

Z80 Cross Development Environment

(XAS-XLD-ITOB)

Operation Manual

The 1st edition

For 2008/11/01 Z80 cross-development-environment Ver.2.2



Seiken-Soft Service Co Ltd

1. Introduction.....	3
1.1 attention	3
1.2 Support System	3
1.3Terms of manuals	4
2. Outline.....	5
2. 1 Z80 Program-Development Schematic Diagram	6
3. Installation.....	6
4. Expiration Date for Use (Trial Period).....	7
5. Starting Method.....	7
6. Uninstallation.....	7
7. Copyright.....	7
8. Reproduction Conditions.....	8
9. Wish of Shareware Registration.....	8
9.1 User's registrations.....	8
10. Operating Procedure.....	9
10.1 Z80 Cross Assembler (XAS)	9
10.2 Z80 Crossing Loader (XLD)-Command Line Specification	10
10.4 Intel Hex Conversion (ITOB)	12
11 Z80 Cross Assembler.....	13
11.1 Pseudo-instructions.....	13
11.2 Details.....	14
11.3 Error Handling.....	27
11.4 The format of listing files.....	28
11.5 Limitations/Special Affairs.....	29
12. Z80 Crossing Loader.....	30
12.1 Format of MAP File.....	30
12.2 Address Solution of Section Absolutely.....	31
12.4 Error messages.....	32
13. Conclusion.....	33

1. Introduction

Thank you very much for using this product. Thank you very much for using this product. Please confirm notes, limitations, and operating instructions for use of this product.

1.1 attention

- (1) We forbid using some or all of this program and a manual without notice, or taking a duplicate.
- (2) When using continuously after passing over a trial period (for after-installation 20 days) ,for shareware, user's registration or a licensing agreement is required for this program. When not registered, it becomes impossible to start.
- (3) We only permits that user's registration or the visitor by whom the licensing agreement was done uses it within the limits of a contract.
- (4) Please understand that we have no responsibility taken about the influence of a result occurred by employing this program and the manual, and use all in a visitor's responsibility.
- (5) WindowsVista/XP/ operates should be used.
- (6) The matter indicated in the specification and the manual of this program may be changed without a preliminary announcement in the future.

1.2 Support System

If you have a trouble nor operating instructions for this product, or thing that you do not understand, please e-mail us from our homepage. The URL is as follows.

<http://www.seiken-soft.com/eng>

Support of this product will be taken as the limitation in Japan. Please understand the situation beforehand. For details, since it has indicated in the attached sheet "transmitting mail guidance .pdf", please refer to it there. Moreover, this description does not explain the following contents. Please confirm by the attachment manual of each product.

- Operating instructions of Windows
- The connection method to the Internet

-A setup and operating instructions of a computer itself and peripheral equipment

1.3Terms of manuals

By this manual, the following terms, a cable address, etc. are used for various kinds of functions or explanation of key operation. Since explanation is described on the basis of WindowsXP may differ from what of description is actual for convenience.

Notation of 1.3.1 mouse operation

By this manual, the following terms, a cable address, etc. are used for various kinds of functions or explanation of key operation. Since explanation is described on the basis of Windows2000, in other versions, operating instructions may differ from a screen. In addition, please understand that the screen of this manual may differ from what of description is actual for convenience.

1.3.1 Notation of mouse operation

- Click

It is the button (usually left button) of a mouse is pushed and the operation detached immediately. In the case of the right button, it is called a right-click.

- Double click

It is the operation which clicks the left button of a mouse twice quickly.

- Drag and drop

It is the operation of that a mouse pointer is moved pushing the left button on a thing to move, and a button is detached in the target position.

