High Level Assembler for Linux on zSeries



# User's Guide

Release 6

High Level Assembler for Linux on zSeries



# User's Guide

Release 6

#### - Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 15.

#### Second Edition (July 2008)

This edition applies to IBM High Level Assembler for Linux on zSeries, Release 6, 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 Reader Comments DTX/E269 555 Bailey Avenue SAN JOSE, CA 95141-1003 U.S.A.

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.

© Copyright International Business Machines Corporation 2005, 2008. All rights reserved. US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

## Contents

About this document
Who should use this document
Organization of this document
Summary of changes
IBM High Level Assembler publications v
Hardcopy Publications
Online publications
Related publications
Introduction
Requirements
System requirements
Machine requirements
Storage requirements
Chapter 1. Installation
Space requirements
Chapter 2. Options
Sources of assembler options
Assembler options
Chapter 3. Starting the assembler 5

Chapter 4. Usage and limitations Usage	7 . 7 . 7
Chapter 5. ELF32 option	<b>9</b> . 9 10
Notices	15 16
High Level Assembler Publications	17 17 17 17 17 17 18
Index	19

## About this document

This document describes how to use the IBM High Level Assembler for Linux on zSeries<sup>®</sup> licensed program, hereafter referred to as High Level Assembler, or simply the assembler. This document does not describe the entire assembler. To understand how to use it you will need to consult the other assembler manuals, in particular the *HLASM Language Reference* and the *HLASM Programmer's Guide*. The purpose of this document is to provide the information that is specific to Linux on zSeries (from here on just mentioned as "Linux"). Thus this document provides information like how to install the assembler on Linux, assembler options that are provided only for Linux, and general options that do not work for Linux.

### Who should use this document

The *HLASM for Linux on zSeries User's Guide* is for application programmers coding in the High Level Assembler language. It also describes installation and maintenance tasks. To use this document, you should be familiar with the basic concepts and facilities of your operating system.

### Organization of this document

This document is organized as follows:

- Chapter 1, "Installation," on page 1 describes how you should install the assembler.
- Chapter 2, "Options," on page 3 describes the options that are different from the descriptions in the *HLASM Programmer's Guide*.
- Chapter 3, "Starting the assembler," on page 5 describes how you invoke the assembler, and what parameters you can provide in the invocation.
- Chapter 4, "Usage and limitations," on page 7 tells you about the limitations that apply to assembler on Linux, compared with standard assembler.
- Chapter 5, "ELF32 option," on page 9 provides information about the only Linux-only option, used to create ELF32 files.
- The "Bibliography" on page 17 lists the IBM publications referred to within this document.

### Summary of changes

The changes to this document reflect the change from Release 5 to Release 6 of High Level Assembler.

The contents of this document have been subsumed into the other High Level Assembler documents. See, in particular, *HLASM Programmer's Guide*.

### **IBM High Level Assembler publications**

The standard High Level Assembler runs on z/OS, z/VM and z/VSE. The publications for these platforms are described in this section.

## **Hardcopy Publications**

The books in the High Level Assembler library are shown in Table 1. This figure shows which books can help you with specific tasks, such as application programming.

Task	Publication	Order Number
Evaluation and Planning	HLASM V1R6 General Information	GC26-4943
Installation and Customization	HLASM V1R6 Installation and Customization Guide	SC26-3494
	HLASM V1R6 Programmer's Guide	SC26-4941
	HLASM V1R6 Toolkit Feature Installation Guide	GC26-8711
Application Programming	HLASM V1R6 Programmer's Guide	SC26-4941
	HLASM V1R6 Language Reference	SC26-4940
	HLASM V1R6 General Information	GC26-4943
	HLASM V1R6 Toolkit Feature User's Guide	GC26-8710
	HLASM V1R6 Toolkit Feature Interactive Debug Facility User's Guide	GC26-8709
Diagnosis	HLASM V1R6 Installation and Customization Guide	SC26-3494
Warranty	HLASM V1R6 Licensed Program Specifications	GC26-4944

Table 1. IBM High Level Assembler for Linux on zSeries Publications

### HLASM General Information

Introduces you to the High Level Assembler product by describing what it does and which of your data processing needs it can fill. It is designed to help you evaluate High Level Assembler for your data processing operation and to plan for its use.

HLASM Installation and Customization Guide

Contains the information you need to install and customize, and diagnose failures in, the High Level Assembler product.

