USER'S MANUAL

NEC

DOC17K

DOCUMENT PROCESSOR

(PRELIMINARY)

Document No. EEU-1536
(O. D. No. EEU-5006)
Date Published May 1995 P
Printed in Japan

Preface

Product Outline:

The Document Processor (DOC17K) extracts program information from 17K Series relocatable assembler (RA17K) or C compiler (*emIC-17K*™) source programs and documents the information in a specified format.

Audience:

This manual is aimed at those engineers who use RA17K or *emIC-17K* to develop application systems based on the 17K Series 4-bit single-chip microcontroller.

Organization:

This manual consists of the following chapters:

General

Running the Document Processor

Types of Output Listings and Options

Output Listings

Error Messages

Reading This Manual:

The reader is assumed to be familiar with the concepts of RA17K or *emIC-17K*.

Conventions:

In this manual, the following symbols are used:

...	:	Indicates repetition.
[ ]	:	Elements in brackets are optional.
{|}	:	One of the characters or character strings, delimited by "|" in braces, must be selected.
< >	:	The actual characters enclosed in the angular brackets are to be specified; often used to indicate a title.
CR	:	Carriage return.
LF	:	Line feed.
TAB	:	Horizontal tab.

xxx : Any character string.

Single-byte : Single-byte character.

Double-byte(Note) : Double-byte character.

(Note) Only the PC-9800 version supports double-byte characters.

File Naming Conventions:

[drive-name:][\directory-name\…]filename[.extension]

A filename includes a drive name and a directory name. A drive name and directory name together constitute a pathname.

Related Documents:

Note that these related documents may be preliminary editions. These documents are not marked ”Preliminary” in this manual.

- **Documents related to software tools**

  Project Manager Version 1.00 User’s Manual : To be released soon

  RA17K User’s Manual : To be released soon

  RA17K Utility User’s Manual : To be released soon

  LK17K User’s Manual : To be released soon

  emIC-17K Version 1.00 User’s Manual : EEU-1427

  DOC17K User’s Manual : This manual

- **Documents related to *SIMPLEHOST*™**

  SIMPLEHOST Version 2.0 corresponding to *emIC-17K*/RA17K

  User’s Manual, Tutorial: To be released soon

  SIMPLEHOST Version 2.0 corresponding to *emIC-17K*/RA17K

  User’s Manual, Reference: To be released soon

The software package may contain a README.DOC file. The README.DOC file provides information that was not available when the manual was printed. Always read the contents of the README.DOC file before attempting to use the product.

Contents

Chapter 1 General 1-1

1.1 Configuration 1-1

1.2 Outline of Functions 1-2

1.3 System Configuration 1-3

1.3.1 Files That Are Referenced or Created 1-4

1.4 Operating Environment 1-5

1.5 Environmental Variable 1-5

1.6 Limitations 1-6

Chapter 2 Running the Document Processor 2-1

2.1 Execution 2-1

2.1.1 Execution Procedure 2-1
2.1.2 Sample Runs 2-3
2.1.3 Sample Runs without Parameters 2-5

2.2 Start and End Messages 2-16

2.3 Cancellation (CTRL + C) 2-17

Chapter 3 Supported Output Listings and Options 3-11

3.1 Options 3-1

3.1.1 Options That Specify a Listing Utility 3-2
3.1.2 Options Common to All Listing Utilities 3-4
3.1.3 Listing-Specific Options 3-6

Chapter 4 Output Listing 4-1

4.1 Option Information Listing 4-1

4.2 Absolute Address Format Listing 4-3

4.3 Cross-Reference Listing 4-7

4.3.1 Embedded Cross-References 4-7
4.3.2 Table Cross-References 4-9

4.4 Module Map and Program Map Listings 4-13

4.4.1 Map Listings 4-13
4.4.2 Symbol Listing 4-18

4.5 Report Listing 4-20

4.6 Document Listing 4-24

4.6.1 Document Listing 4-24
4.6.2 Summary Rules 4-28
4.6.3 Format Definition 4-33

4.7 Call Tree Listing 4-39

4.8 Module Listing 4-40

4.9 Address Map 4-42

Chapter 5 Error Messages 5-1

5.1 Error Handling 5-1

5.2 Error Messages 5-1

5.3 System Error Messages 5-6

List of Figures

Figure 1-1. DOC17K Functions 1-1
Figure 1-2. DOC17K System Configuration 1-3

Figure 3-1. Report Listing Output Format (Default) 3-13
Figure 3-2. Document Listing Output Format (Default) 3-14
Figure 3-3. Module Listing Output Format (Default) 3-16
Figure 3-4. Address Map Output Format 3-16