1.3.2 Notation of explanation cable address

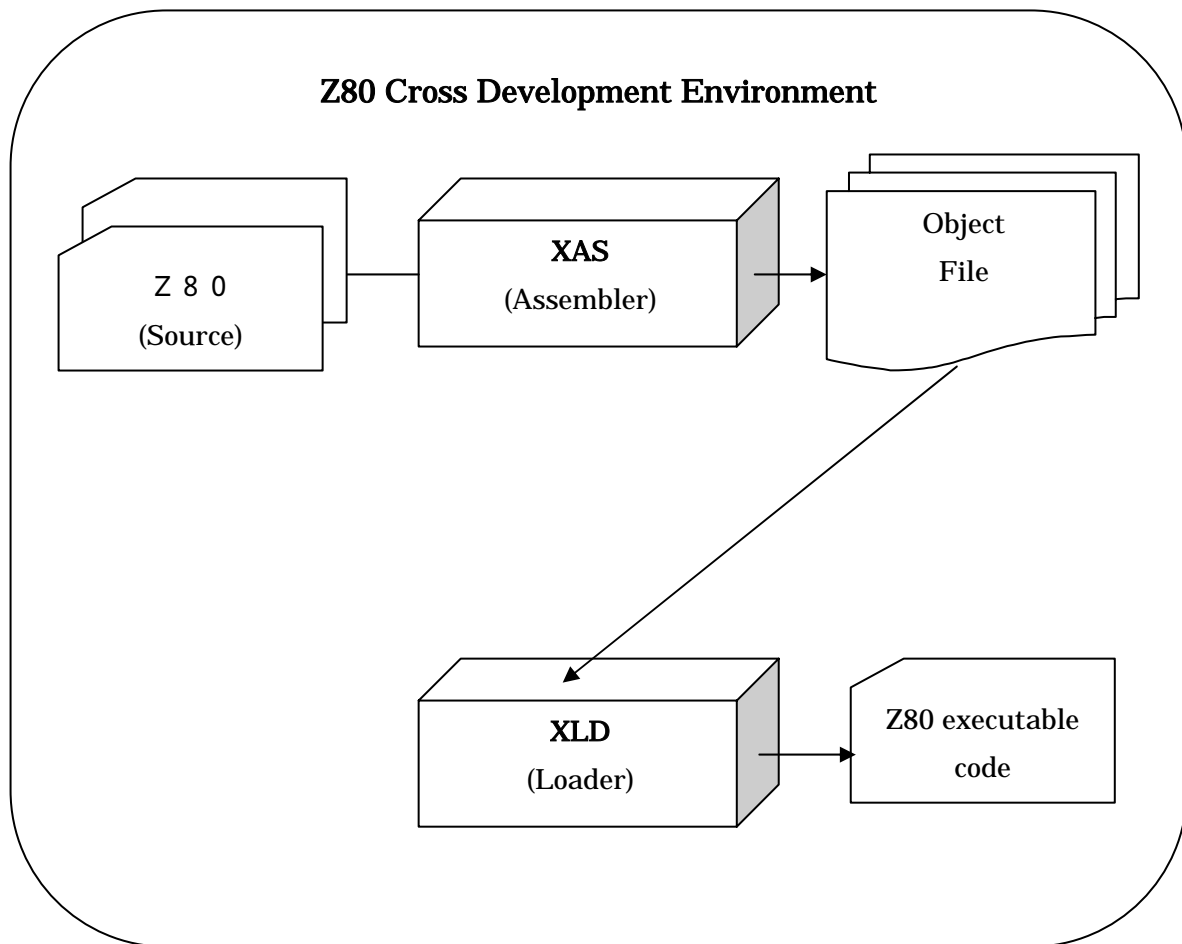
- PC The abbreviation for PC personal computer.
- KSS The abbreviation for KSS Seiken-Soft Service Co Ltd
- XAS XAS Z80 Coss assembler.
- XLD XLD Z80 crossing loader
- ITOB ITOB Intel hexa conversion
- HDD The abbreviation for an HDD hard disk.
- Current folder Point out the folder chosen now [current folder].
- Sub folder Point out the folder of the subordinate of the folder chosen now [subfolder].
- Fullpass Point out the folder of the subordinate of the folder chosen now [subfolder].
- Windows WindowsVista/XP.
 ----Windows is a product of Microsoft Corp.----

2. Outline

Z80 Cross Development is the development environment which can assemble and link Z80 source program and can change it into Intel hexa form or a binary format in the environment where WindowsVista/XP. This program performs by issuing a command from a command prompt. The program of development environment consists of three programs.

- Z80 cross assembler (XAS)
- Z80 crossing loader (XLD)
- Intel hex conversion (ITOB)

2. 1 Z80 Program-Development Schematic Diagram



3. Installation

Since it is distributed in ZIP form and a program should be decompressed by unzip.

An install program is executed by double-clicked after decompressing. Please operate it according to the contents of the display.

* By those who have already installed the old version, those who have registered as a user need to uninstall, after certainly making a note of a user's registration number. Please re-register as a user at the time of installation, or after installation if needed. When installation is normally impossible, again, Windows is restarted, so please rerun setup.exe.

Moreover, the availability of HDD should check enough (not less than 10 M bytes).

-In WindowsVista/XP, manager authority is required.