The diagnosis section of the book helps users determine if a correction for a similar failure has been documented previously. For problems not documented previously, the book helps users to prepare an APAR. This section is for users who suspect that High Level Assembler is not working correctly because of some defect.

### HLASM Language Reference

Presents the rules for writing assembler language source programs to be assembled using High Level Assembler.

HLASM Licensed Program Specifications

Contains a product description and product warranty information for High Level Assembler.

HLASM Programmer's Guide

Describes how to assemble, debug, and run High Level Assembler programs.

HLASM Toolkit Feature Installation and Customization Guide

Contains the information you need to install and customize, and diagnose failures in, the High Level Assembler Toolkit Feature.

- HLASM Toolkit Feature User's Guide Describes how to use the High Level Assembler Toolkit Feature.
- HLASM Toolkit Feature Debug Reference Summary Contains a reference summary of the High Level Assembler Interactive Debug Facility.
- HLASM Toolkit Feature Interactive Debug Facility User's Guide Describes how to use the High Level Assembler Interactive Debug Facility.

## **Online publications**

The High Level Assembler publications are available in the following softcopy formats:

- *MVS Collection* CD-ROM, SK2T-0710
- OS/390 Collection CD-ROM, SK2T-6700
- VM/ESA Collection CD-ROM, SK2T-2067
- VSE Collection CD-ROM, SK2T-0060

For more information about High Level Assembler, see the High Level Assembler web site, at

http://www.ibm.com/software/ad/hlasm

## **Related publications**

See "Bibliography" on page 17 for a list of publications that supply information you might need while using High Level Assembler.

## Introduction

IBM High Level Assembler for Linux on zSeries is an IBM licensed program that can be used to assemble assembler language programs that use the following machine instructions:

- System/370<sup>™</sup>
- System/370 Extended Architecture (370-XA)
- Enterprise Systems Architecture/370<sup>™</sup> (ESA/370)
- Enterprise Systems Architecture/390<sup>™</sup> (ESA/390)
- z/Architecture<sup>™</sup> machine instructions

### Requirements

This section describes the operating systems, the processors, and the amount of storage required to run High Level Assembler.

### System requirements

High Level Assembler runs under the operating systems listed below. Unless otherwise stated, the assembler also operates under subsequent versions, releases, and modification levels of these systems:

- Linux on zSeries (with 32-bit compatibility environment)
- Linux on S/390

High Level Assembler supports the operation codes available with the following mode processors:

- System/370 Architecture
- Extended Architecture (370-XA)
- Enterprise Systems Architecture/370 (ESA/370)
- Enterprise Systems Architecture/390 (ESA/390)
- z/Architecture

### **Machine requirements**

### For assembling High Level Assembler programs

Programs written using High Level Assembler can be assembled, including use of the z/Architecture processor machine instructions, the Extended Architecture mode processor machine instructions, and Enterprise System Architecture mode processor machine instructions, on all System/370 family and its follow-on machines supporting the following operating systems:

- Linux for zSeries
- Linux for S/390

You might require an operating system-specific macro library to assemble programs that run under that operating system, depending on macro usage.

### For running High Level Assembler programs

A generated object program using z/Architecture, Extended Architecture (370-XA), Enterprise Systems Architecture/370 (ESA/370), Enterprise Systems Architecture/390 (ESA/390), Enterprise Systems/9000 (ES/9000) or Vector instructions can be run only on an applicable processor under an operating system that provides the necessary architecture support for the instructions used.

### Tape device

High Level Assembler is distributed on 3480 tape cartridge.

An appropriate tape device is required for installation.

## Storage requirements

### Virtual storage

High Level Assembler requires a minimum of 650K bytes of main storage. 450K bytes of storage are required for High Level Assembler modules. The rest of the storage allocated to the assembler is used for assembler working storage.

### Auxiliary storage space

Depending on the assembler options used, auxiliary storage space might be required for the following data sets:

- System input
- Macro instruction library—either system or private or both
- Print output
- Object module output

### Library space

The space requirements for the High Level Assembler modules and procedures are provided in Chapter 1, "Installation," on page 1.

## **Chapter 1. Installation**

The install should be a standard rpm install, for example: rpm -v --prefix /usr/ --install asma90-1.6.0-3.s390.rpm

The rpm package is relocatable, so the --prefix option can be used to install HLASM into a different path than the default /usr path if required.

Prior to the install you can query the rpm package. Here are some examples.