Figure 4-1. Sample Memory Map Listing 4-14
Figure 4-2. Representation of a Data Memory Map Greater than One 4-15
Figure 4-3. Summary Layout 4-29

[MEMO]

List of Tables

Table 2-1. Listing Utilities 2-1

Table 3-1. Types of Output Listings and Options 3-1
Table 3-2. Options That Specify a Listing Utility 3-2
Table 3-3. Input Files Required to Enable Listing Output 3-3
Table 3-4. Input File Extensions 3-4
Table 3-5. Options That Are Common to All Listing Utilities 3-4
Table 3-6. -TAB Option Settings 3-5
Table 3-7. -FORM Option Settings 3-5
Table 3-8. -HEAD Option Settings 3-6
Table 3-9. Absolute Address Format Listing Options 3-6
Table 3-10. -GEN Option Settings 3-7
Table 3-11. -COND Option Settings 3-7
Table 3-12. -SEQ Option Settings 3-7
Table 3-13. -BRANCH Option Settings 3-8
Table 3-14. Module Map Listing and Program Map Listing Options 3-8
Table 3-15. -AXIS Option Settings 3-9
Table 3-16. -LINEKIND Option Settings 3-9
Table 3-17. -LINEKIND Option Settings (Detailed Settings) 3-10
Table 3-18. -SEPARATE Option Settings 3-10
Table 3-19. -SORT Option Settings 3-11
Table 3-20. Cross-Reference Listing Option 3-12
Table 3-21. -XFORM Option Settings 3-12
Table 3-22. Report Listing Option 3-12
Table 3-23. Document Listing Options 3-13
Table 3-24. Call Tree Listing Option 3-15
Table 3-25. Module Listing Options 3-15

Table 4-1. Relationship Between Components of Contents and Program Units 4-25
Table 4-2. Reserved Item Names in the Format Definition File 4-34

Table 5-1. Error Messages 5-2

[MEMO]