4. Expiration Date for Use (Trial Period)

This program has an expiration date for use per shareware. User's registration is needed after a trial period of 20 days from an installation date. After the term, you cannot start not this program, so please use after consent.

When you do not use continuously, please be sure to uninstall. Please keep in mind that it means that registry information remained when not uninstalling. Although it does not have influence on operation of a personal computer, it will remain as garbage.

5. Starting Method

Starting of this program is started by issuing a command from a command prompt.

Section 10 explains an operating procedure.

6. Uninstallation

Click [Start] - [program]-[Seiken-Soft] - [Z80Cross Development Environment] - [Uninstallation].

Please perform the rest according to an uninstallation screen.

7. Copyright

This program and the manual are protected by the Copyright Act of Japan.

Seiken-Soft Service Co Ltd holds copyright. Please understand that we have no responsibility taken about the influence of a result occurred by employing this program and the manual, and use all in a visitor's responsibility. When you cannot consent, please be sure to stop use immediately and to uninstall.

8. Reproduction Conditions

Be sure to obtain a report and consent to our company by e-mail. (Send us e-mail from our homepage .)

URL is as follows.

<http://www.seiken-soft.com/eng>

Please carry out transmitting mail from [an inquiry].

9. Wish of Shareware Registration

This program is shareware. When continuing use after passing over 20 days of a trial period, Please pay the following registration fees. If it passes over a trial period, it becomes impossible to start.

- For an individual, it is \$19 (it restricts to one personal computer)
- For a corporation (individual number contract), it is \$19 (per personal computer)
- The department licensing agreement [of \$119 (it restricts in its post and unrestricted)]

* It is the on-line remittance by Creditcard throu PayPal.

Please refer to "Z80 Cross-Development Environment remittance method .pdf" of an attached sheet for the remittance method.

9.1 User's registrations

Since a registration number is sent by e-mail after shareware registration, You can register from [Start] [program]-[Seiken-Soft]-[Z80 cross-development-environment]-[user's registration].

10. Operating Procedure

10.1 Z80 Cross Assembler (XAS)

XAS assembles by considering Z80 assembler source as an input, and creates Z80 crossing relocatable object file.

```
> xas Source file name { Qualifier }
```

-Source file name

The source file name of Z80 is specified.

(The file in which an extension exists should specify an extension at any cost.)

- {Qualifier}

-o	Filename	An object file name is specified. The file to which default added the extension XRB to the source file name (except for an extension) serves as an object file name.
-n		An object is not created.
-l	Filename	The listing file name of assembling is specified. As for default, a listing file is not created.
-x		A cross reference is outputted at the time of an assemble list output. The list output of the definition line (it outputs by # after line watch) of a label and the row number currently used is carried out by a listing file.
-r	Filename	The file name which saves the error information at the time of an assembling error is specified. Default serves as a console output.
-e		A message is outputted in English. Default is outputted in English.
-j		A message is outputted in Japanese.
-h		A help message is displayed.

10.2 Z80 Crossing Loader (XLD)-Command Line Specification

XLD creates Z80 executable code based on the Z80 relocatable object file created by XAS.

The form of executable code outputs by Intel HEX or a binary.

```
> xld Relocatable object file name + -- . {Qualifier}
```

- Relocatable object file name

The file which links is specified.

When it links two or more files, the "+" sign describes.

The extension of each file name is omissible.

- {Qualifier}

-i		It specifies, when outputting executable code by Intel HEXA.
-o	Filename	The output file name of executable code is specified. Default creates by the file name which changed the extension of the relocatable object file name into BIN. (An extension when -i is specified is XOB)
-m	Filename	A mapfile name is specified. Default creates by the file name which changed the extension of the relocatable object file name into MAP.
-b	Address	The start address of a code section is specified. A code section is arranged by this specification.
-d	Address	The address of DSEG is specified.
-e		A message is outputted in English. Default is outputted in English.
-j		A message is outputted in Japanese.
-h		A help message is displayed.

• 10.3 Z80 Crossing Loader (XLD)-Response File Specification

Z80 executable code is created like what was explained by section 10.2.

However, it is the form that specification of an object file can be described and specified in a response file in the case of this book type.

```
C:\Z80> xld -r Response file name
```

- {Qualifier}

-r	Filename	The response file name by which link information is defined is specified.
----	----------	---

The form of a response file is shown below.

```
OUT= Output File Name
OBJ= Object File Name :
MAP= mapfile name
BASE= Base Address
DSEG=DSEG address
```

The keyword in a response file is shown below.

