIT
EXIT NOOBJEXIT
ESD
FLAG n
FLAG ALIGN
FLAG CONT
FLAG IMPLEN
FLAG PAGE0
| FLAG PUSH
FLAG RECORD
FLAG SUBSTR
| FLAG USING0
FLAG NOALIGN
FLAG NOCONT
FLAG NOIMPLEN
FLAG NOPAGE0
| FLAG NOPUSH
FLAG NORECORD
FLAG NOSUBSTR
| FLAG NOUSING0

ASSEMBLER OPTION

FOLD
GOFF (MVS and CMS)
GOFF ADATA (MVS and CMS)
GOFF NOADATA (MVS and CMS)
INFO
LANGUAGE
LANGUAGE DE
LANGUAGE EN
LANGUAGE ES
LANGUAGE EU
LANGUAGE JP
LIBMAC
LINECOUNT
LINECOUN
LIST 121
LIST 133
LIST MAX
LIST YES
MXREF
MXREF FULL
MXREF SOURCE
MXREF XREF
NOADATA
NOALIGN
NOASA
NOBATCH
NOCOMPAT
NODBCS
NODECK
NODXREF
NOERASE (CMS)
NOEXIT
NOESD
NOFOLD
NOGOFF (MVS and CMS)
NOINFO
NOLIBMAC
NOLIST
NOMXREF
NOOBJECT
NOPCONTROL
NOPRINT (CMS)
NOPROFILE
NORA2
NORENT
NORLD
NORXREF
NOSEG (CMS)
NOTERM
NOTEST
| NOTHREAD
NOTRANSLATE
NOUSING
NOXOBJECT (MVS and CMS)

ASSEMBLER OPTION

NOXREF
OBJECT
OPTABLE DOS
OPTABLE ESA
OPTABLE UNI
OPTABLE XA
OPTABLE 370
PCONTROL
PCONTROL ON
PCONTROL OFF
PCONTROL GEN
PCONTROL DATA
PCONTROL UHEAD
PCONTROL MCALL
PCONTROL MSOURCE
PCONTROL NOGEN
PCONTROL NODATA
PCONTROL NOUHEAD
PCONTROL NOMCALL
PCONTROL NOMSOURCE
PESTOP
PRINT (CMS)
PROFILE (MVS and CMS)
PROFMEM (MVS and CMS)
RA2
RENT
RLD
RXREF
SEG (CMS)
SIZE
STORAGE
SYSPARM
SYSPARMV
TERM
TERM WIDE
TERM NARROW
TEST
| THREAD
TRANSLATE
USING
USING LIMIT
USING NOLIMIT
USING MAP
USING NOMAP
USING WARN
USING NOWARN
XOBJECT (MVS and CMS)
XOBJECT ADATA (MVS and CMS)
XOBJECT NOADATA (MVS and CMS)
XREF
XREF SHORT
XREF FULL
XREF UNREFS

Chapter 21. Problem Identification Worksheet

Component Identification: _____

Release Level: _____

Type of Failure: _____

System Type: _____

Modifiers: _____

_____

_____

_____

Some keywords may not be applicable to all software problems.

Chapter 22. Internal Trace Facility

The Internal Trace Facility provides the IBM support personnel with information to assist with debugging errors in High Level Assembler.

You might be asked to provide an internal trace, or assist IBM support personnel in obtaining an internal trace, if the information described in Table 52 on page 154 is not sufficient to resolve the reported problem.

How to Invoke the Internal Trace Facility

To invoke the Internal Trace Facility the following procedures are required:

1. IBM provides a trace control module which must be included in a load library (on MVS and CMS), in the standard load module search order, or in a library in the standard phase search order (on VSE), that is available when the assembly is run. Your IBM support representative will provide you with the necessary instructions to perform this procedure.
2. Run the assembly with the ITF assembler option. The ITF option syntax is described below under ITF Assembler Option.
3. Supply the output from the internal trace in machine-readable form, preferably on unlabeled magnetic tape or cartridge, using an IBM-supplied system utility program or command. Your IBM support representative will advise you of the best method for providing this material.

Specifying the Trace Data set

MVS Use the SYSTRACE DD statement to define the Internal Trace Facility output data set:

```
//SYSTRACE DD DSN=dsname,DISP=(NEW,CATLG),SPACE=(CYL,(primary,secondary)),
//          DCB=(LRECL=81,BLKSIZE=n*81,RECFM=F)
```

CMS If you do not issue the SYSTRACE FILEDEF command before the High Level Assembler command, the High Level Assembler command issues the following *FILEDEF* command:

```
FILEDEF SYSTRACE DISK fn SYSTRACE m1 (RECFM F LRECL 81
```

VSE The data set that receives the output from the Internal Trace Facility is defined using the following JCL:

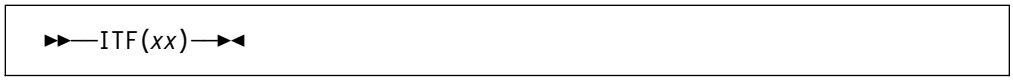
```
// ASSGN SYSTRAC,DISK,VOL=volser,SHR
// DLBL SYSTRAC,'dsname',0,SD
// EXTENT SYSTRAC,volser,1,0,start,tracks
```

VSE

CAUTION:
The trace output data set can be very large depending on the information requested by IBM.

ITF Assembler Option

Invoke the Internal Trace Facility by specifying the ITF assembler option.



Where *xx* specifies the suffix of the trace module supplied by IBM.

Appendixes

Appendix A. High Level Assembler Options 171
ASMAOPT 171
ASMAOPT Options 171
ADATA 171
ADEXIT 171
ALIGN 172
ALIGNWARN 172
ASA (MVS and CMS) 173
BATCH 173
CODEPAGE 173
COMPAT 174
CONTWARN 175
DBCS 176
DECK 176
DELETE 176
DSECT 177
DXREF 178
ESD 178
FLAG 178
FOLD 179
GOFF (MVS and CMS) 179
GOFFADATA (MVS and CMS) 180
IMPLENWARN 180
INEXIT 180
INFO 181
LANGUAGE 181
LIBEXIT 182
LIBMAC 182
LIMIT 183
LINECOUNT 183
LIST 183
MAP 184
MXREF 185
OBJECT 185
OBJEXIT 186
PAGE0WARN 187
PCONTROL 188
PESTOP 190
PROFILE 190
PROFMEM 190
PRTEXIT 191
PUSHWARN 191
RA2 192
RECORDINFO 192
RENT 193
RLD 193
RXREF 193
SIZE 194
STORAGE 195
SUBSTRWARN 196

SYSPARMV 196
TERM 196
TEST 197
THREAD 197
TRANSLATE 198
TRMEXIT 198
USING0WARN 198
WARN 199
XOBJADATA (MVS and CMS) 200
XOBJECT (MVS and CMS) 200
XREF 201

Appendix B. High Level Assembler DDNAMES (MVS and CMS) 203
ASMADD Syntax 203
ASMADD 203
ADATA 203
IN 203
LIB 204
LIN 204
OPTN 204
PRINT 204
PUNCH 205
TERM 205
UT1 205

Appendix C. High Level Assembler Service 207

Appendix D. Create Product Parameter File (PPF) Override (VM) 209

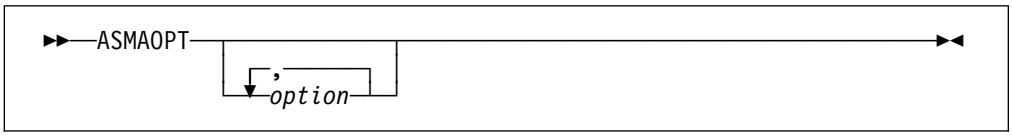
Appendix E. Local Modification Procedures (VM) 211
Full Part Replacement Local Modification 211
Reworking Local Modification 212

Appendix A. High Level Assembler Options

This appendix lists the options for HLASM. In particular, the IBM-supplied default value for each option is indicated. You can use the ASMAOPT installation macro to select different values to be the defaults for your site.

ASMAOPT

The installation macro, ASMAOPT, lets you specify installation defaults for assembler options during installation of High Level Assembler.



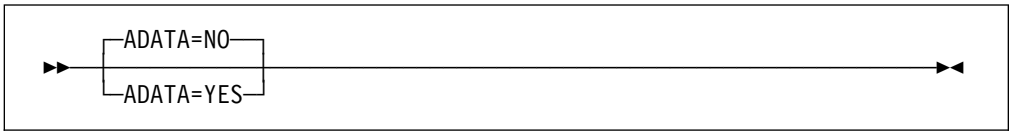
Defaults

If an option is not specified when the ASMAOPT macro is assembled, then the IBM-supplied default is installed for that option. The default value is listed with each option.

ASMAOPT Options

The operands of the ASMAOPT macro, and the values that can specified for each operand, are described below.

ADATA



YES

associated data is written to the file defined SYSADATA DD statement (MVS), the FILEDEF SYSADATA command (CMS), or the DLBL SYSADAT statement (VSE).

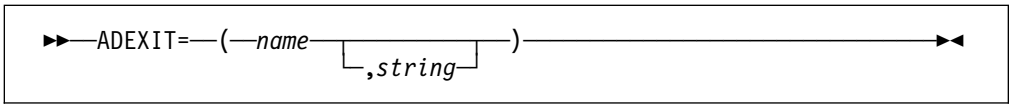
NO

no associated data is collected.

Default

ADATA=NO

ADEXIT



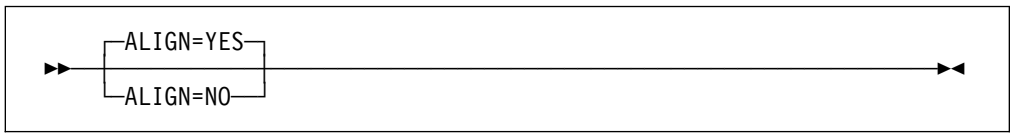
name
identifies the name of a module that is loaded and called by the assembler to monitor the associated data records written by the assembler to SYSADATA (MVS and CMS), or SYSADAT (VSE).

string
the character string that is passed to the exit module as part of the parameter list built by the assembler. The character string is up to 64 characters in length. Any character can be included in the string, subject to the rules for building character strings defined in *HLASM Language Reference*. If the string includes blanks, commas, or parentheses, enclose it in single quotation marks.

Default
No exit specified.

Note: This option can be specified as an assembler invocation parameter by specifying the EXIT(ADEXIT(mod5(str5))) option.

ALIGN



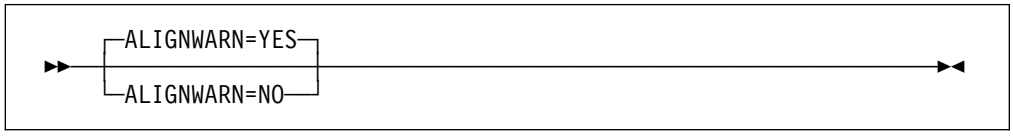
YES
instructs the assembler to check alignment of addresses in machine instructions for consistency with the requirements of the operation code type. DC, DS, DXD, and CXD are aligned on the correct boundaries.

NO
instructs the assembler not to check alignment of unprivileged machine instruction data references, but still to check instruction references and privileged machine instruction data references. DC, DS, and DXD are aligned on the correct boundaries only if the duplication factor is 0.

Default
ALIGN=YES

See also the ALIGNWARN operand on page 172.

ALIGNWARN



YES
instructs the assembler to issue one of the messages ASMA033W, ASMA212W, or ASMA213W when there is an alignment error.

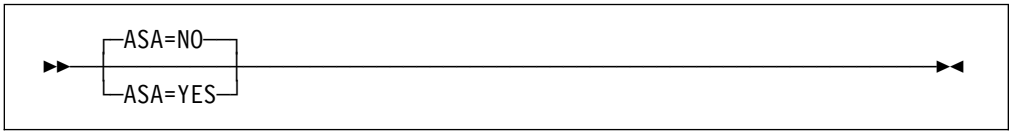
NO
instructs the assembler not to issue a message when there is an alignment error.

Default
ALIGNWARN=YES

See also the ALIGN operand on page 172.

Note: This option can be specified as an assembler invocation parameter by specifying the FLAG(ALIGN) option.

ASA (MVS and CMS)



YES
instructs the assembler to use American National Standard printer-control characters in records written to the assembler listing file.

NO
instructs the assembler to use machine printer-control characters in records written to the assembler listing file.

Default
ASA=NO

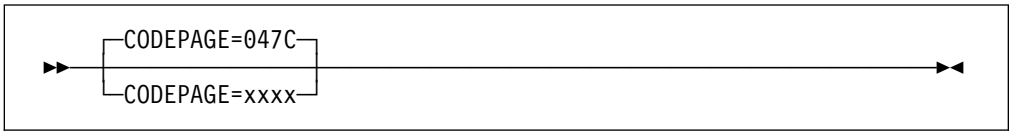
BATCH

YES
instructs the assembler that multiple assembler source programs might be in the input file. The first statement of the second and subsequent source programs must immediately follow the END statement of the previous source program. An end-of-file must immediately follow the last source program.

NO
instructs the assembler that only one assembler source program is in the input file. Statements after the END statement are ignored.

Default
BATCH=YES

CODEPAGE

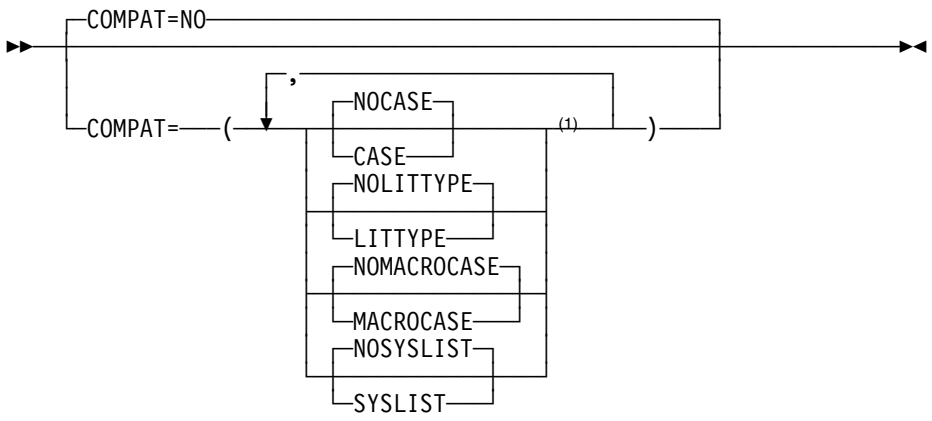


047C
Specifies that characters contained in the Unicode character (CU-type) data constants (DCs) are to be converted using the ECECP: International 1 Unicode-3 mappings contained in module ASMA047C.

xxxx
Specifies that characters contained in the Unicode character (CU-type) data constants (DCs) are to be converted using the Unicode mapping table module ASMA*xxxx* where *xxxx* is the hexadecimal value of the number of the code page contained in the module.

Default
CODEPAGE=047C

COMPAT



Note:
[1] Choose at least one option.

CASE
Instructs the assembler to maintain uppercase alphabetic character set compatibility with earlier assemblers. It restricts language elements to uppercase alphabetic characters *A* through *Z* if they were so restricted in earlier assemblers.

NOCASE
Instructs the assembler to allow a mixed-case alphabetic character set.

LITTYPE
Instructs the assembler to return “U” as the type attribute for all literals.

NOLITTYPE
Instructs the assembler to provide the correct type attribute for literals once they have been defined.

MACROCASE
Instructs the assembler to convert lowercase alphabetic characters (*a* through *z*) in unquoted macro operands to uppercase alphabetic characters (*A* through *Z*).

NOMACROCASE
Instructs the assembler not to convert lowercase alphabetic characters (*a* through *z*) in unquoted macro operands.