Chapter 1

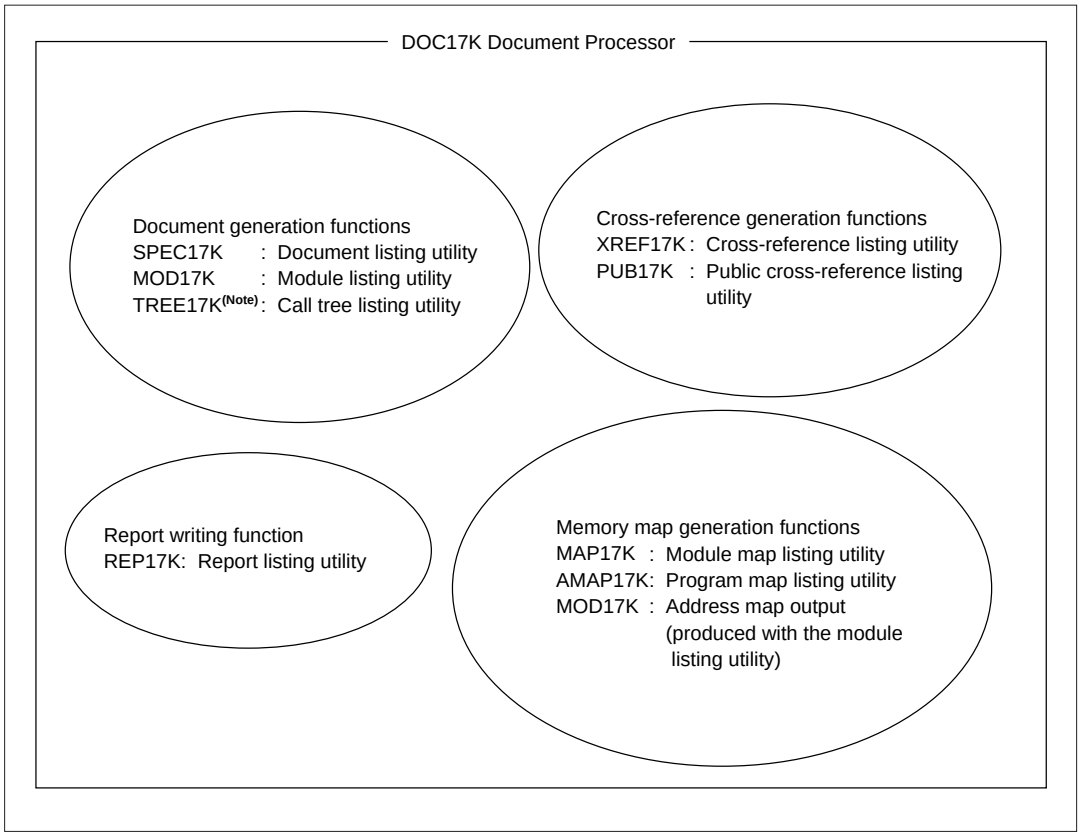
General

The Document Processor (DOC17K) extracts program information from relocatable assembler (RA17K) or C compiler *(emIC-17K)* source programs and documents the information in a specified format.

1.1 Configuration

The multi-functional Document Processor consists of several sets of utilities and programs that manage these utilities.

Figure 1-1. DOC17K Functions



(Note) Available only when *emIC-17K* is used.

1.2 Outline of Functions

The Document Processor provides the following functions:

(1) Document generation

The Document Processor produces the following document components:

(a) Contents

The titles assigned to all blocks are output as a table of contents.

(b) Program specifications

Module (including routine) information in a summary format.

(c) Module listing

A listing of each module file and its routines.

(d) Module configuration

A call tree listing (supported only when *emIC-17K* is being used).

(2) Cross-reference (symbol cross-reference) generation

The Document Processor can produce two kinds of listing:

module-specific cross-reference listings and program public symbol cross-reference listings.

(3) Memory map generation

(a) Memory map/flag map

The Document Processor can produce two kinds of maps:

module-specific memory maps and program memory maps.