The output file name of an OUT= Executed file name link result is specified.

The object file name which performs an OBJ= object file name link is specified.

-A MAP= map file name

map output file name is specified.

-A BASE= base address

base address is specified by the hexadecimal notation.

-A DSEG=DSEG address

DSEG address is specified by the hexadecimal notation.

-Example: The example in the case of linking A.XRB and B.XRB and creating executable code file C.XOB

```

OBJ=A
OBJ=B
OUT=C
MAP=MAP.LST
BASE=2000
DSEG=5100

```

10.4 Intel Hex Conversion (ITOB)

The Intel Hex format of Z80 executable code is changed into a binary format.

```
C: Z80> itob The Intel Hexa file name {Qualifier}
```

- Intel HEX file name

The Intel Hex file name outputted by Z80 crossing loader is specified.

- {Qualifier}

-o	Filename	An output file name is specified. Default outputs the extension of the Intel hex file name to the file changed into BIN.
-a	Address	An output start address is specified. Default is outputted from the minimum effective address of an object.
-e		A message is outputted in English. Default is outputted in English.
-j		A message is outputted in Japanese.
-h		A help message is displayed.

11 Z80 Cross Assembler

A pseudo-instruction, error handling, etc. which can be assembled by Z80 cross assembler (XAS) are explained.

11.1 Pseudo-instructions

The following pseudo-instruction is supported in XAS.

INCLUDE	INCLUD	\$INCLUDE
MACLIB		
MACRO	EXITM	ENDM
CSEG	DSEG	
ASECT	AORG	ORG
ENTRY	GLOBAL	PUBLIC
INTERN		
EXT	EXTERN	EXTERNAL
EXTRN		
DB	DW	DS
DEFB	DEFW	DEFS
DC	DEFM	
RADIX	END	
EQU	=	ASET
REPT	ENDR	
.LIST	.XLIST	.PRINTX
IF	IFDEF	IFNDEF
ELSE	ENDIF	
PAGE	TITLE	

11.2 Details

(1) INCLUDE, INCLUD, \$INCLUDE, MACLIB

It directs to include other source files.

The nest of include is possible up to eight levels.

INCLUDE	Include file name
INCLUD	Include file name
\$INCLUDE	Include file name
MACLIB	Include file name

EX.

```
INCLUDE  BIOSCALL.H
```

(2) MACRO, EXITM, ENDM

The start of a macroscopic description is directed. It is processed as broad view same to ENDM.

Macroscopic use within macro is possible up to eight levels.

Moreover, when interrupting macroscopic execution, it skips to ENDM by EXITM command.

Macro name	MACRO	Parameter [, parameter ...]
	EXITM	
	ENDM	

EX.

```

                SYS    MACRO    BIOS_NO
                RST    30H
                DB     BIOS_NO
                ENDM

```

(3) ASECT

It directs that it is a section absolutely.

ASECT	ROM; ROM area
ASECT	RAM; RAM area

EX.

```

                ASECT    ROM

```

(4) AORG

The address of a section is directed absolutely.

It is assembled in the address directed by this pseudo-instruction.

It is necessary to describe an ASECT pseudo-instruction before this pseudo-instruction.

AORG	Address
------	---------

EX.

ASECT	ROM
AORG	0H
JP	INITIAL
AORG	8H
JP	WORM

(5) ORG

An address is specified.

It is assembled in the address directed by this pseudo-instruction.

ORG	Address
-----	---------

E X .

EX.

ORG	0H
JP	INITIAL

(6) CSEG

It directs that CSEG or subsequent ones is a code segment. Usually, it acts as Assen Bull as a code segment.

CSEG

EX.

CSEG

(7) DSEG

It directs that DSEG or subsequent ones is a data segment.
Moreover, please do not specify except DS command after DSEG.

DSEG

EX.

DSEG

(8) ENTRY, GLOBAL, PUBLIC, INTERN

It is shown that the specified label is referred to from other modules.

ENTRY	Label [, label]
GLOBAL	Label [, label]
PUBLIC	Label [, label]
INTERN	Label [, label]

EX.

ENTRY LABEL

(9) EXT, EXTERN, EXTERNAL, EXTRN

It is shown that the specified label is defined as other modules.

EXTERN	Label [, label]
EXT	Label [, label]
EXTERNAL	Label [, label]
EXTRN	Label [, label]

EX.

EXTERN LABEL

(10) DB, DEFB

The specified data is outputted to an executable file.

DB	Data [, data ...]
----	-------------------

EX.