• To list the spec file:

rpm -v --query --info --package asma90-1.6.0-3.s390.rpm

To list the files within the rpm package:
rpm -v --query --list --package asma90-1.6.0-3.s390.rpm

If there are any install problems then the files can be manually extracted from the rpm package, for example:

mkdir cpio cd cpio

rpm2cpio asma90-1.6.0-3.s390.rpm | cpio --extract --verbose --make-directories --preserve-modification-time

### Space requirements

The package files will require 742K bytes of disk storage.

## **Chapter 2. Options**

There are some differences between the options available for the assembler on Linux and the options described in the *HLASM Programmer's Guide*.

## Sources of assembler options

- The ASMAOPT file is not available.
- The default options module is not available (the description of this is in the *HLASM Installation and Customization Guide*), so any reference to installation default options do not apply to the Linux assembler.

For more information, see Chapter 3 of the HLASM Programmer's Guide.

## **Assembler options**

There are options that are either not available, unique, or different for the Linux assembler.

### NOADATA

Is the default for the ADATA option. The SYSADATA file is not available, so do not change this option.

**EXIT** This is not available to users on Linux (see ELF32 below).

### NODECK

Default. Do not change.

### **OBJECT**

This is the default. If you specify NOOBJECT then no object file will be written to the object path.

### NOGOFF

If you are using ELF32 (see below) then do not change.

- **SIZE** The default value is 32MB.
- ELF32 Linux only option.

The interface module will elide the ELF32 option and replace it with EX(OBX(ASMAXT2E)) and add the OBJ option if it is not already present. See Chapter 5, "ELF32 option," on page 9 for more details.

Chapter 4 of the *HLASM Programmer's Guide* is not applicable. No user exits are generally available.

Chapter 5 of the *HLASM Programmer's Guide* is not applicable. No external functions are available.

## Chapter 3. Starting the assembler

To start the assembler enter the following (the line spacing is only for readability; enter the command on a single line):

- ./asma90 input\_path\_name
  - -1 list\_path\_name
  - -o object\_path\_name
  - -t term\_path\_name
  - -L syslib\_path\_names
  - -E object\_exit\_path\_name
- input\_path\_name is not prefixed by an identifier while the other dataset names are.
- 2. The identifiers are:
  - -1 for the listing output
  - -o for the GOFF/object output
  - -t for the TERM output
  - -L for the SYSLIB datasets
  - -E for exits when the elf32 option is supplied (see below)

Identifiers are case sensitive.

3. Options are specified as --options='options'.

The options identifier must be prefixed by two hyphens and the options enclosed in quotes.

- 4. It is not necessary to enter the full path name for any file. If the current directory is to be used, no directory information is required except for SYSLIB, where ./ must be supplied.
- 5. Multiple SYSLIB path names can be supplied. Each path name is separated by a colon (for example, path\_name\_1:path\_name\_2:...).
- 6. SYSLIB path name extensions are supported. The extension is specified as path\_name/\*.ext. If more than one extension is required, this can be specified as path\_name/\*.ext1:/path\_name/\*ext2:....

You may need to quote the library directory name, as in './\*.mac', or specify a set of specifications separated by colons, as in ./\*.mac:./\*.MAC:....

- 7. The assembler searches for macros or copy files first in upper case, then in lower case. The case of the extension is not changed. Mixed-case names are not recognized.
- 8. Either ASCII or EBCDIC files may be provided as input. The assembler will convert the ASCII input to EBCDIC for internal processing. The assembler determines if the data is in ASCII by looking for an ASCII blank (X'20'), ASCII asterisk (X'21') or ASCII numbers (X'30' to X'39') in the first record of the file.
  - Care must be exercised in using characters whose encodings are not stable when converted between EBCDIC and ASCII. Examples of such characters include not sign (¬), vertical bar (|), and square brackets ([]).
- 9. TERM messages are written in ASCII. If you specify the TERM option but provide no -t term\_path\_name, output goes to stdout (the terminal), and can be redirected with > to a file.