SYSLIST
Instructs the assembler to treat sublists in SETC symbols as compatible with earlier assemblers. SETC symbols that are assigned parenthesized sublists are treated as character strings, not sublists, when passed to a macro definition in an operand of a macro instruction.

NOSYSLIST
Instructs the assembler not to treat sublists in SETC symbols as character strings, when passed to a macro definition in an operand of a macro instruction.

NO
Instructs the assembler to allow lowercase alphabetic characters *a* through *z* in all language elements, to treat sublists in SETC symbols as sublists when passed to a macro definition in the operand of a macro instruction, and to provide the correct type attribute for literals once the have been defined.

Default
COMPAT=NO

CONTWARN

YES
the assembler issues diagnostic messages ASMA430W through ASMA433W when one of the following situations occurs:

- The operand on the continued record ends with a comma and a continuation statement is present but continuation does not start in the continue column (usually column 16).
- A list of one or more operands ends with a comma, but the continuation column (usually column 72) is blank.
- The continuation record starts in the continue column (usually column 16) but there is no comma present following the operands on the previous record.
- The continued record is full but the continuation record does not start in the continue column (usually column 16).

NO
the assembler does not issue diagnostic messages ASMA430W through ASMA433W when an inconsistent continuation is encountered.

Default
CONTWARN=YES

Note: This option can be specified as an assembler invocation parameter by specifying the FLAG(CONT) option.

DBCS

YES
instructs the assembler to accept double-byte character set data, and to support graphic (G-type) constants. The assembler recognizes X'0E' and X'0F' in character strings enclosed by single quotation marks, and treat them as shift-out and shift-in control characters for delimiting DBCS data.

NO
the assembler does not recognize X'0E' and X'0F' as double-byte character set data delimiters, and does not support graphic (G-type) constants.

Default
DBCS=NO

DECK

YES
instructs the assembler to place the generated object module in the file defined by the SYSPUNCH DD statement (MVS), the FILEDEF SYSPUNCH command (CMS), or the ASSGN SYSPCH statement (VSE).

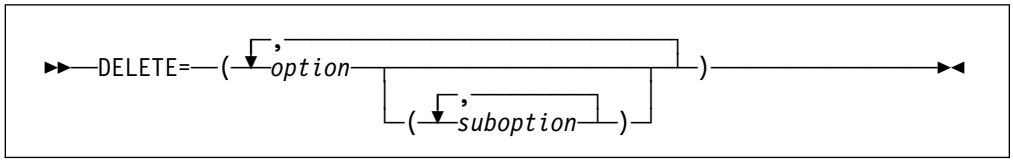
NO
instructs the assembler not to place the generated object module in the file defined by the SYSPUNCH DD statement (MVS), the FILEDEF SYSPUNCH command (CMS), or the ASSGN SYSPCH statement (VSE).

Default
DECK=NO

Important:

VSE The DECK assembler option can only be set using the VSE // OPTION DECK job control statement. The value specified for the DECK option during installation is ignored by the assembler. VSE

DELETE



option
an option that is to have its installation default value fixed. This option cannot be overridden by invocation parameters. The option can be any of the following assembler options:

Options

ADATA	FOLD	OBJECT	STORAGE
ADEXIT	GOFF	OBJEXIT	SUBSTRWARN
ALIGN	GOFFADATA	OPTABLE	SYSPARMV
ALIGNWARN	IMPLENWARN	PAGE0WARN	TERM
ASA	INEXIT	PCONTROL[1]	TEST
BATCH	INFO	PROFILE	THREAD
CODEPAGE	LANGUAGE	PRTEXIT	TRANSLATE
COMPAT[1]	LIBEXIT	PUSHWARN	TRMEXIT
CONTWARN	LIBMAC	RA2	USING0WARN
DBCS	LIMIT	RECORDINFO	WARN
DECK	LINECOUNT	RENT	XOBJADATA
DXREF	LIST	RLD	XOBJECT
ESD	MAP	RXREF	XREF
FLAG	MXREF	SIZE	

Notes:

1. When this option is specified, one or more of the corresponding suboptions must also be specified.

suboption
a suboption of either the COMPAT option or the PCONTROL option that is to have its installation default value fixed. This suboption cannot be overridden by invocation parameters.

When specified in conjunction with the COMPAT option, the suboption can be one of the following:

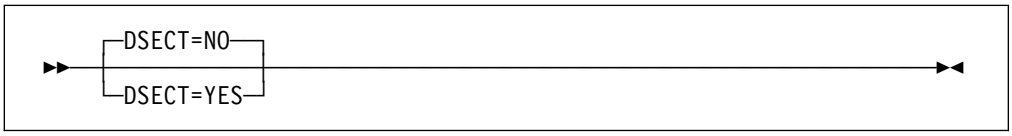
CASE	MACROCASE	NOLITTYPE	NOSYSLIST
LITTYPE	NOCASE	NOMACROCASE	SYSLIST

When specified in conjunction with the PCONTROL option, the suboption can be one of the following:

DATA	MSOURCE	NOMCALL	OFF
GEN	NODATA	NOMSOURCE	ON
MCALL	NOGEN	NOUHEAD	UHEAD

Default
No options deleted.

DSECT

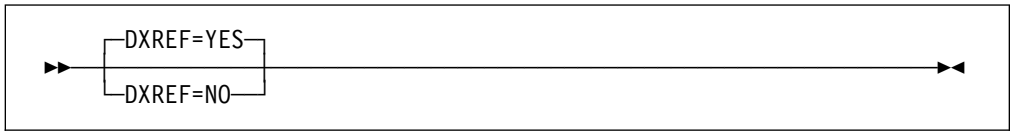


YES
instructs the assembler to produce a DSECT called ASMAOPT.

NO
instructs the assembler to produce a CSECT called ASMADOPT.

Default
DSECT=NO

DXREF

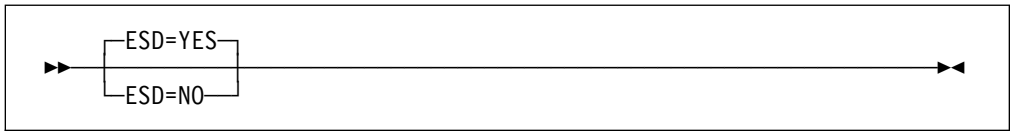


YES
the *DSECT Cross Reference* section is generated as part of the assembler listing. The DSECT cross reference includes the symbolic names of all DSECTs defined in the assembly, the assembled length and the external symbol dictionary identification number (ESDID) of each DSECT, and the number of the statement where the definition of the DSECT began.

NO
the *DSECT Cross Reference* section is not generated.

Default
DXREF=YES

ESD

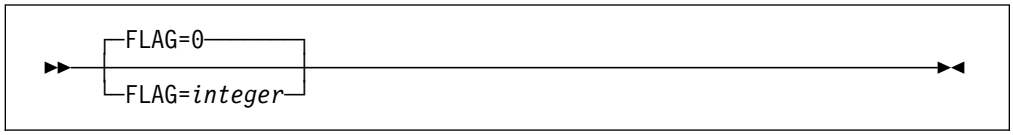


YES
the *External Symbol Dictionary* (ESD) section is generated as part of the assembler listing. The ESD section of the assembler listing contains the external symbol dictionary information that is passed to the linkage editor or loader in the object module.

NO
the ESD section is not included in the assembler listing.

Default
ESD=YES

FLAG

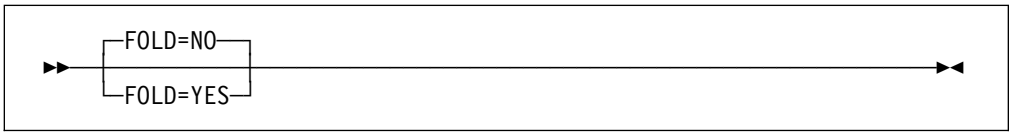


integer
error diagnostic messages with this or a higher severity code are printed in the assembler listing. Error diagnostic messages with a severity code lower than

integer do not appear in the listing, and the severity code associated with those messages are not used to set the return code issued by the assembler. Any severity code from 0 through 255 can be specified. Error diagnostic messages have a severity code of 0, 2, 4, 8, 12, 16, or 20. MNOTEs can have a severity code of 0 through 255.

Default
FLAG=0

FOLD



YES
instructs the assembler to translate lowercase alphabetic characters (a through z) in the assembler listing to uppercase alphabetic characters (A through Z). All lowercase alphabetic characters are translated, including lowercase characters in source statements, assembler error diagnostic messages, and assembler listing lines provided by a user exit. Lowercase alphabetic characters are translated to uppercase alphabetic characters, regardless of the setting of the COMPAT(CASE) option.

NO
lowercase alphabetic characters are not translated to uppercase alphabetic characters.

Default
FOLD=NO

GOFF (MVS and CMS)

YES
the assembler produces a generalized object format data set. The object data set is defined by the SYSLIN DD statement (MVS) or the FILEDEF SYSLIN command (CMS). The generalized object format data set can only be processed by DFSMS/MVS 1.3 or later. This format requires the LIST=133 or LIST=MAX option.

NO
the assembler does not produce a generalized object format data set.

Default
GOFF=NO

Note: The option XOBJECT is treated as a synonym for the GOFF option.

GOFFADATA (MVS and CMS)

YES
the assembler includes the ADATA text record in the extended object format data set (if one is produced — see GOFF). This format requires GOFF=YES.

NO
the assembler does not include the ADATA text record.

Default
GOFFADATA=NO

Note: This option can be specified as an assembler invocation parameter by specifying the GOFF(ADATA) option.

The option XOBJADATA is treated as a synonym for the GOFFADATA option.

IMPLENWARN

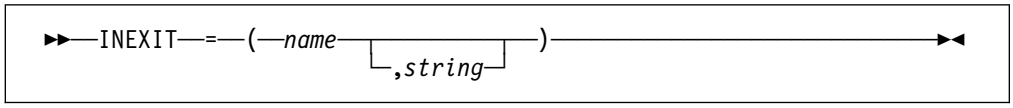
YES
instructs the assembler to issue diagnostic message `ASMA169I` when an explicit length subfield is omitted from an SS-format machine instruction.

NO
instructs the assembler not to issue diagnostic message `ASMA169I` when an explicit length subfield is omitted from an SS-format machine instruction.

Default
IMPLENWARN=NO

Note: This option can be specified as an assembler invocation parameter by specifying the FLAG(IMPLEN) option.

INEXIT



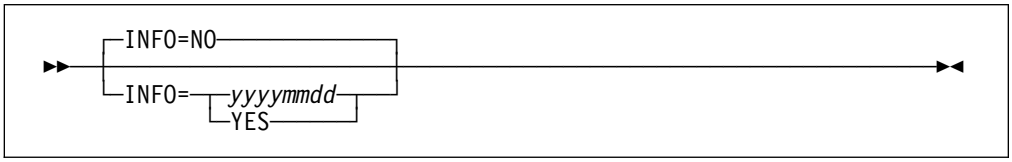
name
identifies the name of a module that is loaded and called by the assembler to obtain source program statements, or to monitor the source program statements read by the assembler from SYSIN (MVS and CMS), or from SYSIPT (VSE).

string
character string that is passed to the exit module as part of the parameter list built by the assembler. The character string is up to 64 characters in length. Any character can be included in the string, subject to the rules for building character strings defined in *HLASM Language Reference*. If the string includes blanks, commas, or parentheses, it must be enclosed in single quotation marks.

Default
No exit specified.

Note: This option can be specified as an assembler invocation parameter by specifying the EXIT(INEXIT(mod1(str1))) option.

INFO



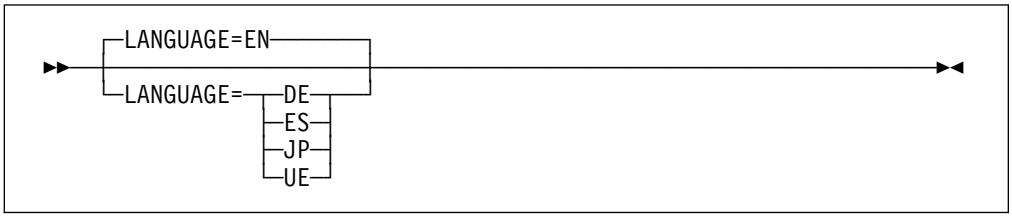
NO
Instructs the assembler not to copy any product information to the list dataset.

yyyymmdd
Instructs the assembler not to copy to the list dataset any product information which is dated prior to *yyyymmdd*.

YES
Instructs the assembler to copy all product information to the list dataset.

Default
INFO=NO

LANGUAGE



EN
diagnostic messages issued by the assembler and assembler listing headings are printed in mixed uppercase and lowercase English.

DE
diagnostic messages issued by the assembler are in German. Assembler listing headings are printed in mixed case English.

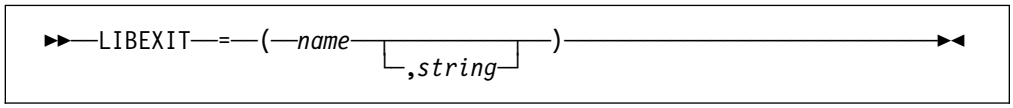
ES
diagnostic messages issued by the assembler are in Spanish. Assembler listing headings are printed in mixed case English.

JP diagnostic messages issued by the assembler are in Japanese. Assembler listing headings are printed in uppercase English.

UE
diagnostic messages issued by the assembler and assembler listing headings are printed in uppercase English.

Default
LANGUAGE=EN

LIBEXIT

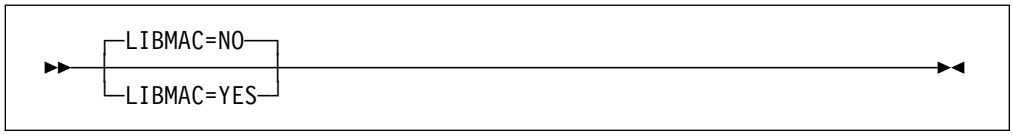


name
identifies the name of a module that is loaded and called by the assembler to obtain macro library or copy library statements, or to monitor the macro library or copy library statements read by the assembler from the file defined by the SYSLIB DD statement (MVS), the GLOBAL MACLIB command (CMS), or the LIBDEF SOURCE statement (VSE). Macro library statements are macro definition statements contained in a macro library. Copy library statements are source program statements contained in a copy library.

string
the character string that is passed to the exit module as part of the parameter list built by the assembler. The character string is up to 64 characters in length. Any character can be included in the string, subject to the rules for building character strings defined in *HLASM Language Reference*. If the string includes blanks, commas, or parentheses, it must be enclosed in single quotation marks.

Default
No exit specified.

LIBMAC



YES
macro definition statements read from a macro library are imbedded in the input source program immediately preceding the first invocation of that macro. The assembler assigns statement numbers to the macro definition statements as though they were included in the input source program.

NO
macro definition statements read from a macro library are not included in the input source program.

Default
LIBMAC=NO

LIMIT

integer
when specified in conjunction with the WARN=8 suboption of the USING option, tells the assembler the maximum displacement that is allowed in base-displacement address resolution before a warning message is issued. When the assembler converts an implicit address (symbolic address) into an explicit address (base-displacement form address), it checks the calculated displacement. If the calculated displacement is greater than the value specified by *integer*, message ASMA304 is issued. *integer* must be a decimal value in the range 0 to 4095. Specifying a value of 4095 is equivalent to specifying LIMIT=NO.

NO
no calculated displacement checking is done.

Default
LIMIT=NO

Note: This option can be specified as an assembler invocation parameter by specifying the LIMIT suboption of the USING option.