(b) Symbol list

(4) Report writing function

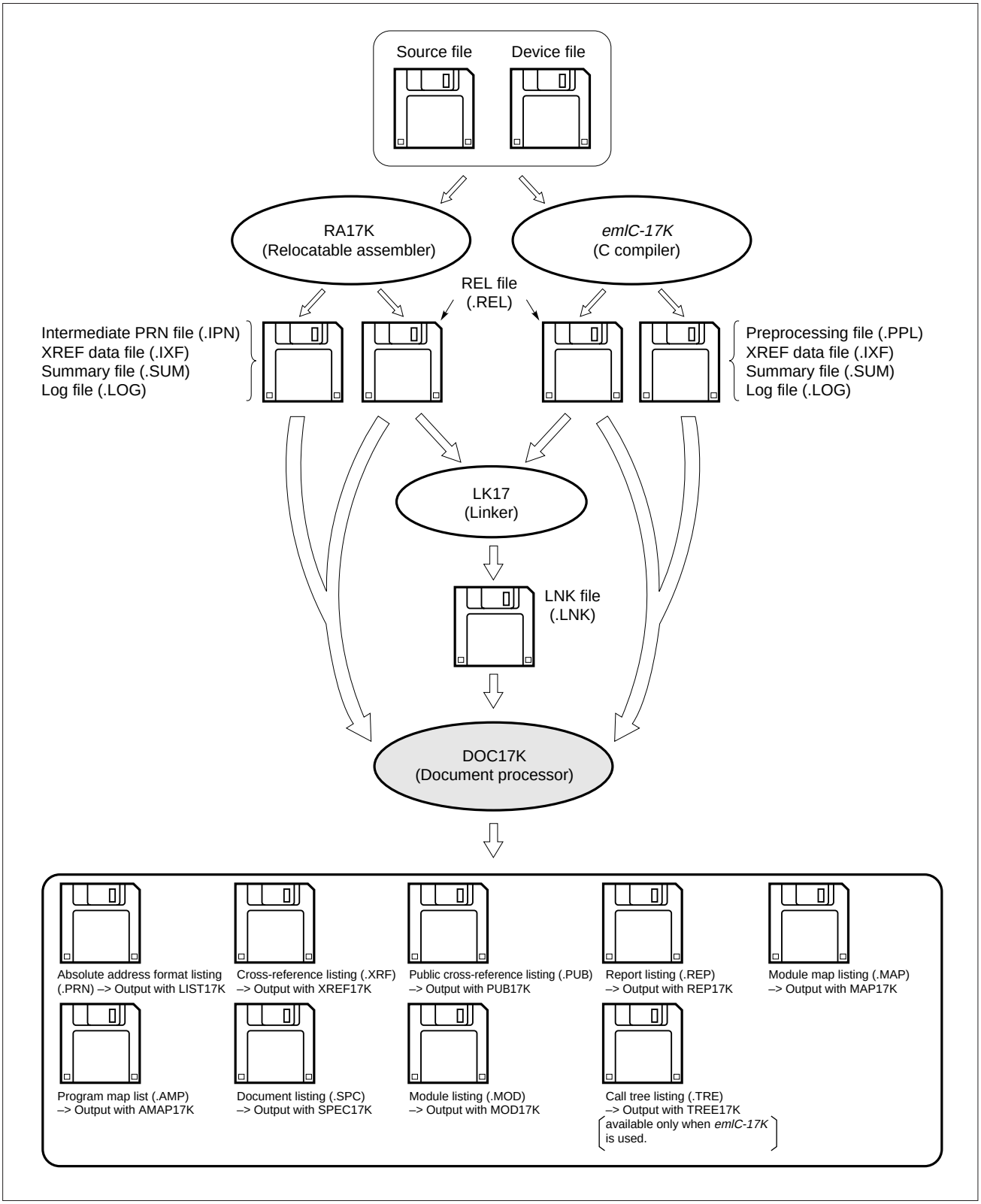
(a) Listing file reformatting

(b) Report (log file) reformatting

1.3 System Configuration

Figure 1-2 shows the configuration of the DOC17K system.

Figure 1-2. DOC17K System Configuration



1.3.1 Files That Are Referenced or Created

(1) Files that are referenced

(a) REL file (.REL)

A file of relocatable object codes and assembly (or compilation) information, produced by RA17K or *emlC-17K*.

A LNK file (.LNK) must be specified at the same time, by specifying the -LNK option.

(b) Log file (.LOG)

A file recording the messages output by RA17K or *emlC-17K* upon assembly or compilation.

(c) Intermediate PRN file (.IPN)

An assembly result list file produced by RA17K.

(d) Preprocessing file (.PPL)

A precompilation result list file produced by *emlC-17K*.

(e) Cross-reference data file (.IXF)

An information file used to produce a symbol cross-reference listing.

(f) Summary file (.SUM)

A file of information defined by a SUMMARY pseudo instruction.

(2) Files that are created

(a) Absolute address format listing (.PRN)

An edited listing, represented in absolute address format, of the contents of an intermediate PRN file, along with the address and object codes stored in it.

(b) Cross-reference listing (.XRF)

An edited listing of cross-reference information.

(c) Public cross-reference listing (.PUB)

An edited listing of cross-references between the public symbols of a program.

(d) Module map listing (.MAP)

A module-specific data memory map, presented as a table.

(e) Program map listing (.AMP)

A program data memory map, presented as a table.

(f) Report listing (.REP)

Run-time information, produced by RA17K or *emIC-17K*, presented as a listing.

(g) Document listing (.SPC)

Summary information, and information derived from an intermediate PRN file, presented as a document.

(h) Call tree listing (.TRE)

A tree listing of all the functions defined and referenced by *emIC-17K*.

(i) Module listing (.MOD)

A module file-specific routine listing, which may contain a program address map if needed.

1.4 Operating Environment

DOC17K requires the operating environment described below:

• Host machine	:	NEC PC-9800 Series
		IBM PC/AT™
• OS	:	MS-DOS™ Ver.3.30 or higher
		PC DOS™ Ver.5.02 or higher
• Main memory	:	640K bytes or more. At least 400K bytes of free memory is required.
• Extended memory (EMS)	:	At least 1M byte of extended memory is recommended.
• External storage	:	At least 1M byte. (The use of a hard disk is recommended.)

1.5 Environmental Variable

The following environmental variable is used with DOC17K:

TMP: Specifies a temporary file search path. If more than one path is specified, only that specified first becomes valid.