To create ELF32 object files on Linux on zSeries, specify the HLASM ELF32 option (--options='ELF32'). You will also need to provide the following parameter so that the assembler can locate and load the object exit that does this translation:

-E /usr/bin/asmalib

Starting the assembler

## Chapter 4. Usage and limitations

### Usage

The following should be considered:

- The generated listing is in EBCDIC, and will be difficult to view on Linux. However, the Linux 'dd' command will automatically convert EBCDIC files to ASCII, insert newlines, and so on. The 'man' pages for 'dd' describe its options.
- 2. When you use FTP to copy source and macro files to Linux, if you specify the 'binary' option, the file is transferred in EBCDIC. This may avoid unexpected mappings of EBCDIC characters to ASCII code points that HLASM does not recognize or process correctly.
- **3**. Be careful if you edit files on Linux, because the original 80-byte fixed-length structure of each record may not be retained.
- 4. Source files are recognized as EBCDIC or ASCII by checking the first record. While the test is usually reliable, there are some cases where the need for conversion is not correctly recognized. Also, some characters (such as those in C-type constants) may not have ASCII equivalents that will cause identical object code to be created.
- 5. High Level Assembler for Linux on zSeries does not use a work file and therefore the default size option has been set at 32M. This may be inadequate for large or complex assemblies, but can be increased by specifying the SIZE(xxM) option.
- 6. Transferring object or listing files to and from Linux: if you wish to copy files to or from Linux using FTP, set the 'binary' and 'locsite fix 80' options (for object files) or 'locsite fix 133' (or 121), for listing files, before each 'get'. When running on Linux: set 'binary' and 'put' to VM. Then, you can post-process each file with either

'PIPE <' fn ft fm '| deblock fixed ' 80 '|> ' ofn oft ofm

for object files, or 'PIPE <' fn ft fm '| deblock fixed ' 133 '|> ' ofn oft ofm

for listing files.

## Limitations

- High Level Assembler for Linux on zSeries executes in 32-bit addressing mode. When executing on Linux for zSeries systems running in 64-bit addressing mode, the Linux 32-bit compatibility interface is required for correct execution.
- 2. High Level Assembler for Linux on zSeries converts only OBJ object files to ELF32 format. To create ELF object files from GOFF object files, a separate conversion module is required.

The limitations that apply to the generation of ELF32 object files are listed in Chapter 5, "ELF32 option," on page 9.

## Chapter 5. ELF32 option

This is a Linux only option. It is provided to allow creation of ELF32 files directly.

If the ELF32 option is specified, then changes are made to the options passed to the assembler to invoke the exit.

### Implementation notes

The following considerations apply to generated ELF32 files:

- 1. All section lengths are rounded up to a doubleword.
- 2. ELF32 name lengths are set to the section length for SD, CM, and PC items, and to 4 for LD items. The name of the section is not affected.
- 3. Zero-length sections are assigned length 8.
- 4. RSects are assigned to ELF32 section '.text.RSECT', and are assigned the executable attribute.
- 5. CSects are assigned to ELF32 section '.data.CSECT', and are also assigned the data and executable attributes.
- 6. Common (CM) sections are assigned ELF32 section names '.bss.<name>', where <name> is the name of the CM section.
- 7. High Level Assembler for Linux on zSeries converts only OBJ object files to ELF32 format.
- 8. When generating ELF32 object files, the following limitations apply:
  - a. A minimum of 25K working storage must be available
  - b. The GOFF option must not be specified.
  - **c.** Do not specify external names starting with an underscore (\_), as they may conflict with Linux-generated names during the linking and loading process.
  - d. No entry points (LD items) may be specified in common sections.
  - e. Q-type and CXD-type address constants are not supported.
  - f. External dummy sections (DXD items, and dummy control sections referenced in Q-type address constants) are not supported.
  - g. Address constants of lengths other than 4 are not supported.
  - h. Only AMODE(31) and RMODE(ANY) are supported.
  - i. The following internal tables are currently limited to:
    - 50 control section (SD, CM, PC) names
    - 150 external symbols (SD, CM, PC, ER. WX)
    - 100 entry point (LD) names
    - 150 address constant items (RLD) residing in RSects
    - 150 address constant items (RLD) residing in CSects
    - 50 ELF32 string-table items
    - 100 ELF32 sections
    - 250 ELF32 symbols (including ELF32 section names)

These limits are easily increased by recompiling ASMAXT2E.

- j. Double relocations, although correctly converted, will not work, because the Linux 1d command does not add the previous contents of an adcon field when performing relocations.
- k. Zero-length Private Code (PC) sections are ignored.
- I. Negative relocations (for example, A(0-\*)) are not supported.

### ASMAXT2E messages

ASMAXT2E may generate the following messages. Each message is prefixed with an assembler indication of the form ASMA70ns, where n is 0, 1, 2, 3, or 4, and s is one of the standard severity-indication letters I, W, E, S, T. This indication is followed by the word "OBJECT: ". Then, the message text is preceded by "ASMAXT2E: ".