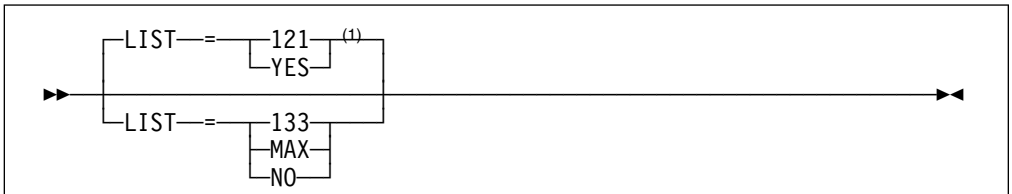
LINECOUNT

integer
the number of lines printed on each page of the assembler listing. *integer* must have a value of 0, or 10 to 3267. If a value of 0 is specified, no page ejects are generated and EJECT, CEJECT, and TITLE statements in the assembly are ignored.

Up to 9 lines on each page might be used for heading lines.

Default
LINECOUNT=60

LIST



Note:
[1] For CMS and MVS, the default i s 121, for VSE the default is YES.

121 (MVS and CMS)
Source and Object section of the assembler listing is produced in the 121-character wide format.

133 (MVS and CMS Ony)
The *Source and Object* section of the assembler listing is produced in the 133-character wide format.

MAX (MVS and CMS)
The *Source and Object* section of the assembler listing is produced in:

121-character wide format
if the logical record length (LRECL) of the listing data set is less than 133.

133-character wide format
if the LRECL of the listing data set is 133 or more.

YES
same as 121.

NO
no assembler listing is produced. If LIST=NO is specified, the options DXREF, ESD, MAP, MXREF, PCONTROL, PRTEXIT, RLD, and XREF are ignored.

Default

CMS, MVS LIST=121 CMS, MVS

VSE LIST=YES VSE

Note: If XOBJECT=YES is specified, LIST=133 or LIST=MAX must be specified.

MAP

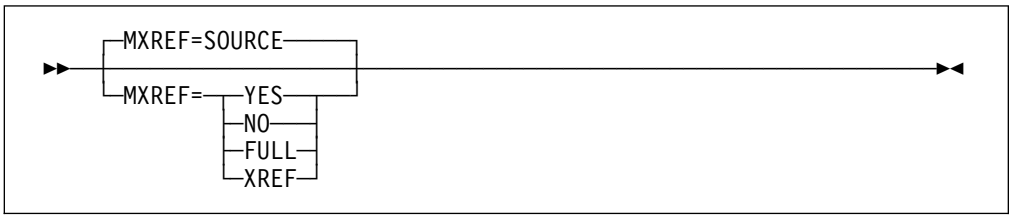
YES
the USING Map section is generated as part of the assembler listing. The USING map is a summary of each USING, DROP, PUSH USING, and POP USING statement in the assembler, including the statement number, the text of the statement, and the registers involved.

NO
the USING map is not generated.

Default
MAP=YES

Note: This option can be specified as an assembler invocation parameter by specifying the MAP suboption of the USING option.

MXREF



SOURCE
the *Macro and Copy Code Source Summary* section only is generated as part of the assembler listing.

YES
same as SOURCE.

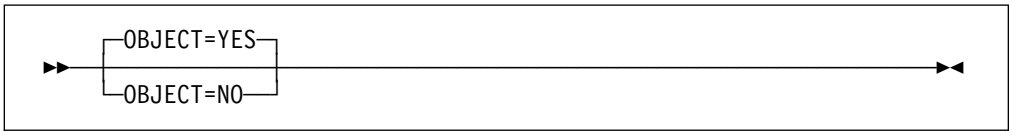
NO
neither the *Macro and Copy Code Source Summary* section nor the *Macro and Copy Code Cross Reference* section are generated as part of the assembler listing.

FULL
the *Macro and Copy Code Source Summary* and *Macro and Copy Code Cross Reference* sections are generated as part of the assembler listing. These sections include the name of each macro library or copy library accessed, the volume serial number of the first DASD volume on which the library resides, and the names of each member retrieved from the library.

XREF
that the *Macro and Copy Code Cross Reference* section only is generated as part of the assembler listing.

Default
MXREF=SOURCE

OBJECT

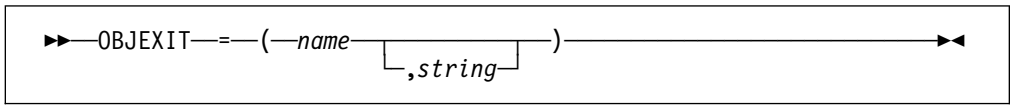


YES
instructs the assembler to place the generated object module in the file defined by the SYSLIN DD statement (MVS), the FILEDEF SYSLIN command (CMS), or the DLBL IJSYSLN and ASSGN SYSLNK statements (VSE).

NO
instructs the assembler not to place the generated object module in the file defined by the SYSLIN DD statement (MVS), the FILEDEF SYSLIN command (CMS), or the DLBL IJSYSLN and ASSGN SYSLNK statements (VSE).

Default
OBJECT=YES

OBJEXIT

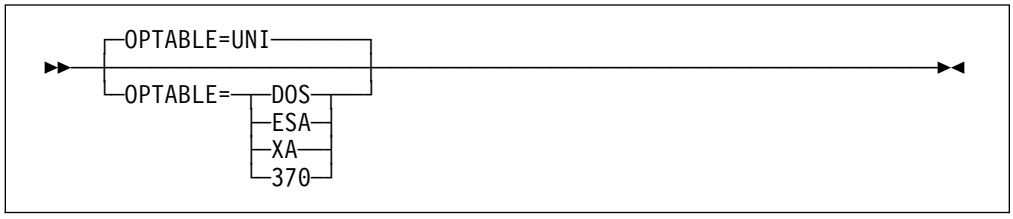


name
identifies the name of a module that is loaded and called by the assembler to receive object module records, or to monitor the object module records written by the assembler to SYSPUNCH or SYSLIN (MVS and CMS), or SYSPCH (VSE).

string
the character string that is passed to the exit module as part of the parameter list built by the assembler. The character string is up to 64 characters in length. Any character can be included in the string, subject to the rules for building character strings defined in *HLASM Language Reference*. If the string includes blanks, commas, or parentheses, it must be enclosed in single quotation marks.

Default
No exit specified.

Note: This option can be specified as an assembler invocation parameter by specifying the EXIT(OBJEXIT(mod5(str5))) option.



DOS
the assembler loads and uses the DOS operation code table. The DOS operation code is designed specifically for assembling programs previously assembled using the DOS/VSE assembler. The operation code table contains the System/370 machine instructions, excluding those with a vector facility.

ESA
the assembler loads and uses the operation code table that contains the symbolic operation codes for the machine instructions specific to systems operating according to the ESA/370 or ESA/390 architecture, including those with a vector facility.

UNI
the assembler loads and uses the operation code table that contains the symbolic operation codes for the machine instructions that can be used on all System/370 and System/390 systems , including those with a vector facility.

XA
the assembler loads and uses the operation code table that contains the symbolic operation codes for the machine instructions specific to systems operating in System/370 extended-architecture mode, including those with a vector facility.

370
the assembler loads and uses the operation code table that contains the symbolic operation codes for the machine instructions specific to System/370 systems, including those with a vector facility.

Default
OPTABLE=UNI

Usage:

1. These operation code tables do not contain symbolic operation codes for machine instructions that are unique to IBM 4300 Processors operating in ECPS:VSE mode.
2. The operation codes supported by High Level Assembler are described in:

   IBM System/370 Principles of Operation, GA22-7000

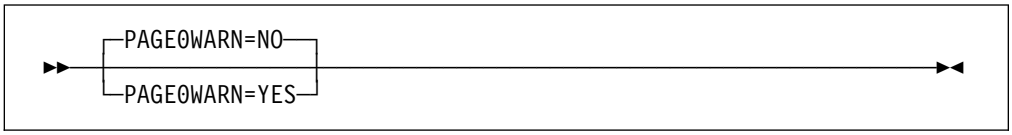
   IBM System/370 Extended Architecture Principles of Operation, SA22-7085

   IBM Enterprise Systems Architecture/370 Principles of Operation, SA22-7200

   IBM Enterprise Systems Architecture/370 and System/370 Vector Operations, SA22-7125

   IBM Enterprise Systems Architecture/390 Principles of Operations, SA22-7205

PAGE0WARN



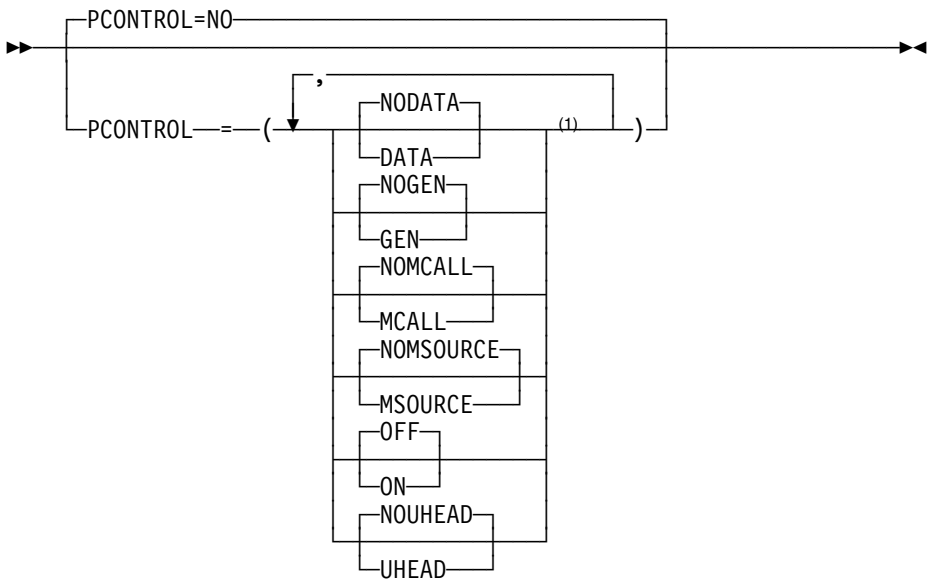
NO
instructs the assembler not to issue diagnostic message `ASMA309W` when an operand is resolved to a baseless address and a base and displacement is expected.

YES
instructs the assembler to issue diagnostic message `ASMA309W` when an operand is resolved to a baseless address and a base and displacement is expected. Since a baseless address is usually correct in the case of load address and all shift instructions this message will not be issued for any of these instructions.

Default
PAGE0WARN=NO

Note: This option can be specified as an assembler invocation parameter by specifying the FLAG(PAGE0) option.

PCONTROL



Note:
[1] Choose at least one option.

DATA
instructs the assembler to print the object code of all constants in full, as though a PRINT DATA statement was specified at the beginning of the source program. All PRINT NODATA statements in the source program are ignored. However, specifying PCONTROL=DATA does not override PRINT OFF or PRINT NOGEN statements in the source program.

NODATA
instructs the assembler to print only the first 8 bytes of the object code of constants, as though a PRINT NODATA statement was specified at the beginning of the source program. All PRINT DATA statements in the source program are ignored.

GEN
instructs the assembler to print all statements generated by the processing of a macro, as though a PRINT GEN statement was specified at the beginning of the source program. All PRINT NOGEN statements in the source program are ignored. However, specifying PCONTROL=GEN does not override PRINT OFF statements in the source program.

NOGEN
instructs the assembler not to print statements generated by the processing of a macro or open code statements with substitution variables, as though a PRINT NOGEN statement was specified at the beginning of the source program. All PRINT GEN and PRINT MSOURCE statements in the source program are ignored.

MCALL
instructs the assembler to print nested macro instructions, as though a PRINT MCALL statement was specified at the beginning of the source program. All PRINT NOMCALL statements in the source program are ignored. However,

specifying PCONTROL=MCALL does not override PRINT OFF or PRINT NOGEN statements in the source program.

NOMCALL
instructs the assembler not to print nested macro instructions, as though a PRINT NOMCALL statement was specified at the beginning of the source program. All PRINT MCALL statements in the source program are ignored.

MSOURCE
instructs the assembler to print the source statements generated during macro processing and the assembled addresses and generated object code of the statements. All PRINT NOMSOURCE statements in the source program are ignored. However, specifying PCONTROL=MSOURCE does not override PRINT OFF or PRINT NOGEN statements in the source program.

NOMSOURCE
instructs the assembler not to print source statements generated during macro processing, but print the assembled addresses and generated object code of the statements. All PRINT MSOURCE statements in the source program are ignored.

ON
instructs the assembler to produce an assembler listing unless the LIST=NO option is specified. All PRINT OFF statements in the source program are ignored.

OFF
instructs the assembler not to produce the *Source and Object* section of the assembler listing. All PRINT ON statements in the source program are ignored.

UHEAD
instructs the assembler to print a summary of active USINGs in the heading lines of each page of the *Source and Object* section of the listing, as though a PRINT UHEAD statement was specified at the beginning of the source program. All PRINT NOUHEAD statements in the source program are ignored. However, specifying PCONTROL=UHEAD does not override PRINT OFF statements in the source program.

NOUHEAD
instructs the assembler not to print a summary of active USINGs, as though a PRINT NOUHEAD statement was specified at the beginning of the source program. All PRINT UHEAD statements in the source program are ignored.

NO
the assembler honors all PRINT statements in the source program. The standard PRINT operands active at the beginning of an assembly are ON, GEN, NODATA, MSOURCE, and UHEAD.

Default
PCONTROL=NO

Note: The PCONTROL option cannot be used to override the LIST=NO option. If the LIST=NO option is specified, the PCONTROL option is ignored.

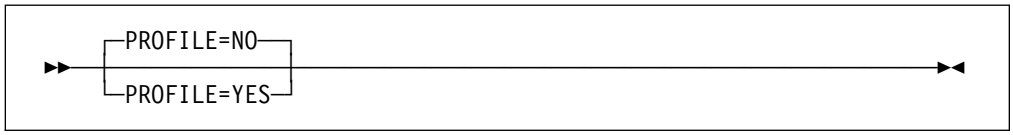
PESTOP

YES
if the assembler detects errors in the invocation parameters or in any *PROCESS statements specified when the assembler is called, all such errors are reported in assembler error diagnostic messages, and the assembly terminates. Similarly, if the invocation parameters or *PROCESS statements contain any options that have a fixed installation default value (the option was specified in the DELETE operand of the ASMAOPT macro), all such attempts to override a fixed installation default are reported in assembler error diagnostic messages, and the assembly terminated.

NO
the assembler reports errors in invocation parameters and in any *PROCESS statements, and continues the assembly using the installation default value for the erroneously specified option. Similarly, the assembler reports any attempts to override fixed installation defaults, and continue the assembly using the default value.

Default
PESTOP=NO

PROFILE



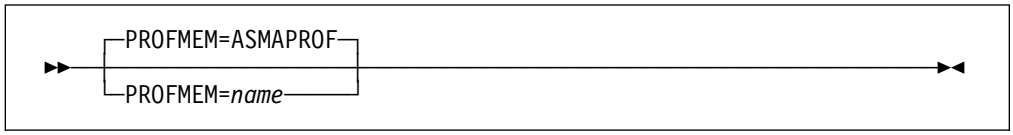
YES
the assembler copies the profile member into the source program, as if the source program contained a COPY instruction as its first statement.

If you do not specify a profile member using the PROFMEM operand, the default member name is ASMAPROF.

NO
the assembler does not copy the profile member into the source program.

Default
PROFILE=NO

PROFMEM



ASMAPROF
the default profile member.

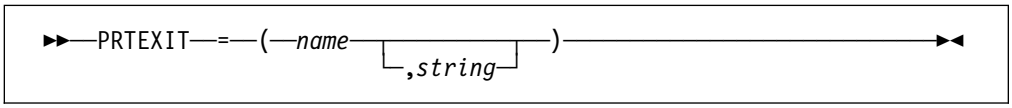
name
the profile member name.

Default
PROFMEM=ASMAPROF

See also the PROFILE operand on page 190.

Note: The member name can be specified as an assembler invocation parameter by specifying the PROFILE(name) option.