Paths are used in the following order:

<1> If a work path option (-WORK) is specified, temporary files are created on that path.

<2> If environmental variable TMP is specified, temporary files are created on that path.

<3> If neither **<1>** nor **<2>** is specified, temporary files are created on the current path.

1.6 Limitations

The limitations imposed on DOC17K are summarized below:

(1) Utilities other than MAP17K support up to 8,000 symbols. MAP17K supports up to 6,000 symbols for each data memory bank of the device (17K Series). If EMS memory is available, a symbol area is allocated in EMS memory; otherwise, it is allocated in main memory. When EMS memory is not available, the number of symbols supported is limited by the amount of free main memory.

(2) During list processing, a work area is allocated as needed. The size of the allocated work area depends on the number of symbols to be processed. The size of the work area that can be allocated is limited by the amount of free main memory in the host.

Chapter 2

Running the Document Processor

2.1 Execution

The Document Processor runs the individual listing utilities for which options have been specified. Run the Document Processor (DOC17K) to produce all the listings at one time.

The listing utilities may also be run independently.

2.1.1 Execution Procedure

(1) Command

- **Running DOC17K**

```
>DOC17K[.EXE]<module-name>-LNK=<link-filename>[-<option>...]
```

- **Running listing utilities independently**

```
>xxx17K[.EXE]<module-name>-LNK=<link-filename>[-<option>...]
```

Table 2-1 lists the listing utilities and indicates the order in which they are excuted by DOC17K.

Table 2-1. Listing Utilities

Name	Description	Execution order
LIST17K	Absolute address format listing utility	1
XREF17K	Cross-reference listing utility	2
PUB17K	Public cross-reference listing utility	3
MAP17K	Module map listing utility	4
AMAP17K	Program map listing utility	5
REP17K	Report listing utility	6
SPEC17K	Document listing utility	7
MOD17K	Module listing utility	8
TREE17K	Call tree listing utility	9

(a) module-name

Specify the REL filename(Note) (.REL) of a module to be output.

To name more than one module, specify a parameter file containing a list of modules.

Code a parameter file in '@filename' format, along with any extension.

Each module name appearing in the parameter file must also be a REL filename.

(Note) For TREE17K, specify the filename of a preprocessing file (.PPL) produced by *emlC-17K*, not a REL filename.

(b) link-filename

Specify the filename of the link object file (.LNK) containing the module to which information is to be directed.

(c) option

Specify listing options as necessary. Which options are valid varies with the listings.

If more than one module name is specified, the option(s) specified here applies to all the specified modules.

To produce the same listing but with different options, execute a listing utility for each module involved.

Options may also be specified in the parameter file. Any options specified in the parameter file are overridden, however, by options specified from the command line. Options can be altered, therefore, simply by specifying the initial options, without having to modify the parameter file.

(2) Parameter file format

Options and a REL filename can be written in the parameter file.

The options must precede the REL filename. Otherwise, an error will occur.

When the same option is coded more than once, that coded last becomes effective.

A character string beginning with semicolon (;) is handled as a comment.

Example

```
; options
-LNK=SAMPLE.LNK -LIST -COL=128   ;  listing
-LNK=SAMPLE.LNK -MAP -AXIS=R     ;  map listing
; REL filename
SAMPLE1.REL
SAMPLE2.REL
```

(3) Listing output destination specification

There are two ways of specifying the destination to which listings are to be directed.

When more than one module is named, if a filename is not specified as a listing option, module-specific listings are produced. Therefore, always specify an output filename when lists are to be directed to one file.

(i) When the filename is omitted

If a single module is named, an output file having the REL filename as its filename is produced.

If more than one module is named in the parameter file, an output file is produced for each named module if the filename following the options is omitted. Program map and public cross-reference listings, however, are directed to the single file named ”module-name.extension.” If a single module is named, or to a single file named ”parameter-file.extension” if more than one module is named.

(ii) When the filename is specified

When ”=filename” is specified after the listing options, all output listings are directed to the named file, regardless of whether a single module or more than one module has been named.

2.1.2 Sample Runs

(1) DOC17K sample runs

(a) Single-module specification

```
>DOC17K SAMPLE1.REL -LNK=SAMPLE1.LNK -LIST -MAP -XREF
```

SAMPLE1.PRN, SAMPLE1.MAP, and SAMPLE1.XRF are produced.