### Exit not coded at same level (2,3) as Assembler

Severity: 16

**Explanation:** The ASMAXT2E module expects a different I/O exit interface from that supplied by High Level Assembler.

**User response:** This is an internal error. Report the problem to IBM service.

System action: Processing is terminated.

### Exit called for other than PUNCH or OBJECT

Severity: 16

**Explanation:** The ASMAXT2E module was invoked as an exit of type other than PUNCH or OBJECT.

**User response:** This is caused by incorrect assembler invocation options. Correct the options string and re-assemble.

System action: Processing is terminated.

### Exit not initialized, and not entered for OPEN

Severity: 16

**Explanation:** The ASMAXT2E module was expected to be open and initialized, but was not

**User response:** This is an internal error. Report the problem to IBM service.

System action: Processing is terminated.

### Insufficient working storage for exit

Severity: 16

**Explanation:** The ASMAXT2E module requires more working storage than is available.

**User response:** Specify a larger storage area, or specify a smaller SIZE option.

System action: Processing is terminated.

### Invalid action or operation type requested

Severity: 16

**Explanation:** The ASMAXT2E module was invoked as an I/O exit with an invalid action or operation type.

**User response:** This is an internal error. Report the problem to IBM service.

System action: Processing is terminated.

Expecting input record, zero buffer length

Severity: 16

**Explanation:** The ASMAXT2E module expected to receive an object file record to process, but none was present.

**User response:** This is an internal error. Report the problem to IBM service.

System action: Processing is terminated.

### Close request for wrong exit type

Severity: 16

**Explanation:** The ASMAXT2E module received a CLOSE request, but for the wrong exit type.

**User response:** This is an internal error. Report the problem to IBM service.

System action: Processing is terminated.

### Invalid request-list options value

Severity: 12

**Explanation:** The ASMAXT2E module was invoked to process a record, but an options string was provided.

**User response:** This is an internal error. Report the problem to IBM service.

System action: Processing is terminated.

### Invalid parm-string length

Severity: 12

**Explanation:** The ASMAXT2E module was provided with a parameter string, but its length was invalid.

**User response:** Check that the assembler options specifying the I/O exit are valid.

System action: Processing is terminated.

### Too many ESD IDs

Severity: 12

**Explanation:** The program being processed has too many ESD items with distinct ESD IDs.

**User response:** Restructure the program to reduce the number of independently relocatable items.

System action: Processing is terminated.

#### Too many RLD items

Severity: 12

**Explanation:** The program being processed has too many relocatable address constants.

**User response:** Restructure the program to reduce the number of address constants.

System action: Processing is terminated.

#### Too many OBJ SD/CM/PC sections

Severity: 12

**Explanation:** The program being processed has too many control sections.

**User response:** Restructure the program to reduce the number of control sections.

System action: Processing is terminated.

#### Too many ELF sections

Severity: 12

**Explanation:** The program being processed requires more ELF sections than can be provided.

**User response:** Restructure the program to reduce the number of independent items, or contact IBM service to request that ASMAXT2E be enhanced.

System action: Processing is terminated.

#### String table overflow

### Severity: 12

**Explanation:** The program being processed requires more space in the ELF string table than can be provided.

**User response:** Restructure the program to reduce the number of or length of external names, or contact IBM service to request that ASMAXT2E be enhanced.

System action: Processing is terminated.

#### Too many LD items

### Severity: 12

**Explanation:** The program being processed contains too many ENTRY names.

**User response:** Restructure the program to reduce the number ENTRY statements, or contact IBM service to

request that ASMAXT2E be enhanced.

System action: Processing is terminated.

### Too many ELF symbols

Severity: 12

**Explanation:** The program being processed requires more ELF symbols than can be processed.

**User response:** Restructure the program to reduce the number of names, or contact IBM service to request that ASMAXT2E be enhanced.

System action: Processing is terminated.

### Insufficient storage for OBJ TXT records

Severity: 12

**Explanation:** The program being processed requires more work space than is available to process machine language instructions and data.

**User response:** Specify a larger storage area, or specify a smaller SIZE option.

System action: Processing is terminated.

## AMODE/RMODE 24 or 64 not supported in section 'xxxxxxx'

Severity: 8

**Explanation:** The program being processed contains a control section *xxxxxxxx* with an unsupported AMODE or RMODE.

**User response:** Modify the program to specify valid AMODE and RMODE.

**System action:** The section is discarded, and processing continues. No ELF object file will be produced.

### External Dummy (XD) item 'xxxxxxx' not supported

### Severity: 8

**Explanation:** The program being processed contains a dummy external control section (XD) *xxxxxxxx*.

**User response:** Modify the program to remove the XD item.

**System action:** The item is discarded, and processing continues. No ELF object file will be produced.

## Adcon at xxxxxxx in section with ID xxxx not 4 bytes long

Severity: 8

**Explanation:** The program being processed contains an address constant at address *xxxxxxxx* in a control section with ESDID *xxxx* that is not 4 bytes long.