PRTEXIT

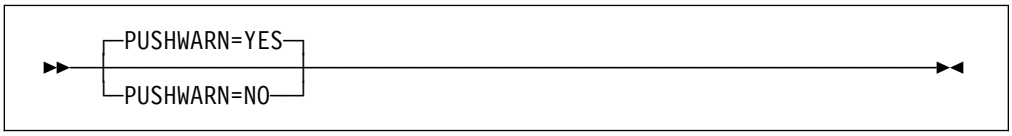


name
identifies the name of a module that is loaded and called by the assembler to receive assembler listing records, or to monitor the assembler listing records written by the assembler to SYSPRINT (MVS and CMS) or SYSLST (VSE).

string
the character string that is passed to the exit module as part of the parameter list built by the assembler. The character string is up to 64 characters in length. Any character can be included in the string, subject to the rules for building character strings defined in *HLASM Language Reference*. If the string includes blanks, commas, or parentheses, it must be enclosed in single quotation marks.

Default
No exit specified.

PUSHWARN



YES
the assembler issues warning diagnostic message ASMA138 when a PUSH/POP stack is not empty at the completion of an assembly.

NO
the assembler suppresses warning diagnostic message ASMA138 when a PUSH/POP stack is not empty at the completion of an assembly.

Default
PUSHWARN=YES

RA2

YES
the assembler suppresses error diagnostic message ASMA066 when 2-byte relocatable address constants, such as AL2(*) and Y(*), are defined in the source program.

NO
the assembler issues error diagnostic message ASMA066 when 2-byte relocatable address constants, such as AL2(*) and Y(*), are defined in the source program.

Default
RA2=NO

RECORDINFO

YES
instructs the assembler to do the following:

- Issue diagnostic message ASMA435I immediately after the last diagnostic message for each statement in error. The message text describes the record number and input data set name of the statement in error.
- Include the member name (if applicable), the record number and the input data set concatenation value with the statement number in the list of flagged statements in the *Diagnostic Cross Reference and Assembler Summary* section of the assembler listing.

NO
instructs the assembler to do the following:

- Not issue diagnostic message ASMA435I for statements in error.
- Only show the statement number in the list of flagged statements in the *Diagnostic Cross Reference and Assembler Summary* section of the assembler listing.

Default
RECORDINFO=YES

Note: This option can be specified as an assembler invocation parameter by specifying the FLAG(RECORD) option.

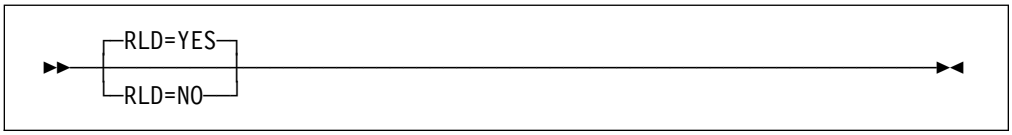
RENT

YES
the assembler checks for possible coding violations of program reenterability.

NO
the assembler does not check for possible coding violations of program reenterability except in executable control sections defined with the RSECT instruction.

Default
RENT=NO

RLD

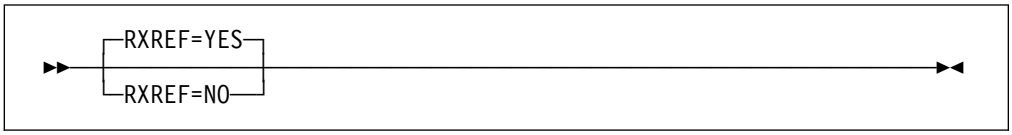


YES
the *Relocation List Dictionary* (RLD) section is generated as part of the assembler listing. The RLD section of the assembler listing contains the relocation list dictionary information that is passed to the linkage editor or loader in the object module.

NO
the RLD is not included in the assembly listing.

Default
RLD=YES

RXREF



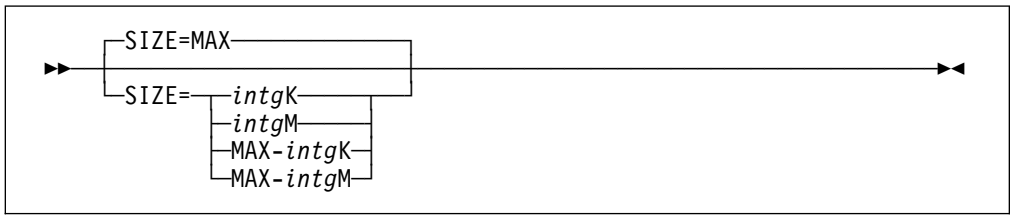
YES
Instructs the assembler to produce the General Purpose Register Cross Reference section of the assembler listing. The General Purpose Register Cross Reference includes:

- The register number
- The statement number which references the register

NO
Instructs the assembler not to produce the General Purpose Register Cross Reference section of the assembler listing.

Default
RXREF=YES

SIZE



SIZE=MAX
specifies that the assembler requests all the available space in the user region (MVS), virtual machine (VM), or in the partition GETVIS (VSE).

SIZE=*intg*K
specifies the amount of virtual storage in 1024-byte (1K) increments.

The minimum acceptable value is 200K.

SIZE=*intg*M
specifies the amount of virtual storage in 1048576-byte (1M) increments.

The minimum acceptable value is 1M.

SIZE=MAX-*intg*K
Specifies that the assembler requests all the available space in the user region (MVS), virtual machine (CMS), or partition GETVIS (VSE), less the amount of *intg*K of storage (1K equals 1024 bytes).

The minimum acceptable value is 1K.

MAX-*intg*M
Specifies that the assembler requests all the available space in the user region (MVS), virtual machine (CMS), or partition GETVIS (VSE), less the amount of *intg*M of storage (1M equals 1048756 bytes).

The minimum acceptable value is 1M.

Notes:

1. The maximum storage value you can specify might not be available in the user region (MVS), virtual machine (CMS), or in the partition GETVIS (VSE),
2. The assembler obtains the storage specified, from above or below the 16mb line depending on the setting of the STORAGE option, see page 195.
3. The minimum amount of working storage required by the assembler is 200K or 10 times the work data set block size, whichever is the greater.
4. When you specify the MAX suboption, the assembler releases 128K back to the user region (MVS), virtual machine (CMS), or the partition GETVIS (VSE), for system usage. When you specify the MAX suboption, there might not be enough storage remaining in the user region (MVS), virtual machine (CMS), or the partition GETVIS (VSE), to load any exits you specify, or any external functions you use in your assembly.

5. The assembler loads user I/O exits before it obtains the working storage. If the user exit obtains storage, then it reduces the amount available for the assembler.
6. The assembler loads external function routines after it obtains working storage. If you use external functions in your program, you should reduce the value you specify in the SIZE option, to allow storage space for the external function modules, and any storage they might acquire.

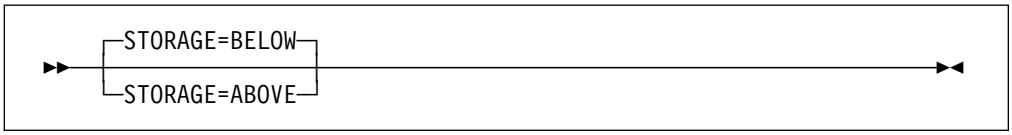
High Level Assembler acquires the amount of storage you specify in the SIZE option from the user region (MVS), virtual machine (CMS) or partition GETVIS (VSE). The assembler only requires a work data set when it has insufficient virtual storage to perform an in-storage assembly. An in-storage assembly usually reduces the elapsed time needed to complete the assembly.

The statistics in the *Diagnostic Cross Reference and Assembler Summary* section of the assembly listing shows the amount of storage the assembler used and an estimate of the amount of storage it requires to perform an in-storage assembly. If you do not provide a work data set, you must specify a large enough value on the SIZE option to allow the assembler to perform an in-storage assembly.

Use the STORAGE operand of the installation default options macro, ASMAOPT, to specify the equivalent of the ABOVE suboption.

Default
SIZE=MAX

STORAGE



BELOW
the assembler obtains the storage specified by the SIZE option below the 16MB line.

ABOVE
the assembler obtains the storage specified by the SIZE option above the 16MB line, if available. If storage above the 16MB line is not available, the assembler obtains the storage below the 16MB line. ABOVE can only be specified if the SIZE option includes the MAX keyword.

Default
STORAGE=BELOW

Note: This option can be specified as an assembler invocation parameter by specifying the SIZE(MAX,ABOVE), SIZE(MAX-integerK,ABOVE), If the Storage option BELOW is used, then the High Level Assembler Option Summary for an assembly which does not specify ABOVE for the assembler SIZE option will only report the first subparameter of SIZE. That is, the Option Summary will not specifically state that BELOW was the default.

SUBSTRWARN

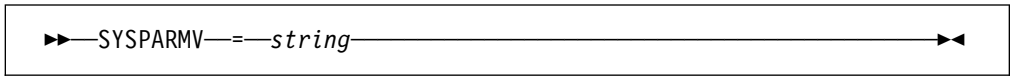
YES
the assembler issues error diagnostic message ASMA094 when the second subscript value of the substring notation indexes past the end of the character expression.

NO
the assembler does not issue error diagnostic message ASMA094 when the second subscript value of the substring notation indexes past the end of the character expression.

Default
SUBSTRWARN=NO

Note: This option can be specified as an assembler invocation parameter by specifying the FLAG(SUBSTR) option.

SYSPARMV

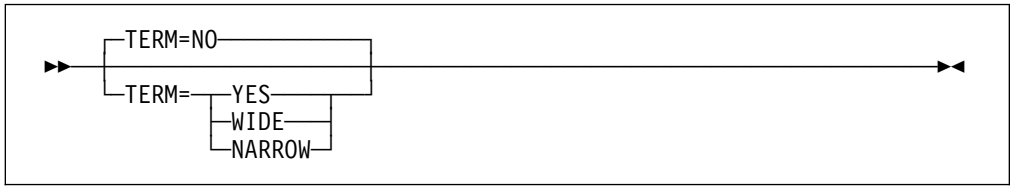


string
the character string used as the default value of the &SYSPARM system variable symbol. The character string is up to 255 characters in length. Any character can be included in the string, subject to the rules for building character strings defined in *HLASM Language Reference*. If the string includes blanks, commas, or parentheses, it must be enclosed in single quotation marks.

Default
SYSPARMV not specified.

Note: This option can be specified as an assembler invocation parameter by specifying the SYSPARM(string) option. (VSE) If a SYSPARM value is provided in the default options, it is not possible to override it with a null string in the // OPTION card. It is possible to override it only with a non-null value. unless you use the invocation parm string to do it.

TERM



YES
equivalent to WIDE. See the description of TERM=WIDE below.

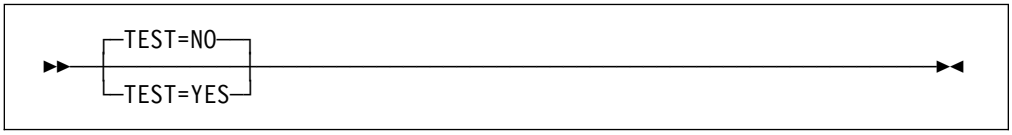
WIDE
assembler messages are written to the file defined by the SYSTERM DD statement (MVS), the FILEDEF SYSTERM statement (CMS), or the ASSGN SYSLOG statement (VSE). Messages written to the terminal file do not have multiple consecutive blanks compressed to a single blank.

NARROW
assembler messages are written to the file defined by the SYSTERM DD statement (MVS), the FILEDEF SYSTERM statement (CMS), or the ASSGN SYSLOG statement (VSE). Messages written to the terminal file have multiple consecutive blanks compressed to a single blank.

NO
assembler messages are not written to the file defined by the SYSTERM DD statement (MVS), the FILEDEF SYSTERM statement (CMS), or the ASSGN SYSLOG statement (VSE).

Default
TERM=NO

TEST



YES
the object module contains the special source symbol table (SYM records). TEST=YES cannot be specified with XOBJECT=YES.

NO
the object module does not contain the special source symbol table (SYM records).

Default
TEST=NO

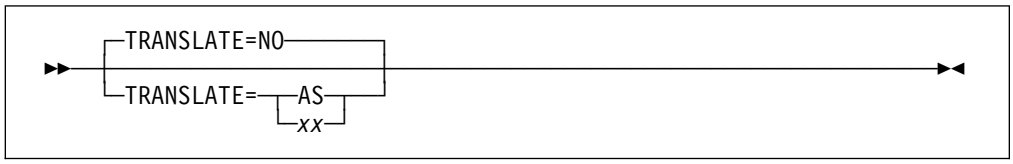
THREAD

YES
instructs the assembler to not reset the location counter at the beginning of each CSECT.

NO
instructs the assembler to reset the location counter at the beginning of each CSECT.

Default
THREAD=YES

TRANSLATE



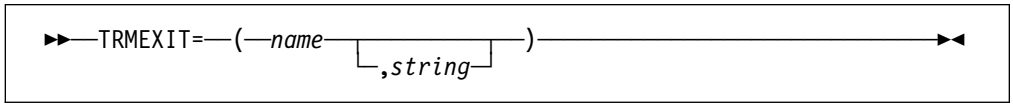
AS
use ASCII translation table ASMALTAS to translate characters contained in data constants and literals.

xx use user translation table ASMALT*xx* to translate characters contained in data constants and literals.

NO
use the standard EBCDIC character set to translate characters contained in data constants and literals.

Default
TRANSLATE=NO

TRMEXIT



name
the user-supplied terminal (SYSTERM) exit. This module is invoked for TERM exit type processing.

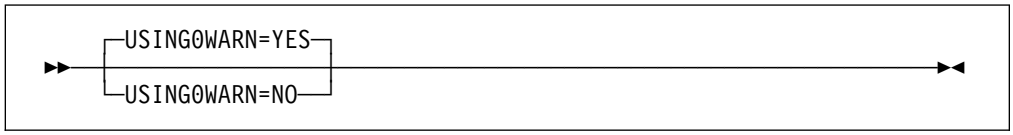
This exit might be used, for example, to write variable-length assembler terminal records.

string
the character string that is passed to the exit module as part of the parameter list built by the assembler. The character string is up to 64 characters in length. Any character can be included in the string, subject to the rules for building character strings defined in *HLASM Language Reference*. If the string includes blanks, commas, or parentheses, it must be enclosed in single quotation marks.

Default
No exit specified.

Note: This option can be specified as an assembler invocation parameter by specifying the EXIT(TRMEXIT(mod6(str6))) option.

USING0WARN



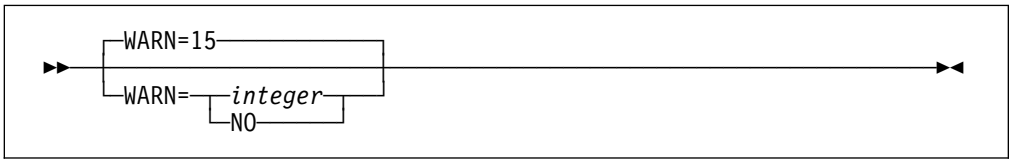
YES
the assembler issues warning diagnostic message ASMA306 when a USING:

- Is coincident with the implied USING 0,0 and option US(WARN(1)) is in effect.
- Overlaps the implied USING 0,0 and option US(WARN(4)) is in effect.

NO
the assembler suppresses warning diagnostic message ASMA306.

Default
USING0WARN=YES

WARN



integer
the conditions under which the assembler issues various warning messages pertaining to the USING instructions used in the source program. The permissible value for *integer* is in the range 1 to 15.

Several conditions can be combined by adding together the associated condition numbers. For example, specifying WARN(12) instructs the assembler to issue warning diagnostic messages for the conditions with condition numbers 4 and 8.