(b) Multi-module specification

Assume that a parameter file, named SAMPLE.INP, contains a list of module names (SAMPLE1.REL, SAMPLE2.REL).

```
>DOC17K @SAMPLE.INP -LNK=SAMPLE.LNK -LIST - MAP -XREF
```

Listings are produced for each module specified by SAMPLE.INP.

SAMPLE1.PRN, SAMPLE1.MAP, SAMPLE1.XRF, SAMPLE2.PRN, SAMPLE2.MAP, and SAMPLE2.XRF are produced.

(2) Listing utility sample runs

(a) Single-module specification

```
>LIST17K SAMPLE1.REL -LNK=SAMPLE1.REL -COL=128
```

SAMPLE1.PRN is produced in 128 positions per line.

(b) Multi-module specification and module-specific output

Assume that a parameter file, SAMPLE.INP, contains a list of module names (SAMPLE1.REL, SAMPLE2.REL) and a link filename (-LNK=SAMPLE.LNK).

```
>LIST17K @SAMPLE.INP
```

An absolute address format listing is produced for each module specified by SAMPLE.INP.

SAMPLE1.PRN and SAMPLE2.PRN are produced.

(c) Multi-module specification and single file output

Assume that a parameter file, named SAMPLE.INP, contains a list of module names (SAMPLE1.REL, SAMPLE2.REL).

```
>MAP17K @SMAPLE.INP -LNK=SAMPLE.LNK -MAP=SAMPLE.MAP
```

A module map listing for each module specified by SAMPLE.INP is output to SAMPLE.MAP.

Caution LIST17K does not support output file specification.

2.1.3 Sample Runs without Parameters

If no parameters are specified (only the EXE filename is entered), control returns to the DOS prompt and the following HELP messages are displayed:

(1) DOC17K

```
17K Series Document Processor Vx. xx [DD MM YY]
     Copyright (c) NEC Corporation 1994

usage1   :  DOC17K input-file [option[ ...]]
usage2   :  DOC17K @input-file [option[ ...]]

input_file   :  Specify object file name.
@input_file:  Specify file name of object file name list.

The option is as follows ([ ] means omissible).

    -LNK=file                          :Specify linkage file name.
    -LIS[T]/-NOL[IST]                  :Create assemble list / Not.
    -XRE[F][=file]/-NOX[REF]           :Create cross referrence list
                                            [with specified name] / Not.
    -PUB[XREF][=file]/-NOP[UBXREF]     :Create public cross reference list
                                            [with specified name] / Not.
    -MAP[=file]/-NOM[AP]               :Create module memory map list
                                            [with specified name] / Not.
    -AMA[P][=file]/-NOA[MAP]           :Create program memory map list
                                            [with specified name] / Not.
    -REP[ORT][=file]/-NOR[EPORT]       :Create report list
                                            [with specified name] / Not.
    -SPE[C][=file]/-NOSP[EC]           :Create document list
                                            [with specified name] / Not.
    -MOD[ULE][=file]/-NOMO[DULE]       :Create module list
                                            [with specified name] / Not.
    -TRE[E][=file]/-NOTR[EE]           :Create tree list
                                            [with specified name] / Not.
    -ROW=n                             :Specify row in a page.
    -COL[UMN]=n                        :Specify column in a line.
    -TAB/-NOT[AB][=n]                  :Specify tab code or space.
    -FOR[M]/-NOF[ORM]                  :Specify form feed code or line feed code.
    -HEA[D]/-NOHE[AD]                  :Specify printing header or not.
    -WOR[K]=path_name                  :Specify path name for temporary file.
    -GEN/-NOG[EN]                      :Specify printing generate lines or not.
    -CON[D]/-NOC[OND]                  :Specify printing condition lines or not.
    -SEQ/-NOS[EQ]                      :Specify printing options or not.
    -BRA[NCH]/-NOB[RANCH]              :Specify printing label reffer or not.
    -AXI[S]={C|R}                      :Specify axis column or row.
```