### ASMAXT2E messages

**User response:** Modify the program to correct the address constant.

**System action:** The item is discarded, and processing continues. No ELF object file will be produced.

## Adcon at xxxxxxx in section with ID xxxx not type A or V

### Severity: 8

**Explanation:** The program being processed contains an address constant at address *xxxxxxxx* in a control section with ESDID *xxxx* of type Q, or DXD, or CXD.

**User response:** Modify the program to remove the address constant.

**System action:** The item is discarded, and processing continues. No ELF object file will be produced.

## Adcon at xxxxxxx in section with ID xxxx requires unsupported negative relocation

### Severity: 8

**Explanation:** The program being processed contains an address constant at address *xxxxxxxx* in a control section with ESDID *xxxx* which requires negative relocation.

**User response:** Modify the program to remove the address constant.

**System action:** The item is discarded, and processing continues. No ELF object file will be produced.

### ELF file not produced due to previous errors

Severity: 8

**Explanation:** Previous errors have suppressed the production of an ELF object file.

User response: Correct the errors.

### Section length on END record not supported

### Severity: 8

**Explanation:** The OBJ END record contains the length of a control section.

**User response:** The object file may be very old, possibly produced by a language translator that put the section length on the OBJ END record. (This condition will not arise with object files produced by High Level Assembler.)

**System action:** The item is discarded, and processing continues. No ELF object file will be produced.

### Invalid END-record entry point request

Severity: 8

**Explanation:** The OBJ END record contains a request for a particular entry point in the program.

**User response:** Remove the entry point request from the END record.

**System action:** The item is discarded, and processing continues. No ELF object file will be produced.

### LD item 'xxxxxxx' in unsupported section is ignored

Severity: 8

**Explanation:** The ENTRY name *xxxxxxx* is in a control section that was previously discarded.

**User response:** Correct the original error causing the section to be rejected.

**System action:** The item is discarded, and processing continues. No ELF object file will be produced.

## RLD item at address 'xxxxxxx' in unsupported section with ID 'xxxx' is ignored

Severity: 8

**Explanation:** The RLD item at address *xxxxxxxx* in a control section with ESDID *xxxx* is in a control section that was previously discarded.

**User response:** Correct the original error causing the section to be rejected.

**System action:** The item is discarded, and processing continues. No ELF object file will be produced.

## RLD item at address 'xxxxxxx' in unsupported section with ID 'xxxx' references unsupported section

### Severity: 8

**Explanation:** The RLD item at address *xxxxxxxx* in a control section with ESDID *xxxx* references a position in a control section that was previously discarded.

**User response:** Correct the original error causing the section to be rejected.

**System action:** The item is discarded, and processing continues. No ELF object file will be produced.

## Section 'xxxxxxx' ignored, addresses exceed X'FFFFFF'

### Severity: 8

**Explanation:** The control section named *xxxxxxxx* contains addresses either very close to or exceeding X'FFFFFF'.

**User response:** Reduce the size of the section, change its starting address, or (if the source program contains

multiple control sections) re-assemble with the NOTHREAD option.

**System action:** The item is discarded, and processing continues. No ELF object file will be produced.

## RLD items at 'xxxxxxx' in section with ID 'xxxx' cause double relocation

### Severity: 8

**Explanation:** The RLD items at address *xxxxxxxx* in a control section with ESDID *xxxx* require more than a single relocation at that address.

**User response:** Change the program so that only a single relocation is required at a given address.

**System action:** The items are discarded, and processing continues. No ELF object file will be produced.

### No TXT records in OBJ file

### Severity: 8

**Explanation:** The program generates no machine language instructions or data.

**User response:** Verify that the program is correctly coded. (A DSECT statement may be misplaced.)

**System action:** The item is discarded, and processing continues. No ELF object file will be produced.

### Entry point in zero-length PC section rejected

Severity: 8

**Explanation:** An entry point was specified in a zero-length Private Code section.

**User response:** Verify that the program is correctly coded.

**System action:** The item is discarded, and processing continues. No ELF object file will be produced.

## LD item 'xxxxxxx' in section with ID 'xxxx' lies outside its owning section

### Severity: 8

**Explanation:** An entry point named *xxxxxxx* was specified that lies outside the bounds of its owning control section with ESDID *xxxx*.