DB	10	- Decimal
DB	0FFH	-Hexadecimal notation
DB	01001001B	- Binary
DB	'XAS'	- Character string
DB	F'3.14'	- BCD float

(11) DC, DEFM

The specified character string is outputted to an executable file.

DC	Data [, data ...]
DEFM	Data [, data ...]

EX.

DC	'XAS'	-Character string
DEFM	'XAS1'	-Character string

(12) DW, DEFW

2 bytes of specified data is outputted to an executable file.

DW	Data [, data ...]
DEFW	Data [, data ...]

EX.

DW	10	- Decimal
DW	0FFFFH	- Hexadecimal notation
DW	LABEL	-Label specification

(13) DS, DEFS

The area of designated size is secured.

DS	Byte size
DEFS	Byte size

EX.

DS	10
----	----

(14) RADIX

Change of the fiducial value after this command is directed.

At the time of XAS starting, a decimal is a fiducial value.

RADIX	10; let a fiducial value be a decimal.
RADIX	16; a fiducial value is made into the hexadecimal notation.

(15) END

The end of a source file is directed.

Assembling is not performed even if the command is described after the END command.

END

(16) EQU, =

The value of a formula is assigned to a label.

Label	EQU	Formula
Label	=	Formula

EX.

L1	EQU	110
L2	=	L1 + 5

(17) ASET

A value is redefined on the label which EQU or = command defined.

Label	ASET	Formula
-------	------	---------

EX.

L1	EQU	10
L1	ASET	L1 + 5

(18) REPT, ENDR

The number-of-times part deployment specified by the formula in the sauce to ENDR is repeated.

REPT	Formula
ENDR	

EX.

```
REPT    5
DB      00H
ENDR
```

(19) XIST, .XLIST

A start (default) or an output is forbidden for the output to a print file.

.LIST	Output start
.XLIST	Output inhibition

EX.

```
.XLIST
INCLUDE BIOSCALL.H
.LIST
```

(20) PRINTX

The character string or value specified with the parameter is displayed on a console.
In the case of a character string, it specifies in the form of a 'character string'.

.PRINTX Parameter [, parameter ...]

EX.

```
                .PRINTX   'It is [ XXX INCLUDE ] under
                        processing.'
L1      EQU      1
                .PRINTX   L1
L1      ASET     L1+20
                .PRINTX   L1
```

(21) IF, ELSE, ENDIF

When the result specified by the formula is truth (a value, except 0), even ELSE (ENDIF when not ELSE specifying) is assembled.

Moreover, "<", ">", and "=" can be used as a comparison operator.

IF	Formula
ELSE	
ENDIF	

EX. In the case of the following example, from ELSE to ENDIF is assembled.

L1	EQU	10
L2	EQU	20
	IF	L1 > L2
		L1 > L2
	LD	A,1
	ELSE	
	LD	A,2
	ENDIF	

(22) IFDEF, IFNDEF, ENDIF

The specified label performs the judgment of a definition or the undefined, and, in a true case, assembles even ELSE (in the ELSE case of not specifying ENDIF).

IFDEF	Label
	Een ELSE is assembled when the label is defined.
ELSE	
ENDIF	
IFNDEF	Label
	When a label is an undefined, from ELSE to ENDIF is assembled.
ELSE	
ENDIF	

EX. In the case of the following example, from ELSE to ENDIF is assembled.

L1	EQU	10
	IFNDEF	L1
	LD	A,1
	ELSE	
	LD	A,2
	ENDIF	

(23) PAGE

A form feed is performed at the time of an assemble list output.

PAGE

(24) TITLE

The title at the time of assembling is specified. At the time of assembling, it displays on a console.

TITLE Title

EX.

TITLE Z80

11.3 Error Handling

In XAS, source is read with the path 1, it changes into the machine language of Z80, and conversion of unsolved data is performed after reading all the source.

For this reason, when the error which comes out, respectively occurs, it outputs to an error file (error file default is a console).

(1) An error with the path 1

When it becomes an error by assembling, the line used as an error is outputted.

File name Row number ET The contents of a source line

E shows an error sign and the contents are as follows.

- S A command does not exist.
- O The error of an operand

The label of U undefined was specified.

- A The error of addressing
- L Label error
- M The duplication error of a label
- I The error by IF command
- R The error by REPT command

T shows row attributes and the contents are as follows.

- Space It is the usual source line.
- It is the macro expansion within an include file.
-
- It is a source line of an include file.
- + It is macro expansion.