The meaning of the various condition number values for *integer* are:

1 the assembler issues message:

- ASMA300 when a earlier active ordinary (unlabeled) USING's range coincides with and supersedes that of the USING being processed.
- ASMA301 when the range of the USING being processed coincides with and supersedes that of a earlier active ordinary (unlabeled) USING.
- ASMA306 when the range of the USING being processed coincides with the implicit USING 0,0 (for example USING 0,2).

2 the assembler issues message ASMA302 when a USING specifies R0 as a base register, with a non-zero absolute or relocatable expression for the base address.

4 Multiple resolutions: The assembler issues message:

- ASMA303 when multiple resolutions are possible for an implicit address.
- ASMA306 when the range of the USING being processed overlaps the range of the implicit USING 0,0 (for example USING 16,2).

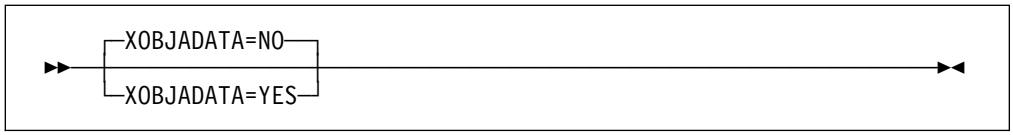
8 the assembler issues message ASMA304 when the calculated displacement in any valid resolution exceeds the threshold specified in the LIMIT suboption.

NO
no USING warning messages are issued.

Default
WARN=15

Note: This option can be specified as an assembler invocation parameter by specifying the WARN suboption of the USING option.

XOBJADATA (MVS and CMS)



YES
the assembler includes the ADATA text record in the generalized object format data set (if one is produced — see XOBJECT). This format requires XOBJECT=YES.

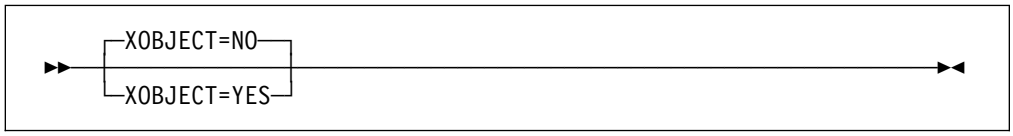
NO
the assembler does not include the ADATA text record.

Default
XOBJADATA=NO

Note:

- This option can be specified as an assembler invocation parameter by specifying the XOBJECT(ADATA) option.
- This option has been superceded by the option GOFFADATA.

XOBJECT (MVS and CMS)



YES
the assembler produces a generalized object format data set. The object data set is defined by the SYSLIN DD statement (MVS) or the FILEDEF SYSLIN command (CMS). The generalized object format data set can only be processed by DFSMS/MVS* 1.3 or later. This format requires the LIST=133 or LIST=MAX option.

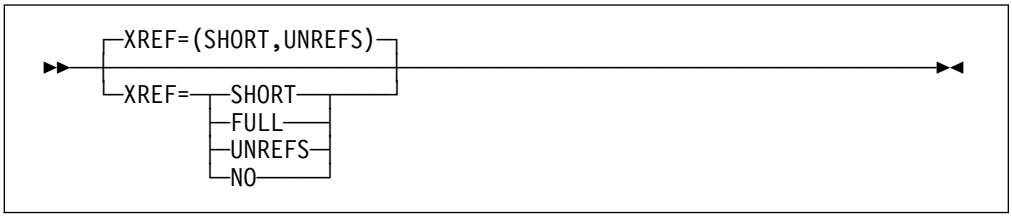
NO
the assembler does not produce a generalized object format data set.

Default
XOBJECT=NO

Note:

- This option has been superceded by the option GOFF.
- See also the XOBJADATA operand on page 200.

XREF



SHORT
symbol cross-reference information for only those symbols that are referred to is generated as part of the assembler listing. Any symbols that are defined, but not referred to, are not included in the cross-reference listing.

FULL
symbol cross-reference information for all symbols used in the assembly is generated as part of the assembler listing. This includes symbols that are defined, but never referred to.

UNREFS
the assembler produces the *Unreferenced Symbols Defined in CSECTs* section of the assembler listing. The symbols are listed in symbol name order. UNREFS can be specified with the SHORT suboption to produce a cross-reference list of referenced symbols. The UNREFS suboption can not be specified with the FULL suboption.

NO
symbol cross-reference information is not generated as part of the assembler listing.

Default
XREF=(SHORT,UNREFS)

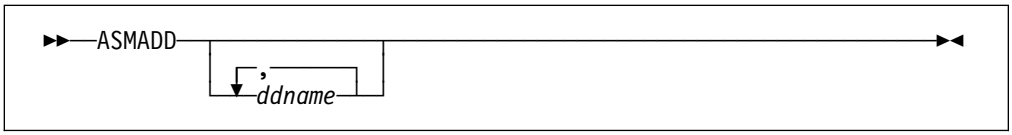
Appendixes

Appendix B. High Level Assembler DDNAMES (MVS and CMS)

The MVS and CMS installation macro, ASMADD, enables you to specify installation default ddnames for the assembler data sets, during installation of High Level Assembler.

ASMADD Syntax

The syntax of the ASMADD installation macro is given below.



ASMADD

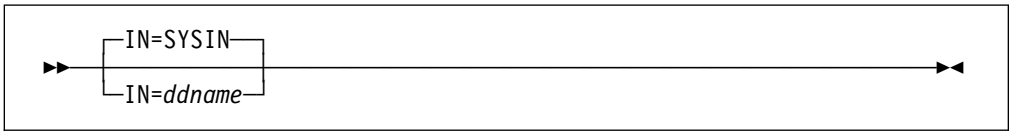
The operands of the ASMADD macro, and the values that can be specified for each operand, are described below. The IBM-supplied default values are shown above the main path.

ADATA

ddname
the *ddname* of the data set containing the assembler language program data that is generated by the assembler when the ADATA assembler option is specified.

Default
ADATA=SYSADATA

IN



ddname
the *ddname* of the input data set containing source statements to be processed.

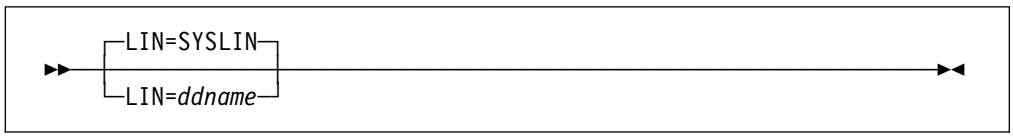
Default
IN=SYSIN

LIB

ddname
the *ddname* of the data set containing system macros, installation-cataloged macros, and members to be included by COPY statements.

Default
LIB=SYSLIB

LIN



ddname
the *ddname* of the data set containing the object module generated by the assembler when the OBJECT or XOBJECT assembler option is specified.

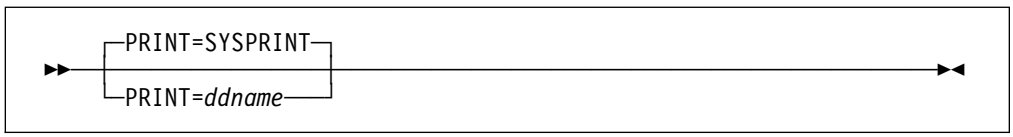
Default
LIN=SYSLIN

OPTN

ddname
the *ddname* of the input data set containing the assembler options.

Default
OPTN=ASMAOPT

PRINT



ddname
the *ddname* of the data set containing the assembler listing generated by the assembler when the LIST assembler option is specified.

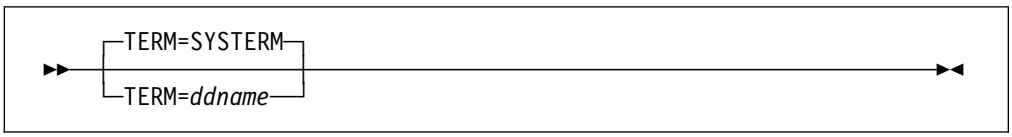
Default
PRINT=SYSPRINT

PUNCH

ddname
the *ddname* of the data set containing the object module generated by the assembler when the DECK assembler option is specified.

Default
PUNCH=SYSPUNCH

TERM



ddname
the *ddname* of the data set containing error messages and assembler completion messages to be displayed at the terminal when the TERM assembler option is specified.

Default
TERM=SYSTERM

UT1

ddname
the *ddname* of the data set used as the assembler work file.

Default
UT1=SYSUT1

Appendixes

Appendix C. High Level Assembler Service

The following service for High Level Assembler Release 3 has been applied to Release 4.

Table 53 (Page 1 of 2). Service History

APAR	MVS PTF	VM PTF	VSE PTF
PQ20235	UQ24747	UQ24764	UQ24746
PQ20847	UQ25843	UQ25845	UQ25844
PQ21028	UQ23747	UQ23750	UQ23748
PQ21274	UQ25902	UQ25901	UQ25903
PQ21980	UQ29640	UQ29646	UQ29641
PQ21982	UQ25944	UQ25946	UQ25945
PQ22004	UQ25572	UQ25573	UQ25568
PQ22005	UQ28142	UQ28141	UQ28143
PQ22017	UQ26107	UQ26106	UQ26108
PQ22954	UQ26672	UQ26603	UQ26671
PQ23317	UQ26237	UQ26235	UQ26234
PQ23558	UQ26334	UQ26338	UQ26337
PQ23562	UQ26360	UQ26414	UQ26361
PQ23999	UQ27814	UQ27816	UQ27818
PQ24125	UQ27615	UQ27522	UQ27535
PQ24169	UQ28198	UQ28180	UQ28197
PQ24177	UQ28958	UQ29014	UQ28959
PQ25312	UQ32401	UQ32405	UQ32411
PQ26118	UQ30452	UQ30450	UQ30453
PQ26966	UQ33113	UQ33038	UQ33114
PQ27386	UQ30692	UQ30690	UQ30691
PQ27453	UQ31033	UQ31032	UQ30975
PQ27904	UQ32441	UQ32440	UQ32457
PQ28052	UQ32493	UQ32492	UQ32494
PQ28212	UQ32579	UQ32568	UQ32569
PQ28852	UQ32700	UQ32703	UQ32701
PQ29589	UQ33495	UQ33502	UQ33494
PQ29608	UQ33622	UQ33627	UQ33625
PQ29618	UQ33412	UQ33413	UQ33422
PQ29959	UQ33955	UQ33960	UQ33991
PQ29995	UQ33863	UQ33853	UQ33852
PQ30097	UQ34176	UQ34181	UQ34175
PQ30279	UQ34432	UQ34421	UQ34430
PQ30758	UQ34940	UQ34941	UQ34943

Table 53 (Page 2 of 2). Service History

APAR	MVS PTF	VM PTF	VSE PTF
PQ30928	UQ35288	UQ35287	UQ35290
PQ32106	UQ36656	UQ36655	UQ36657
PQ32309	UQ36688	UQ36686	UQ36687
PQ32457	UQ36839	UQ36804	UQ36802
PQ32884	UQ37157	UQ37156	UQ37158
PQ33837	UQ38583	UQ38580	UQ38584
PQ34387	UQ39008	UQ39015	UQ39033
PQ34827	UQ39519	UQ39523	UQ39512
PQ35537	UQ40335	UQ40270	UQ40336
PQ35643	UQ40661	UQ40658	UQ40694
PQ35959	UQ41054	UQ41038	UQ41056
PQ36020	UQ41105	UQ41103	UQ41107
PQ36086	UQ41145	UQ41109	UQ41147
PQ36476	UQ41423	UQ41420	UQ41434

Appendix D. Create Product Parameter File (PPF) Override (VM)

This section provides information to help you create a product parameter file (PPF) override. The example used in this section shows how to change the shared file system (SFS) file pool where High Level Assembler files reside.

Note: Do **not** modify the product supplied 5696234E $PPF or 5696234E PPF files to change the file pool name or any other installation parameters. If the 5696234E $PPF file is serviced, the existing $PPF file will be replaced, and any changes to that file will be lost; by creating your own $PPF override, your updates will be preserved.

The following process describes changing the default file pool name, VMSYS, to MYPOOL1:

1. Create a new $PPF override file, or edit the override file created via the 'Make Override Panel' function.

Command	Explanation
xedit *overname* **$PPF** *fm***2**	*overname* is the PPF override file name (such as “myHLASM”) that you want to use. *fm* is an appropriate file mode. If you create this file yourself, specify a file mode of A.

   If you modify an existing override file, specify a file mode of A or D, based on where the file currently resides (A being the file mode of a R/W 191 minidisk, or equivalent; D, that of the MAINT 51D minidisk).

2. Create (or modify as required) the Variable Declarations (:DCL.) section for the HLASMSFS override area, so that it resembles the :DCL. section shown below. This override is used for the installation of High Level Assembler.

```
:OVERLST. HLASMSFS
*
* ================================================================== *
* Override Section for Initial Installation (Using SFS Directories)   *
* ================================================================== *
:HLASMSFS. HLASMSFS 5696234E
:DCL. REPLACE
&191     DIR MYPOOL1:P696234E.
&SAMPZ   DIR MYPOOL1:P696234E.HLASM.LOCAL
&DELTZ   DIR MYPOOL1:P696234E.HLASM.DELTA
&APPLX   DIR MYPOOL1:P696234E.HLASM.APPLYALT
&APPLZ   DIR MYPOOL1:P696234E.HLASM.APPLYPROD
&BLD0Z   DIR MYPOOL1:P696234E.HLASM.TBUILD
&BAS1Z   DIR MYPOOL1:P696234E.HLASM.OBJECT
&HLAID1  USER  P696234E
:EDCL.
:END.
*
```

(This override replaces the :DCL. section of the HLASMSFS override area of the 5696234E $PPF file.)

3. If your $PPF override file was created on the A disk, copy it to file mode D—the Software Inventory minidisk (MAINT 51D). Then erase it from filemode A.

Command	Explanation
copyfile *overname* **$PPF** *fm* **= = d (olddate**	Move PPF file to software inventory disk.
erase *overname* **$PPF** *fm*	Erase redundant file on *fm* disk.

Options:

OLDDATE Use the date and time on the input file as the date and time of last update of the output file.

4. Compile your changes to create the usable *overname* PPF file.

Command	Explanation
vmfppf *overname* **HLASMSFS**	*overname* is the file name of your $PPF override file.

Now that the *overname* PPF file has been created, you should specify *overname* instead of 5696234E as the PPF name for those VMSES/E commands that require a PPF name.

Appendix E. Local Modification Procedures (VM)

If a local modification exists for ASMADOPT or one is planned then the following should be followed.

Full Part Replacement Local Modification

To be done at initial install or first time a local modification is made to the ASMADOPT ASSEMBLE file.

1. Establish HLASM R4 minidisk order

   VMFSETUP 5696234E {HLASM | HLASMSFS}

2. Build a new copy of the MACLIB (This is needed so you get the current copy of any macros that the assemble file needs.)

   VMFBLD PPF 5696234E {HLASM | HLASMSFS} ASMBLMAC (SERVICED

3. Make a copy of the assemble file on the 2C2 (E-disk) using the local modid as part of the file type.

   COPYFILE ASMADOPT ASSEMBLE *fm-2c2*= **ASML***nnnn fm-2c2*

   Note: ASML is a required filetype for local modifications. The nnnn is a user-defined number assigned to this fix

4. Incorporate your local modification into the copy of the assemble (ASML*nnnn*) file on your LOCALSAM 2C2 disk.

5. Update the VVT tables for both the TXT and ASM files by issuing the following VMFSIM commands.

   VMFSIM LOGMOD 5696234E VVTLCL E TDATA :MOD LCL*nnnn***:PART ASMADOPT TXT**

   VMFSIM LOGMOD 5696234E VVTLCL E TDATA :MOD LCL*nnnn***:PART ASMADOPT ASM**

6. Build the new assemble file to be used in next step

   VMFBLD PPF 5696234E {HLASM | HLASMSFS} ASMBLSAM ASMADOPT (ALL

7. Issue the assemble command for the file.

   VMFHLASM ASMADOPT 5696234E {HLASM | HLASMSFS} ($SELECT OUTMODE E NOCKGEN HLASM NOSEG EHLASM

   Note: Consult VMSES/E Introduction and Reference for additional information, for extra ASSEMBLE options.