**User response:** Verify that the program is correctly coded.

**System action:** The item is discarded, and processing continues. No ELF object file will be produced.

## LD item 'xxxxxxx' in COM section with ID 'xxxx' not supported

### Severity: 8

**Explanation:** An entry point named *xxxxxxx* was specified in a common (COM) control section with ESDID *xxxx*.

**User response:** Correct the program to remove the ENTRY point. (ENTRY points in COM sections are not supported; you may be able to reference items not at the origin of a COM section by specifying a constant offset in a referencing address constant.)

**System action:** The item is discarded, and processing continues. No ELF object file will be produced.

### Invalid or duplicate parm-string character ignored

### Severity: 0

**Explanation:** An invalid or duplicated character appears in the parameter string provided to ASMAXT2E.

**User response:** Correct the options invoking ASMAXT2E to specify valid characters.

**System action:** The item is ignored, and processing continues.

### nnnnnnn SYM records ignored

Severity: 0

**Explanation:** The OBJ file contained *nnnnnnn* SYM records, produced when the TEST option is provided to High Level Assembler.

**User response:** Correct the options invoking High Level Assembler.

**System action:** The records are ignored, and processing continues.

### nnnnnnn non-OBJect record(s) ignored

### Severity: 0

**Explanation:** The OBJ file contained *nnnnnnn* records that are not object-module records.

**User response:** The program may contain PUNCH or REPRO statements. Verify that they are not required.

**System action:** The records are ignored, and processing continues.

### Zero-length Private Code (PC) section ignored

Severity: 0

**Explanation:** The program contains a zero-length Private Code (PC) section.

User response: Correct the source program.

### **ASMAXT2E** messages

**System action:** The section is ignored, and processing continues.

xxxxxxx object records processed, xxxx usable control section(s)

Severity: 0

**Explanation:** The program produced *xxxxxxxx* object

records, and *xxxx* control sections were converted to ELF format.

ELF file length X'xxxxxxx'

Severity: 0

**Explanation:** The generated ELF file is *X'xxxxxxx'* bytes long.

## 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 2455 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 106-0032, 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

### Notices

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.

### AIX

BookMaster Enterprise Systems Architecture/370 Enterprise Systems Architecture/390 IBM MVS MVS/ESA OS/390 System/370 VSE/ESA z/Architecture z/OS z/VM zSeries

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

## **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 and Customization Guide, GC26-8711

# Related Publications (Architecture)

Enterprise Systems Architecture/390 Principles of Operation, SA22-7201

Enterprise Systems Architecture/370 Principles of Operation, SA22-7200

System/370 Principles of Operation, GA22-7000

System/370 Extended Architecture Principles of Operation, SA22-7085

z/Architecture Principles of Operation, SA22-7832

## **Related Publications for z/OS**

### z/OS:

z/OS MVS JCL Reference, SA22-7597

z/OS MVS JCL User's Guide, SA22-7598

z/OS MVS Programming: Assembler Services Guide, SA22-7605

*z/OS MVS Programming: Assembler Services Reference, Volumes 1 and 2,* SA22-7606, SA22-7607

z/OS MVS Programming: Authorized Assembler Services Guide, SA22-7608 z/OS MVS Programming: Authorized Assembler Services Reference, Volumes 1 - 4, SA22-7609 -SA22-7612

z/OS MVS Program Management: User's Guide and Reference, SA22-7643

z/OS MVS System Codes, SA22-7626

z/OS MVS System Commands, SA22-7627

z/OS MVS System Messages, Volumes 1 - 10, SA22-7631 - SA22-7640

z/OS Communications Server: SNA Programming, SC31-8829

### OS/390:

OS/390 MVS System Messages, Volumes 1 - 5, GC28-1784 - GC27-1788

### UNIX System Services:

z/OS UNIX System Services User's Guide, SA22-7801

 $\mathbf{MVS}/\mathbf{DFP}^{\mathsf{TM}}$ :

MVS/DFP V3R3 Utilities, SC26-4559 MVS/DFP V3R3 Linkage Editor and Loader,

SC26-4564

### DFSMS/MVS<sup>®</sup>:

*z/OS DFSMS Program Management,* SC27-1130 *z/OS DFSMSdfp Utilities,* SC26-7414

### TSO/E (z/OS):

z/OS TSO/E Command Reference, SA22-7782

TSO/E (OS/390):