(2) The error by unsolved data conversion processing

At the time of the error generation in a conversion process, it outputs in the form of the following.

- File name (row number) Address Error

In JR command etc., the address became the outside of the range.

- File name (row number) Syntax Error

Description has an error or a label is an undefined.

11.4 The format of listing files

When a listing file is specified by a command, it outputs to a file in the form of the following.

		1	INCLUDE	A.H
0064		1-	LABEL EQU	100
0000	54455354	2	DB	'TEST'
0004	1A	3	DB	'Z'-40H
0005	000000	4	DB	0,0,0
0008	45584530	5	DB	'EXEC01'
000D	000000	6	DB	0,0,0
0010	310046	7	LD	SP,4600H
0013*	1800	8	JR	

The contents of source

The attribute of a line (refer to error handling)

Data of an object

[*] is outputted when there is unsolved data with the path 1.

An address or an EQU value

Error sign (refer to error handling)

11.5 Limitations/Special Affairs

(1) Pseudo-instruction

The limitations in a pseudo-instruction are shown below.

- REPT can nest and don't become precocious.

(2) The number of characters of one line

You have to describe one line within 128 characters.

(3) The number of operands in one line

As for the number of operands of one line, a label, a command, and the sum total of an operand, you have to describe. less than 31.

(4) Formula

The following operators can be used for a formula.

() It can be used even if it attaches.

+ Addition

- Subtraction

* Multiplication

/ Division

MOD Surplus

& Logical product

| Logical sum

^ Exclusive OR

~ 1's complement

EX. L1 EQU 10

L2 EQU 5

L3 EQU L1 + L2

L4 EQU L1 - L2

L5 EQU (L1 + 20) * L4

L6 EQU (L1 + L2) / 2

L7 EQU L1 MOD L2

L8 EQU L1 & L2

(5) The symbolic convention of BCD float

The BCD float symbolic convention in DB pseudo-instruction is shown below.

DB F'{+/-} exponent part and fixed point part'

In +, the sign of exponent part is omissible.

Moreover, it is omissible also about a fixed point part.

The symbolic convention of a fixed point part may can come out and, in +, can omit about a sign by E {+/-} mantissa.

Example DB F'-3.14E-5'

12. Z80 Crossing Loader

12.1 Format of MAP File

When specification of a MAP file is performed by the XLD command, the information on a link result is outputted in the form of the following.

(1) Address arrangement information on each module

Module Linkage Map

Addr Size Module

0000 0277 MAIN

(2) Arrangement information of GLOBAL or ENTRY declaration

Global Symbol Map

Addr Symbol

0277 MAIN

(3) Data segment arrangement information

Dseg Symbol Map

Module Addr Size Symbol

MAIN 4310 0001 MAIN

12.2 Address Solution of Section Absolutely.

When ASECT specification is performed by the source file, address solution (arrangement) of the object is carried out with an AORG value.

It becomes an error when ASECT specification is carried out by two or more objects.

Moreover, a command line or a response file must describe the object of ASECT specification at the head (specification).

• 12.3 Address Solution of Data Segment

In XLD, address solution of the data segment is carried out from 8000H street.

However, when a DSEG address is specified by /DSEG, address solution is performed from a specification address. An order of address solution is shown below.

- (1) Arrange first that by which GLOBAL(ing) or an ENTRY declaration was made from 8000H street.
(It is a specification address at the time of /DSEG specification)
- (2) Next, arrange that by which GLOBAL(ing) or an ENTRY declaration is not made after the address arranged by (1).

12.4 Error messages

The error message at the time of XLD command execution is explained.

- Command Line Error

Specification of a command line has an error.

- Switch Error

Switch specification of a command line has an error.

- Responce File Syntax Error

A response file has an error.

- Memory Alloc Error

A memory was not able to be secured.

- External Symbol Undefined

The external symbol is not defined.

- Convert Error

The error occurred in address solution etc.

- Not XRB Format File

It was specified except the object file.

- Not XLB Format File

It was specified except the library file.

- Object Size Over

The size of the object exceeded 64 KB.

- Asect Addr Error

It already finished. [absolutely / the address of a section / a definition]

Absolutely section by two or more modules cannot be specified.

13. Conclusion

We will keep on making effort to develop convenient and useful products. We are pleased if you use our products forever in the future.

Development team all the persons concerned

<p>Selling agency: 1-6-7, Kita-Kojiya, Ota-ku, Tokyo, 144-0032 Seiken-Soft Service Co Ltd http://www.seiken-soft.com/eng</p>