   If the assemble function is successful, then file ASMADOPT TXTL*nnnn* will be placed on the LOCALSAM 2C2 (E) disk

8. Build your local modification on the test build disk by issuing the following command.

   VMFBLD PPF 5696234E {HLASM | HLASMSFS} (SERVICED

9. Place the local modification into production

   Typically this would be placing the ASMADOPT module on MAINT's 19E disk. Use VMFCOPY from within MAINT's userid.

Reworking Local Modification

To rework a full part replacement local modification to the ASMADOPT ASSEMBLE file do the following.

1. Establish HLASM R4 minidisk order

   VMFSETUP 5696234E {HLASM | HLASMSFS}

2. If you are re-working your local modification to an assemble file because the assemble file was just serviced then you need to find the highest level of IBM service to the part because you will need to compare the IBM serviced level to your local mod'ed level. To do this, you need to issue a VMFSIM GETLVL against just that assemble file.

   VMFSIM GETLVL 5696234E {HLASM | HLASMSFS} TDATA :PART ASMADOPT ASM (HISTORY

   The command returns information to determine the highest level of this part as shipped by IBM. If the response contains the :PTF tag, then there is IBM service to the part. The highest level of IBM service to the part is fn ftabbrev-ptfnumber. Use these names for the file name and file type of the highest level of the part. If the response does not contain the :PTF tag, then there is no IBM service to the part.

3. Compare your local modification on the LOCALSAM 2C2 disk with the highest level of IBM service. Make any changes necessary.

4. If there were any changes made to your local mod then build a new copy of the ASSEMBLE file.

   VMFBLD PPF 5696234E {HLASM | HLASMSFS} ASMBLSAM ASMADOPT (ALL

5. If you are re-working your local modification to an assemble file because the macro and/or text file was serviced that this assemble file uses then you need to rebuild the new macro or maclib, if it is in one, before during the assembly. To do this you need to issue a VMFBLD command.

   VMFBLD PPF 5696234E {HLASM | HLASMSFS} ASMBLMAC (SERVICED

6. Issue the assemble command for the file.

   VMFHLASM ASMADOPT 5696234E {HLASM | HLASMSFS} ($SELECT OUTMODE E HLASM NOSEG EHLASM

   Note: Consult VMSES/E Introduction and Reference for additional information, for extra ASSEMBLE options.

   If the assemble function is successful, then file ASMADOPT TXTL*nnnn* will be placed on the LOCALSAM 2C2 (E) disk

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. 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 IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
522 South Road
Poughkeepsie New York 12601-5400
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

BookManager
IBM
LibraryReader
MVS/ESA
OS/390
OS/2
SP
VM/ESA
VSE/ESA
400

Glossary

This glossary defines terms and abbreviations that are used in this book. If you do not find the term you are looking for refer to the index, to the glossary of the appropriate high-level language (HLL) manual, or to the *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

A

abend. Abnormal end of application.

accept. An SMP/E process that moves distributed code and programs to the distribution libraries.

activate. To make a program available for use.

addressing mode (AMODE). An attribute that refers to the address length that a routine is prepared to handle upon entry. Addresses may be 24 or 31 bits long.

address space. Domain of addresses that are accessible by an application.

AMODE. Addressing mode.

APAR. Authorized program analysis report.

authorized program analysis report (APAR). A request for correction of a problem caused by a defect in a current release of a program.

authorized program facility (APF). The authorized program facility (APF) is a facility that an installation manager uses to protect the system. In MVS, certain system functions, such as all or part of some SVCs, are sensitive; their use must be restricted to users who are authorized. An authorized program is one that executes in supervisor state, or with APF authorization.

auxiliary file. In CMS, a file that contains a list of file types of update files to be applied to a particular source file.

B

base. The core product, upon which features may be separately ordered and installed.

batch. Pertaining to activity involving little or no user action. Contrast with *interactive*.

byte. The basic unit of storage addressability, usually with a length of 8 bits.

C

CBIPO. Custom-Built Installation Process Offering.

CBPDO. Custom-Built Product Delivery Offering.

CE. IBM customer engineer.

CLIST. TSO command list.

CMS. Conversational monitor system.

compiler options. Keywords that can be specified to control certain aspects of compilation. Compiler options can control the nature of the load module generated by the compiler, the types of printed output to be produced, the efficient use of the compiler, and the destination of error messages.

component. (1) Software that is part of a functional unit. (2) A set of modules that performs a major function within a system.

condition code. A code that reflects the result of a previous input/output, arithmetic, or logical operation.

control block. A storage area used by a computer program to hold control information.

control file. In CMS, a file that contains records that identify the updates to be applied and the macrolibraries, if any, needed to assemble a particular source program.

control program (CP). A computer program designed to schedule and to supervise the execution of programs of a computer system.

control section (CSECT). The part of a program specified by the programmer to be a relocatable unit, all elements of which are to be loaded into adjoining main storage locations.

control statement. (1) In programming languages, a statement that is used to alter the continuous sequential execution of statements; a control statement can be a conditional statement, such as IF, or an imperative statement, such as STOP. (2) In JCL, a statement in a job that is used in identifying the job or describing its requirements to the operating system.

conversational monitor system (CMS). A virtual machine operating system that provides general interactive time sharing, problem solving, and program development capabilities, and operates only under the control of the VM/370 control program.

corrective maintenance. Maintenance performed specifically to overcome existing problems.

CP command. In VM, a command by which a terminal user controls his or her virtual machine. The VM/370 control program commands are called CP commands.

CPPL. Command processor parameter list.

CP privilege class. In VM, one or more classes assigned to a virtual machine user in the user's VM directory entry; each privilege class allows access to a logical subset of the CP commands.

CSI. Consolidated software inventory data set. See *SMPCSI*.

CSECT. Control section.

cumulative service tape. A tape sent with a new function order, containing all current PTFs for that function.

Custom-Built Installation Process Offering (CBIPO). A CBIPO is a tape that has been specially prepared with the products (at the appropriate release levels) requested by the customer. A CBIPO simplifies installing various products together.

Custom-Built Product Delivery Offering (CBPDO). A CBPDO is a tape that has been specially prepared for installing a particular product and the related service requested by the customer. A CBPDO simplifies installing a product and the service for it.

D

data definition name (DDNAME). The logical name of a file within an application. The DDNAME provides the means for the logical file to be connected to the physical file.

data set. On MVS, a named collection of related data records that is stored and retrieved by an assigned name. Equivalent to a CMS *file*.

data set name (dsname). The data set name on the DD statement in the JCL or the dsname operand of the TSO ALLOC command.

DBCS. Double-byte character set.

DDDEF. Dynamic data definition.

DDNAME. Data definition name.

default. A value that is used when no alternative is specified.

DD statement. In MVS, connects the logical name of a file and the physical name of the file.

DELTA disk. In VM, the virtual disk that contains program temporary fixes (PTFs) that have been installed but not merged.

distribution libraries. IBM-supplied partitioned data sets on tape containing one or more components that the user restores to disk for subsequent inclusion in a new system.

distribution medium. The medium on which software is distributed to the user; for example, 9-track magnetic tape, tape cartridge.

distribution zone. In SMP/E, a group of VSAM records that describe the SYSMODs and elements in the distribution libraries.

DITTO utility. Data Interfile Transfer, Testing and Operations utility.

double-byte character set (DBCS). A collection of characters represented by a 2-byte code.

driving system. The system used to install the program. Contrast with target system.

dsname. Data set name.

dynamic data definition (DDDEF). The process of defining a data set and allocating auxiliary storage space for it while, rather than before, a job step executes.

dynamic storage. Storage acquired as needed at run time. Contrast with *static storage*.

E

ECMODE. Extended control mode.

executable program. (1) A program that has been link-edited and therefore can run in a processor. (2) The set of machine language instructions that constitute the output of the compilation of a source program.

Extended control mode (ECMODE). A mode in which all features of a System/370 computing system, including dynamic address translation, are operational.

Extended Service Option (ESO). A service option that gives a customer all the new fixes for problems in IBM licensed programs that operate under that customer's operating system.

F

feature. A part of an IBM product that may be ordered separately by a customer.

feature number. A four-digit code used by IBM to process hardware and software orders.

file. A named collection of related data records that is stored and retrieved by an assigned name. Equivalent to an MVS *data set*.

FILEDEF. File definition statement.

file definition statement (FILEDEF). In CMS, connects the logical name of a file and the physical name of a file.

fix. A correction of an error in a program, usually a temporary correction or bypass of defective code.

FMID. Function modification identifier.

Fortran. A high-level language primarily designed for applications involving numeric computations.

function. A routine that is invoked by coding its name in an expression. The routine passes a result back to the invoker through the routine name.

function modification identifier (FMID). The value used to distinguish separate parts of a product. A product tape or cartridge has at least one FMID.

H

I

IBM customer engineer (CE). An IBM service representative who performs maintenance services for IBM hardware.

IBM program support representative (PSR). An IBM service representative who performs maintenance services for IBM software at a centralized IBM location.

IBM service representative. An individual in IBM who performs maintenance services for IBM products or systems.

IBM Software Distribution (ISD). The IBM department responsible for software distribution.

IBM Support Center. The IBM department responsible for software service.

IBM systems engineer (SE). An IBM service representative who performs maintenance services for IBM software in the field.

initial program load (IPL). (1) The initialization procedure that causes an operating system to commence operation. (2) The process by which a configuration image is loaded into storage, as at the beginning of a work day or after a system malfunction or as a means to access updated parts of the system. (3) The process of loading system programs and preparing a system to run jobs.

inline. Sequential execution of instructions, without branching to routines, subroutines, or other programs.

IPL. Initial program load.

interactive. Pertaining to a program or system that alternately accepts input and responds. In an interactive system, a constant dialog exists between user and system. Contrast with *batch*.

Interactive Interface. A series of panels, allowing the user to use the facilities of the VSE/ESA operating. This interface runs within CICS/VSE.

Interactive System Productivity Facility (ISPF). ISPF is a dialog manager for interactive applications. It provides control and services to permit execution of dialogs.

ISD. IBM Software Distribution.

J

JCL. Job control language.

JCLIN data. The JCL statements associated with the ++JCLIN statement or saved in the SMPJCLIN data set. They are used by SMP/E to update the target zone when the SYSMOD is applied. Optionally, SMP/E can use the JCLIN data to update the distribution zone when the SYSMOD is accepted.

JES. Job Entry Subsystem

Job Entry Subsystem. A system facility for spooling, job queueing, and managing the scheduler work area.

job control language (JCL). A sequence of commands used to identify a job to an operating system and to describe a job's requirements.

job step. You enter a program into the operating system as a job step. A job step consists of the job control statements that request and control execution of a program and request the resources needed to run the program. A job step is identified by an EXEC statement. The job step can also contain data needed by the program. The operating system distinguishes job control statements from data by the contents of the record.

K

L

librarian. In VSE, the set of programs that maintains, services, and organizes the system and private libraries.

library. A collection of functions, subroutines, or other data.

link pack area (LPA). In MVS, an area of main storage containing reenterable routines from system libraries. Their presence in main storage saves loading time when a reenterable routine is needed.

linkage editor. A program that resolves cross-references between separately assembled object modules and then assigns final addresses to create a single relocatable load module. The linkage editor then stores the load module in a program library in main storage.

link-edit. To create a loadable computer program by means of a linkage editor.

load module. An application or routine in a form suitable for execution. The application or routine has been compiled and link-edited; that is, address constants have been resolved.

logical saved segment. A portion of a physical saved segment that CMS can manipulate. Each logical saved segment can contain different types of program objects, such as modules, text files, execs, callable services libraries, language repositories, user-defined objects, or a single minidisk directory. A system segment identification file (SYSTEM SEGID) associates a logical saved segment to the physical saved segment in which it resides. See *physical saved segment* and *saved segment*.

LPA. Link pack area.

M

maintain system history program (MSHP). In VSE, a program used for automating and controlling various installation, tailoring, and service activities for a VSE system.

MCS. Modification control statement

minidisk. In VM, all, or a logical subdivision of, a physical disk storage device that has its own address, consecutive storage space for data, and an index or description of stored data so that the data can be accessed. Synonymous with virtual disk.

module. A language construct that consists of procedures or data declarations and can interact with other such constructs.

MSHP. Maintain system history program.

MVS. Multiple Virtual Storage operating system.

N

Named Saved System. A copy of an operating system that a user has named and saved in a file. The user can load the operating system by its name, which is more efficient than loading it by device number.

national language support (NLS). Translation requirements affecting parts of licensed programs; for example, translation of message text and conversion of symbols specific to countries.

NLS. National language support.

nonexecutable components. Components of a product that cannot be run.

non reentrant. A program that cannot be shared by multiple users.

nonreenterable. See *non reentrant*.

NSS. named saved system

O

object code. Output from a compiler or assembler which is itself executable machine code or is suitable for processing to produce executable machine code.

object deck. Synonymous with *object module, text deck*.

object module. A portion of an object program suitable as input to a linkage editor. Synonymous with *text deck, object deck*.

online. (1) Pertaining to a user's ability to interact with a computer. (2) Pertaining to a user's access to a computer via a terminal.

operating system. Software that controls the running of programs; in addition, an operating system may provide services such as resource allocation, scheduling, input/output control, and data management.

P

parameter. Data items that are received by a routine.

partition. A fixed-size division of storage.

phase. In VSE, the smallest complete unit of executable code that can be loaded into virtual storage.

physical saved segment. One or more pages of storage that have been named and retained on a CP-owned volume (DASD). When created, it can be loaded within a virtual machine's address space or outside a virtual machine's address space. Multiple users can load the same copy. A physical saved segment can contain one or more logical saved segments. A system segment identification file (SYSTEM SEGID) associates a physical saved segment to its logical saved segments. See *logical saved segment* and *saved segment*.

preventive maintenance. Maintenance performed specifically to prevent problems from occurring.

preventive service planning (PSP). The online repository of program temporary fixes (PTFs) and other service information. This information could affect installation.

procedure. A named block of code that can be invoked, usually via a call.

procedure library (PROCLIB). A program library in direct access storage with job definitions. The reader/interpreter can be directed to read and interpret a particular job definition by an execute statement in the input stream.

PROCLIB. Procedure library.

program level. The modification, release, version, and fix level of a product.

program number. The seven-digit code (in the format *xxxx-xxx*) used by IBM to identify each program product.

program temporary fix (PTF). A temporary solution or bypass of a problem diagnosed by IBM as resulting from a defect in a current unaltered release of the program.

PSP. Preventive service planning.

PSR. IBM program support representative.

PTF. Program temporary fix.

PWS. Programmable workstation.

Q

qualifier. A modifier that makes a name unique.

R

reentrant. The attribute of a routine or application that allows more than one user to share a single copy of a load module.

reenterable. See *reentrant*

relative file tape (RELFILE tape). A standard label tape made up of two or more files. It contains a file of the MCSs for one or more function SYSMODs and one or more relative files containing unloaded source data sets and unloaded, link-edited object data sets at the distribution library level. A relative file tape is one way of packaging SYSMODs, and is typically used for function SYSMODs.

relative files (RELFILEs). Files containing modification text and JCL input data associated with a SYSMOD.

RELFILEs. Relative files

RELFILE tape. Relative file tape

relocatable load module. On CMS, a combination of object modules having cross references resolved and prepared for loading into storage for execution.

residence mode (RMODE). The attribute of a load module that specifies whether the module, when loaded, must reside below the 16MB virtual storage line or may reside anywhere in virtual storage.

resident modules. A module that remains in a particular area of storage.

return code. A code produced by a routine to indicate its success. It can be used to influence the execution of succeeding instructions.