OS/390 TSO/E Command Reference, SC28-1969

### SMP/E (z/OS):

SMP/E Messages, Codes, and Diagnosis, GA22-7770

SMP/E Reference, SA22-7772

SMP/E User's Guide, SA22-7773

### SMP/E (OS/390):

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 z/VM**

*z/VM: VMSES/E Introduction and Reference,* GC24-6130 *z/VM: Service Guide,* GC24-6117 *z/VM: CMS Commands and Utilities Reference,* SC24-6073

*z/VM: CMS File Pool Planning, Administration, and Operation, SC24-6074* 

*z/VM: CP Planning and Administration,* SC24-6083

*z/VM: Saved Segments Planning and Administration,* SC24-6116

*z/VM: Other Components Messages and Codes,* GC24-6120

*z/VM: CMS and REXX/VM Messages and Codes,* GC24-6118

*z/VM: CP System Messages and Codes,* GC24-6119

*z/VM: CMS Application Development Guide,* SC24-6070

*z/VM: CMS Application Development Guide for Assembler,* SC24-6070

z/VM: CMS User's Guide, SC24-6079

z/VM: XEDIT User's Guide, SC24-6132

*z/VM: XEDIT Commands and Macros Reference,* SC24-6132

*z/VM: CP Commands and Utilities Reference,* SC24-6081

## **Related Publications for z/VSE**

*z/VSE: Guide to System Functions*, SC33-8233 *z/VSE: Administration*, SC33-8224 *z/VSE: Installation*, SC33-8222 *z/VSE: Planning*, SC33-8221 *z/VSE: System Control Statements*, SC33-8225 *z/VSE: Messages and Codes*, Vol.1, SC33-8226 *z/VSE: Messages and Codes*, Vol.2, SC33-8227 *z/VSE: Messages and Codes*, Vol.3, SC33-8228 *REXX/VSE Reference*, SC33-6642 *REXX/VSE User's Guide*, SC33-6641

## Index

## Α

ASCII conversion to EBCDIC 5 ASMATXT2E messages 10 assembler options *See* options

## B

books, High Level Assembler vi

## С

CD-ROM publications vii Customization book vi

## D

documentation High Level Assembler 17 documentation, High Level Assembler vi

## Ε

EBCDIC converting generated listing to ASCII 7 copying files using FTP 7 editing files on Linux limitations 7 ELF32 assembler option 3 ELF32 object files creating 5, 9 ELF32 option implementation notes 9 EXIT assembler option 3 external functions 3

## G

General Information book vi generated listing converting from EBCDIC to ASCII 7 GOFF output location 5

## Η

hardcopy publications vi High Level Assembler publications vi

install problems manual extraction 1 installation 1 installation and customization book information vi

## L

Language Reference vi large or complex assemblies increasing SIZE option 7 license inquiry 15 Licensed Program Specifications vi limitations 7 listing output location 5

## Μ

machine instructions, publications 17 manuals High Level Assembler 17 manuals, High Level Assembler vi messages 10

## Ν

NOADATA assembler option3NODECK assembler option3NOGOFF assembler option3NOOBJECT assembler option3

## 0

OBJECT assembler option 3 object output location 5 online publications vii options ELF32 3 EXIT 3 NOADATA 3 NODECK 3 NOGOFF 3 NOOBJECT 3 not available 3 OBJECT 3 SIZE 3 sources 3 specifying 5 organization of this document v

## Ρ

Programmer's Guide vii publications High Level Assembler vi, 17 HLASM Toolkit 17 machine instructions 17 online (CD-ROM) vii z/OS 17 z/VM 17 z/VSE 18

## R

rpm package installing 1 manual extraction 1 querying 1

## S

SIZE assembler option 3 softcopy publications vii space requirements 1 start-up identifiers 5 start-up parameters 5 starting 5 SYSLIB dataset locations 5

## Т

TERM output location 5 Toolkit Customization book vii Toolkit installation and customization book information vii

## U

usage 7 user exits 3

## Ζ

z/OS publications 17 z/VM publications 17 z/VSE publications 18

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

High Level Assembler for Linux on zSeries User's Guide Release 6

### Publication No. SC18-9611-01

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

• Send your comments to the address on the reverse side of this form.

If you would like a response from IBM, please fill in the following information:

Name

Address

Company or Organization

Phone No.

E-mail address



Cut or Fold Along Line



## IBW ®

Program Number: 5696-234

Printed in USA

SC18-9611-01