RIM. Related installation materials

RMODE. Residence mode.

run. To cause a program, utility, or other machine function to be performed.

S

save area. Area of main storage in which contents of registers are saved.

SBCS. Single-byte character set.

SE. IBM systems engineer.

service level. The modification level, release, version, and fix level of a program. The service level incorporates PTFs if there are any.

saved segment. A segment of storage that has been saved and assigned a name. Saved segments can be physical saved segments that CP recognizes or logical saved segments that CMS recognizes. The segments can be loaded and shared among virtual machines, which helps use real storage more efficiently, or a private, nonshared copy can be loaded into a virtual machine. See *logical saved segment* and *physical saved segment*.

shared segment. In VM, a feature of a saved system that allows one or more segments of reenterable code in real storage to be shared among many virtual machines.

shared storage. An area of storage that is the same for each virtual address space. Because it is the same space for all users, information stored there can be shared and does not have to be loaded in the user region.

shared virtual area (SVA). In VSE, a high address area of virtual storage that contains a system directory list (SDL) of frequently used phases, resident programs that can be shared between partitions, and an area for system support.

severity code. A part of run-time messages that indicates the severity of the error condition (1, 2, 3, or 4).

single-byte character set (SBCS). A collection of characters represented by a 1-byte code.

SMPCSI. The SMP/E data set that contains information about the structure of a user's system as well as information needed to install the operating system on a user's system. The SMPCSI DD statement refers specifically to the CSI that contains the global zone. This is also called the master CSI.

softcopy. One or more files that can be electronically distributed, manipulated, and printed by a user.

software inventory disk. In VM, the disk where the system level inventory files reside.

source code. The input to a compiler or assembler, written in a source language.

source program. A set of instructions written in a programming language that must be translated to machine language before the program can be run.

SREL. System release identifier

stacked tape. An IBM-supplied, product-shipment tape containing the code of two or more licensed programs.

statement. In programming languages, a language construct that represents a step in a sequence of actions or a set of declarations.

sublibrary. In VSE, a subdivision of a library.

SUBSET. The value that specifies the function modifier (FMID) for a product level. It further specifies an entry in RETAIN* for a product level.

subsystem. A secondary or subordinate system, or programming support, usually capable of operating independently of or asynchronously with a controlling system. Examples are CICS and IMS.

SVA. Shared virtual area.

syntax. The rules governing the structure of a programming language and the construction of a statement in a programming language.

SYSMOD. system modification.

SYSMOD ID. system modification identifier.

system abend. An abend caused by the operating system's inability to process a routine; can be caused by errors in the logic of the source routine.

T

target disk. In VM, the disk to which a program is installed.

target libraries. In SMP/E, a collection of data sets in which the various parts of an operating system are stored. These data sets are sometimes called system libraries.

target zone. In SMP/E, a collection of VSAM records describing the target system macros, modules, assemblies, load modules, source modules, and libraries copied from DLIBs during system generation, and the system modifications (SYSMODs) applied to the target system.

text deck. Synonym for *object module, object deck*.

time sharing option/extended (TSO/E). An option on the operating system; for System/370, the option provides interactive time sharing from remote terminals.

TSO/E. Time sharing option/extended.

U

UCLIN. In SMP/E, the command used to initiate changes to SMP/E data sets. Actual changes are made by subsequent UCL statements.

UPGRADE. An alphanumeric identifier that specifies a product level.

user exit. A routine that takes control at a specific point in an application.

USERMOD. User modification.

user modification (USERMOD). A change to product code that the customer initiates.

V

virtual machine (VM). (1) A functional simulation of a computer and its associated devices. Each virtual machine is controlled by a suitable operating system. (2) In VM, a functional equivalent of either a System/370 computing system or a System/370-Extended Architecture computing system.

VMFINS. An installation aid supplied as part of VMSES/E to make installation on VM consistent.

VM Serviceability Enhancements Staged/Extended (VMSES/E). A program product for installing and maintaining products on VM.

VMSES/E. VM Serviceability Enhancements Staged/Extended.

VOLSER. Volume serial number.

volume. (1) A certain portion of data, together with its data carrier, that can be handled conveniently as a unit. (2) A data carrier mounted and demounted as a unit; for example, a reel of magnetic tape, a disk pack.

volume label. An area on a standard label tape used to identify the tape volume and its owner. This area is the first 80 bytes and contains VOL 1 in the first four positions.

volume serial number (VOLSER). A number in a volume label assigned when a volume is prepared for use in a system.

VSAM. Virtual storage access method. A high-performance mass storage access method. Three types of data organization are available: entry sequenced data sets (ESDS), key sequenced data sets (KSDS), and relative record data sets (RRDS).

Glossary

Bibliography

High Level Assembler Publications

HLASM General Information, GC26-4943

HLASM Installation and Customization Guide, SC26-3494

HLASM Language Reference, SC26-4940

HLASM Licensed Program Specifications, GC26-4944

HLASM Programmer's Guide, SC26-4941

Toolkit Feature Publications

HLASM Toolkit Feature User's Guide, GC26-8710

HLASM Toolkit Feature Debug Reference Summary, GC26-8712

HLASM Toolkit Feature Interactive Debug Facility User's Guide, GC26-8709

HLASM Toolkit Feature Installation Guide, GC26-8711

Related Publications (Architecture)

Enterprise Systems Architecture/390 Principles of Operation, SA22-7201

Vector Operations, SA22-7207

System/370 Enterprise Systems Architecture Principles of Operation, SA22-7200

System/370 Principles of Operation, GA22-7000

System/370 Extended Architecture Principles of Operation, SA22-7085

Related Publications for MVS

OS/390 MVS:

OS/390 MVS JCL Reference, GC28-1757

OS/390 MVS JCL User's Guide, GC28-1758

OS/390 MVS Programming: Assembler Services Guide, GC28-1757

OS/390 MVS Programming: Assembler Services Reference, GC28-1910

OS/390 MVS Programming: Authorized Assembler Services Guide, GC28-1763

OS/390 MVS Programming: Authorized Assembler Services Reference ALE-DYN, GC28-1764

OS/390 MVS Programming: Authorized Assembler Services Reference ENF-IXG, GC28-1765

OS/390 MVS Programming: Authorized Assembler Services Reference LLA-SDU, GC28-1766

OS/390 MVS Programming: Authorized Assembler Services Reference SET-WTO, GC28-1767

OS/390 MVS System Codes, GC28-1780

OS/390 MVS System Commands, GC28-1781

OS/390 MVS System Messages, Vol 1 (ABA-ASA), GC28-1784

OS/390 MVS System Messages, Vol 2 (ASB-ERB), GC28-1785

OS/390 MVS System Messages, Vol 3 (EWX-IEB), GC28-1786

OS/390 MVS System Messages, Vol 4 (IEC-IFD), GC28-1787

OS/390 MVS System Messages, Vol 5 (IGD-IZP), GC28-1788

MVS/ESA Version 5:

MVS/ESA JCL Reference, GC28-1479

MVS/ESA JCL User's Guide, GC28-1473

MVS/ESA Programming: Assembler Services Guide, GC28-1466

MVS/ESA Programming: Assembler Services Guide, GC28-1474

MVS/ESA Programming: Authorized Assembler Services Guide, GC28-1467

MVS/ESA Programming: Authorized Assembler Services Reference Volumes 1 - 4, GC28-1475, GC28-1476, GC28-1477, GC28-1478

MVS/ESA System Codes, GC28-1486

MVS/ESA System Commands, GC28-1442

MVS/ESA System Messages Volumes 1 - 5 , GC28-1480, GC28-1481, GC28-1482, GC28-1483, GC28-1484

MVS/ESA OpenEdition®:

MVS/ESA OpenEdition MVS User's Guide, SC23-3013

OS/390 OpenEdition:

OS/390 UNIX System Services User's Guide, SC28-1891

MVS/DFP™:

MVS/DFP Version 3.3: Utilities, SC26-4559

MVS/DFP Version 3.3: Linkage Editor and Loader, SC26-4564

DFSMS/MVS®:

OS/390 DFSMS Program Management, SC27-0806

TSO/E (MVS):

TSO/E Command Reference, SC28-1881

TSO/E (OS/390):

OS/390 TSO/E Command Reference, SC28-1969

MVS SMP/E:

SMP/E Messages and Codes, SC28-1108

SMP/E Reference, SC28-1107

SMP/E Reference Summary, SX22-0006

SMP/E User's Guide, SC28-1302

OS/390 SMP/E:

OS/390 SMP/E Messages and Codes, SC28-1738

OS/390 SMP/E Reference, SC28-1806

OS/390 SMP/E User's Guide, SC28-1740

Related Publications for VM

VM/ESA CMS Application Development Guide, SC24-5761

VM/ESA CMS Application Development Guide for Assembler, SC24-5763

VM/ESA CMS Application Development Reference, SC24-5762

VM/ESA CMS Application Development Reference for Assembler, SC24-5764

VM/ESA CMS User's Guide, SC24-5775

VM/ESA XEDIT Command and Macro Reference, SC24-5780

VM/ESA XEDIT User's Guide, SC24-5779

VM/ESA CP Command and Utility Reference, SC24-5750

VM/ESA Planning and Administration, SC24-5773

VMSES/E Introduction and Reference, SC24-5837

VM/ESA Service Guide, SC24-5838

VM/ESA CMS Command Reference, SC24-5776

VM/ESA CMS File Pool Planning, Administration, and Operation, SC24-5751

VM/ESA System Messages and Codes Reference, SC24-5841

Related Publications for VSE

VSE/ESA Administration, SC33-6505

VSE/ESA Guide to System Functions, SC33-6511

VSE/ESA Installation, SC33-6504

VSE/ESA Planning, SC33-6503

VSE/ESA System Control Statements, SC33-6513

General Publications

BRIEF OS/390 Software Management Cookbook, SG24-4775

Index

Special Characters

$SVAASMA 125
$VMFBLD $MSGLOG 91, 93
$VMFINS $MSGLOG message log 65, 69, 70
$VMFREC $MSGLOG 85

A

abend
  See abnormal termination problems
ABENDxxxx keyword 142
abnormal termination problems 142
access VMSES/E programs 63
ADATA
  keyword 146
  option defined 171
ADATA exit 82
adding segment definition 73
ALIGN, FLAG assembler suboption 136
alignment 172, 173
alternate DELTA disk 86
APAR 12, 57, 105, 106
  information required 153—156
  obtaining a list 151
  preparing 153
  service tape, VSE 127
  submitting the documentation 156—157
apply 25
apply check 25
applying maintenance 45, 83
applying service checklist 45, 83
applying service updates 45, 83
ASMA90 module 30, 41, 95
ASMAAPLY job 26
ASMAC procedure 35
ASMACG procedure 35
ASMACL procedure 35
ASMACLG procedure 35
ASMADD
  ADATA 203
  IN 203
  installation macro 203
  LIB 204
  LIN 204
  OPTN 204
  PRINT 204
  PUNCH 205
  TERM 205
  UT1 205
ASMADEF 113
ASMADEL0 47
ASMADITT 112
ASMADOPT 32, 81, 123
ASMAINST 118
ASMAIVP 26
ASMAIVPS 121
ASMALLOC 25
ASMAMACR 120
ASMAOPT 171
ASMAOPT macro
  ADATA 171
  ADEXIT 172
  ALIGN 172
  ALIGNWARN 173
  ASA 173
  BATCH 174
  CODEPAGE 173
  COMPAT 175
  CONTWARN 175
  DBCS 176
  DECK 176
  DELETE 177
  DSECT 178
  DXREF 178
  ESD 178
  FLAG 179
  FOLD 179
  GOFF (MVS and CMS) 179
  GOFFADATA (MVS and CMS) 179
  IMPLENWARN 180
  INEXIT 181
  INFO 181
  LANGUAGE 182
  LIBEXIT 182
  LIBMAC 182
  LIMIT 183
  LINECOUNT 183
  LIST 184
  MAP 184
  MXREF 185
  OBJECT 185
  OBJEXIT 186
  OPTABLE 187
  PAGE0WARN 187
  PCONTROL 187
  PESTOP 190
  PROFILE 190
  PROFMEM 191
  PRTEXIT 191
  PUSHWARN 191
  RA2 192
  RECORDINFO 192

ASMAOPT macro *(continued)*
RENT 193
RLD 193
SIZE 194, 195
SUBSTRWARN 196
SYSPARMV 196
TERM 197
TEST 197
THREAD 197
TRANSLATE 198
TRMEXIT 199
USING0WARN 199
WARN 200
XOBJADATA 200
XOBJECT 201
XREF 201
ASMAOPTL 125
ASMAOPTV 123
ASMARECV 24
ASMASAM1 source 119
ASMASAM2 source 119
ASMASAMP 26, 91
ASMASVA2 125
ASMAVTOC 112
ASMAXADT 32, 82
ASMAXFLU 32, 82
ASMAXFMT 32, 82
ASMAXFSK 32, 82
ASMAXINV 32, 82
ASMAXPRT 32, 82
ASMDDDEF 25
assembler
diagnostic aids 136
error 135
language element as a keyword 159, 160
option
as a keyword 159, 162
list of 136, 154
Assembler H 79, 95
authorized program analysis report
See APAR
Authorized Program Facility (APF) 32

B

backup copy of High Level Assembler (VM) 84
backup of High Level Assembler (MVS) 47
backup of High Level Assembler (VSE) 128
basic material
MVS 4
VM 52
VSE 102
blocks, SFS 66
books
basic publications 5, 53, 103
build message log 93
build saved segment 71
build status messages 90
build status table 70

C

CBIPO RIMs 10
CBPDO 4, 11, 21
changing DDNAMES, MVS 30
changing DDNAMES, VM 81
changing options, MVS 30
changing options, VM 81
changing options, VSE 123
checklist
installation planning for MVS 3
installation planning for VM 51
installation planning for VSE 101
MVS installation 22
VM installation 63
VSE installation 111
clear alternate apply disk 84, 86
clear service 86
coding rules
assembler options syntax 81
DDNAMES syntax 81
compiling PPF 89
component id 47, 96, 131
component identification keyword 141
component name 47, 96, 131
components 4, 52, 102
CONT, FLAG assembler suboption 136
convert macros to source format 119
cumulative service tape 53, 103
current service viii
customization
add options entry for SMP/E 31
considerations 13, 59, 107
create IEV90 alias 43
LPA considerations 19
LPA move 41
modify DDNAMES 32, 81
modify options 32, 81
old procedure names 17
SMP/E considerations 29
standard procedures 35
supplied procedures 17
user exits 16, 62
worksheet, DDNAMES 16, 61
worksheet, MVS 14
worksheet, VM 59
worksheet, VSE 107
customization checklist 29

D

DASD storage
  MVS 6
  VM 55
  VSE 103
DDNAMES
  modifying on MVS 32
  modifying on VM 81
de-editing macros 119
define saved segment 71
delete fmids 10
delete product
  MVS 47
  VM 96
  VSE 130
DELTA disk alternate 86
diagnosis procedure 135
diagnostics 173, 175, 180, 187, 192, 196
directory entry location 65
directory maintenance 66
DIRMAINT 66
disk order 85
distribution libraries 5, 25
distribution media 4, 52, 102
DITTO DVT 112
DOC keyword 148
documentation
  High Level Assembler 223
  related publications 224
documentation problems 148
driving system 5
dsspace 23
DUPLICATE keyword 146
duplicate output
  See output problems
DXREF
  assembler option 136, 154
  keyword 146

E

electronic service 84
enroll user 67
error
  abnormal termination 142
  accompanied by message 144
  assembler 135
  diagnosis procedure 135
  documentation 148
  loop 145
  no response 145
  output 146
  slow performance 147
esasegs parameter 72
ESD
  assembler option 136, 154
  keyword 146
ESERV 119
experienced users vii, viii
external symbol dictionary
  See ESD

F

feature number 3, 4, 51, 52, 101, 102
filepool query 67
filepool references 67
fix default option values 177
FLAG assembler option 136, 178
FMID 3, 4, 10, 47
format notation, description viii—x

G

german nls 11, 56, 105

H

holddata 11

I

IBM Support Center
  reporting problems 47, 96, 131
  UPGRADE value and SUBSET ID, MVS 48
  UPGRADE value and SUBSET ID, VM 97
  UPGRADE value and SUBSET ID, VSE 131
IBMLink (ServiceLink) 97
IEBCOPY 4, 23
IEV90 module 41, 79, 95
  SMP/E utility entry 30
IJSYS03 103
IJSYSLN 185
incorrect output
  See output problems
INCORROUT keyword 146
installation jobstream, MVS
  ASMAASM job 31
  ASMAIEV job 43
  ASMAOPTS job 34
  ASMASTD job 35
installation planner 63
installation planning, MVS 3
installation planning, VM 51
installation problems 137
installation tape, MVS 23
installation user ID 68
installation verification, MVS 47
installation verification, VM 91
installation, MVS
  accept 27

installation, MVS *(continued)*
accept batch 27
allocate 228 catalog product libraries 25
apply 25
apply batch 26
ASMAACPT job 27
ASMAAPLY job 26
ASMAIVP job 26
ASMARECV 24
batch 24, 25
checklist 22
copy jobs from tape 23
DDDEF entries 25
experienced users vii
installation verification program (IVP)
service installation 47
interactive 24, 25
modify install jobs 21
overview 21
receive 24
verification 26
installation, VM
allocate resources 65
build status table 70
checklist 63
directory entry 65
experienced users viii
format disks 68
minidisk use 66
override 69
overview 63
place into production 77
planinfo file 65
preparation 63
profile EXEC 68
saved segment 71
SFS use 66
software inventory disk 68
update CMS system disk 77
verify 70
installation, VSE
allocate library space 112
ASMADEF job 113
ASMAINST job 118
ASMAIVPS job 121
ASMAMACR job 120
ASMAOPTL 125
ASMAOPTV job 123
backup original system 111
condition codes and messages 116, 117
create library 113
DITTO DVT job 112
experienced users viii
Interactive Interface
non-stacked tape 116
stacked tape 114
installation, VSE *(continued)*
list directory entries 119
LVTOC job 112
retrace High Level Assembler 119
summary of steps 111
using batch job 117
using the interactive interface 114, 116
installing user ID 54
Interactive Interface 114
internal trace facility
data set attributes 167
invoking 167
ITF assembler option 167
INVALID keyword 146

J

japanese nls 11, 56, 105
JCL - MVS
ASMAASM job 31
ASMAC procedure 37
ASMACG procedure 40
ASMACL procedure 38
ASMACLG procedure 39
ASMAIEV job 43
ASMAOPTS job 34
ASMASTD job 36
procedures 17
sample source 18
JCL, listings required for APAR 155
job control statements required for APAR 155

K

kanji 11, 56, 105
keyword
ABENDxxxx 142
ADATA 146
component identification 141
definition 139
DOC 148
DUPLICATE 146
DXREF 146
ESD 146
INCORROUT 146
INVALID 146
LOOP 145
MESSAGE 146
MISSING 146
modifier 159
MSGx 145
MXREF 146
OBJECT 146
PERFM 147
release level 141
RLD 146

keyword *(continued)*
SOURCE 146
STAT 146
string
building 139
using as a search argument 151
system-type 149
TERM 146
type-of-failure 142
types 139
UMAP 146
XREF 146

L

label information area 114
LIBR
allocate space 113
blocks 103
default library 118
delete 113
DITTO DVT 112
extent 113
LISTDIR 104
LVTOC 112
member names 119
sublibrary 118, 119, 120
libraries
distribution 9
target 8
license inquiry 213
LIST assembler option 136, 154
listings required for APAR 153
loading memo 64
LOCAL disk 79, 81, 95
LOOP keyword 145
LPA
eligible modules 20
reasons for use 19
LVTOC 112

M

machine instructions, publications 223
macro 171, 203
de-edited form 119
MAINT 19E 78, 79, 94, 95
MAINT 51D 54, 84
maintenance 45, 83, 127
manuals
High Level Assembler 223
list for MVS 5
list for VM 53
list for VSE 103
related publications 224
MDISK statement 66
MESSAGE keyword 146
message log 65, 69, 70
message log merged 86
message problems 144
migration 35, 41, 79, 95
minidisk addresses 55
minidisk order 85
minidisk specifics 66
MISSING keyword 146
missing output
See output problems
modifier keyword 159
modifying options 107, 123
ASMADOLK link book 125
ASMAOPT macro 123
ASMAOPTL 124, 125
ASMAOPTV 123
linkedit job 125
worksheet 107
MSGx keyword 145
MSHP
INSTALL product 118
INSTALL statement 118
retrace APARs 128
RETRACE component 118
retrace PTFs 128
service history 128
standard labels 117
MXREF
assembler option 136, 154
keyword 146

N

national language support 3, 11, 56, 105
nls 11, 56, 105
no-response problems 145
notation, description viii—x
NOTE macro 119

O

OBJECT keyword 146
option
See assembler option
options
example 81
installation macro 171
modifying 32, 81, 107
MVS example 34
USERMOD 32
output problems 146

P

PCONTROL assembler option 136
pemax 23
PERFM keyword 147
performance problems 147
physical saved segment 71
placing High Level Assembler into SVA 125
PLANINFO file 65
PPF override 54, 65, 69, 89
prerequisites 45, 83, 127
preventive service planning
  See PSP
problem
  reporting, MVS 47
  reporting, VM 96
  reporting, VSE 131
problem identification worksheet 165
problem isolation 135
procedure names 17
procedures
  cataloged 35
  match environment 16
  supplied 35
ProductPac 12
profile EXEC 68
program services 11, 57, 105
program temporary fix
  See PTF
program update tape
  See PUT
program, source required for APAR 155
PSP 57
  information content 135
  obtain service updates, MVS 11
  obtain service updates, VM 105
  obtain service updates, VSE 131
  UPGRADE value and SUBSET ID, MVS 48
  UPGRADE value and SUBSET ID, VM 97
  UPGRADE value and SUBSET ID, VSE 131
PTF 12, 57, 106
  list of applied PTFs viii
  service tape, MVS 45
  service tape, VM 83
  service tape, VSE 128
publications
  basic, MVS 5
  basic, VM 53
  basic, VSE 103
  general 224
  High Level Assembler 223
  HLASM Toolkit 223
  machine instructions 223
  MVS 223
  useful during install, MVS 12
  useful during install, VM 58
publications *(continued)*
  useful during install, VSE 106
purge named saved system 91
PUT (program update tape) 151

Q

query filepool 67

R

railroad track format, how to read viii—x
RCVPDO 21
re-installing, MVS 45
re-installing, VM 83
re-installing, VSE 127
reapply local service 84
rebuild serviced parts 90
receive
  documentation 85
  message log 85
  service, VM 86
RECORD, FLAG assembler suboption 136
reenterability 191, 193, 199
release level keyword 141
relocation dictionary
  See RLD
remove segment 91
removing High Level Assembler 130
required programs 6, 53, 103
RETAIN release 47, 96, 131
review build message log 91
review message log 87
rework level 10
RIMs 10
RLD
  assembler option 136, 154
  keyword 146

S

sample install jobs, MVS 21
sample program 26, 91, 120
saved segment
  map 72
  planning 56
  reference sources 56
  virtual storage 56
  where built 54
search argument procedure 151
segment
  build process 76, 92
  definition add 73
  definition panel 72
  map 72
  map refreshed 74

segment map 72
service
  checklist, MVS 45
  checklist, VM 83
  checklist, VSE 127
  user ID for VM 84
service level 12, 57, 105
service update level
  See component identification keyword
SETPARM
  ASMAMACR 120
  ASMAOPTV 124
shared file system 54
  directories 55, 85
  directory create 67
  directory examples 67
  GRANT AUTHORITY 68
  newread 68
  read authority 68
  specific install steps 66
  subdirectories 67
  top directory 66
  work space 66
SMP/E
  accept 34, 44, 46
  APPLY 46
  APPLY CHECK 46
  APPLY REDO 45
  assembler 30
  change default assembler 30
  compress 34, 43
  create IEV90 alias 41
  CSI alter 31
  data sets 8
  dialogs 3
  how to use 3
  level 6
  RECEIVE 46
  RESTORE 29
  rework 34, 43
  USERMOD for options 34
  utility 30
  work data sets 7
software inventory
  access disk 64, 84
  reapply local service 84
  shared file system access 54
software support data base
  searching 151
  summary 139
SoftwareXcel Extended 48, 131
SOURCE
  keyword 146
  program required for APAR 155
spanish nls 11, 56, 105
stacked items ix
standard labels 117, 118
standard procedures
  ASMAC 35
  ASMACG 35
  ASMACL 35
  ASMACLG 35
  ASMASTD job 35
  copying 35
  modifying 16
STAT keyword 146
subset ID 11, 48, 57, 97, 105, 131
SVA
  ASMASVA 125
  High Level Assembler phases 109
  planning 109
  SETSDL 110
syntax notation, description viii—x
SYSADAT 171, 172
SYSADATA 171, 172, 203
SYSIN 181, 203
SYSIPT 181
SYSLIB 25, 204
SYSLIN 185, 186, 204
SYSLNK 185
SYSLOG 197
SYSOPTN 204
SYSPCH 176, 186
SYSPRINT 204
SYSPUNCH 176, 186, 205
system history file 104, 117, 118
System Modification Program Extended (SMP/E) 10
system-type keyword 149
SYSTERM 205
sysut1 7, 9, 205
sysut2 7
sysut3 7
sysut4 7

T

tape
  content 4, 52, 102
  cumulative service 12, 53, 103
  density 4, 102
  files 4, 52, 102
  material 4, 52, 102
  non-stacked format 116
  stacked format 114
  virtual address 68
target libraries 5, 25
target system 5
temporary library 9
TERM keyword 146
terminology viii

test case, writing 136
trace facility 167
type-of-failure keyword 142

U

UMAP keyword 146
update build status table 90
upgrade 11, 57, 105
  value, MVS 11, 48
  value, VM 57, 97, 105
user exits 62
  ADATA exit 32, 82
  customizing, VM 62
  Listing exit 32, 82
  Source exit 32, 82
user modifications 29
USING assembler option 136, 154

V

V5696234 EXEC 91
value, VSE 131
verification
  after service 47, 84
  ASMAIVP job 26
  EXEC 91
  sample output, VM 70
verify applied service 130
verify installation
  ASMAIVPS 119
  ASMAMACR 120
virtual tape address 63, 68
VM/System Delivery Option 53
VMFAPP2120W 88
VMFAPPLY 84
VMFBLD 84
VMFMRDSK 84
VMFREC 84
VMFSETUP 85
VMSES/E 54
  $VMFBLD $MSGLOG 91, 93
  $VMFREC $MSGLOG 85
  alternate DELTA disk 86
  apply message log 87
  apply service 87
  build message log 93
  build status messages 90
  clear alternate apply disk 86
  command syntax 83
  compiling PPF 89
  esasegs parameter 72
  LOCAL disk 81
  local modification 81
  local modifications 88
  merged message log 86
VMSES/E *(continued)*
  no segment support 71, 91
  PLANINFO file 65
  PPF override 65, 69, 89
  rebuild serviced parts 90
  receive message log 85
  reference manuals 83
  requisite checking 65
  review build message log 91
  review message log 87
  SEGBLIST EXC00000 file 72
  update build status table 88, 90
  version vector table (VVT) 87
  VMFAPP2120W 88
  VMFAPPLY 84
  VMFAPPLY PPF command 87
  VMFBLD PPF command 76, 88, 90
  VMFBLD rebuild serviced parts 84
  VMFBLD SEGBLD command 92
  VMFBLD update build status table 84
  VMFCOPY 78, 79, 94, 95
  VMFINS BUILD 70
  VMFINS INSTALL 64, 69
  VMFINS INSTALL command 64
  VMFINS PRODLIST file 64
  VMFMRDSK 84
  VMFMRDSK command 86
  VMFPPF command 89
  VMFREC 84
  VMFREC INFO command 85
  VMFREC PPF command 86
  VMFSETUP 85
  VMFSETUP command 85
  VMFSGMAP command 72
  VMFVIEW BUILD command 90, 91, 93
  VMFVIEW command 85, 86, 87
  VMFVIEW INSTALL 65, 69, 70
  VMSES PARTCAT 79, 95
  VVT 87
VSE/ESA
  planning worksheet 101
  publications 106
  system requirements 103
VTOC (volume table of contents) 112

W

wait state 145
warnings 200
what you receive
  service 45, 83, 127
worksheet
  DDNAMES, MVS 16
  DDNAMES, VM 61
  modifying options 107
  options, MVS 14

worksheet *(continued)*
options, VM 59, 81
options, VSE 107
planning to install, MVS 3
planning to install, VM 51
planning to install, VSE 101
problem identification 165
write access software inventory disk 64
WRITE macro 119
WTO macro 119

X

XREF
assembler option 154
keyword 146

We'd Like to Hear from You

High Level Assembler for MVS® & VM & VSE
Installation and Customization Guide
Release 4

Publication No. SC26-3494-02

Please use one of the following ways to send us your comments about this book:

- Mail—Use the Readers' Comments form on the next page. If you are sending the form from a country other than the United States, give it to your local IBM branch office or IBM representative for mailing.
- Fax—Use the Readers' Comments form on the next page and fax it to this U.S. number: 800-426-7773.
- Electronic mail—Use one of the following network IDs:
  - IBMLink: HLASMPUB at STLVM27
  - Internet: COMMENTS@VNET.IBM.COM

  Be sure to include the following with your comments:
  - Title and publication number of this book
  - Your name, address, and telephone number if you would like a reply

Your comments should pertain only to the information in this book and the way the information is presented. To request additional publications, or to comment on other IBM information or the function of IBM products, please give your comments to your IBM representative or to your IBM authorized remarketer.

IBM may use or distribute your comments without obligation.

Readers' Comments

High Level Assembler for MVS® & VM & VSE
Installation and Customization Guide
Release 4

Publication No. SC26-3494-02

How satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Technically accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐
Grammatically correct and consistent	☐	☐	☐	☐	☐
Graphically well designed	☐	☐	☐	☐	☐
Overall satisfaction	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

May we contact you to discuss your comments? ☐ Yes ☐ No

Name

Address

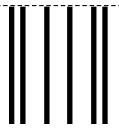
Company or Organization

Phone No.

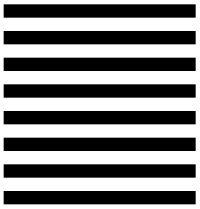
Readers' Comments
SC26-3494-02

Cut or Fold
Along Line

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



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Department J58
International Business Machines Corporation
PO BOX 49023
SAN JOSE CA 95161-9945

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

SC26-3494-02

Cut or Fold
Along Line

Program Number: 5696-234

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

High Level Assembler Publications

SC26-4941 *HLASM Programmer's Guide*.
GC26-4943 *HLASM General Information*.
GC26-4944 *HLASM Licensed Program Specifications*.
SC26-4940 *HLASM Language Reference*.
SC26-3494 *HLASM Installation and Customization Guide*.

High Level Assembler Toolkit Feature Publications

GC26-8709 *HLASM Toolkit Feature Interactive Debug Facility User's Guide*.
GC26-8710 *HLASM Toolkit Feature User's Guide*.
GC26-8711 *HLASM Toolkit Feature Installation Guide*.
GC26-8712 *HLASM Toolkit Feature Debug Reference Summary*.

SC26-3494-02

Spine information:

HLASM Installation

Release 4