-NAM[ELENGTH]=n	:Specify name length.
-LIN[EKIND]={H|Z}	:Specify line kind.
-SEP[ARATE]={PERIOD|POINT}	:Specify separate line period or point.
-DEL[IMITER]=character	:Specify delimiter character.
-NRO[W]=n	:Specify lines of name.
-NV=character	:Specify nibblev character.
-NB=character	:Specify nibble character.
-STR=character	:Specify structure/union character.
-SOR[T]=n	:Specify sorting symbol.
-SYM[LENGTH]=n	:Specify symbol name length.
-LU=character	:Specify left up corner character.
-LD=character	:Specify left down corner character.
-RU=character	:Specify right up corner character.
-RD=character	:Specify right down corner character.
-UC=character	:Specify up cross character.
-DC=character	:Specify down cross character.
-LC=character	:Specify left cross character.
-RC=character	:Specify right cross character.
-CR=character	:Specify cross character.
-XFO[RM]=n	:Specify cross referrence list output format.
-RFM[T]=file	:Specify report format file name.
-SUM[MARY]='text', file	:Specify program title and text file name.
-DFM[T]=file	:Specify document format file name.
-MAI[N]=function_name	:Specify main function name.
-MFM[T]=file	:Specify module format file name.
-PMA[P]	:Create program map list.
-MLE[N]=n	:Specify module name length
-RLE[N]=n	:Specify routine name length.

DEFAULT ASSIGNMENT: -LIS -NOX -NOP -NOM -NOA -NOR -NOSP -NOMO -NOTR -ROW=66 -COL=80 -NOTA=8 -NOF -HEA -WOR=current -GEN -CON -NOSE -NOB -AXI=C -NAM=8 -LIN=H -SEP=PERIOD -DEL = * -NRO=1 -NV=^ -NB=> -STR=& -SOR=1 -SYM=32 -LU =+ -LD=+ -RU=+ -RD=+ -UC=+ -DC=+ -LC=+ -RC=+ -CR=+ -XFO=0 -MLE=32 -RLE=12

(2) LIST17K

17K Series Document Processor Vx. xx [DD MM YY]
Copyright (c) NEC Corporation 1994

usage1 : LIST17K input-file [option[...]]
usage2 : LIST17K @input-file [option[...]]

input_file : Specify object file name.
@input_file: Specify file name of object file name list.

The option is as follows ([ ] means omissible).

-LIST[T]	:Create assemble list.
-LNK=file	:Specify linkage file name.
-ROW=n	:Specify row in a page.
-COL[UMN]=n	:Specify column in a line.
-TAB/-NOT[AB][=n]	:Specify tab code or space.
-FOR[M]/-NOF[ORM]	:Specify form feed code or line feed code.
-HEA[D]/-NOHE[AD]	:Specify printing header or not.
-WOR[K]=path_name	:Specify path name for temporary file.
-GEN/-NOG[EN]	:Specify printing generate lines or not.
-CON[D]/-NOC[OND]	:Specify printing condition lines or not.
-SEQ/-NOS[EQ]	:Specify printing options or not.
-BRA[NCH]/-NOB[RANCH]	:Specify printing label reffer or not.

DEFAULT ASSIGNMENT: -LIS -ROW=66 -COL=80 -NOT=8 -NOF -HEA -WOR=current -GEN -CON -NOS -NOB

(3) MAP17K

```
17K Series Document Processor Vx. xx [DD MM YY]
    Copyright (c) NEC Corporation 1994

usage1   : MAP17K input-file [option[ ...]]
usage2   : MAP17K @input-file [option[ ...]]

input_file   : Specify object file name.
@input_file: Specify file name of object file name list.

The option is as follows ([ ] means omissible).

  -MAP[=file]                          :Create map list [with specified name].
  -LNK=file                            :Specify linkage file name.
  -ROW=n                               :Specify row in a page.
  -COL[UMN]=n                          :Specify column in a line.
  -TAB/-NOT[AB][=n]                    :Specify tab code or space.
  -FOR[M]/-NOF[ORM]                    :Specify form feed code or line feed code.
  -HEA[D]/-NOHE[AD]                    :Specify printing header or not.
  -WOR[K]=path_name                    :Specify path name for temporary file.
  -AXI[S]={C|R}                        :Specify axis column or row.
  -NAM[ELENGTH]=n                      :Specify name length.
  -LIN[EKIND]={H\Z}                    :Specify line kind.
  -SEP[ARATE]={PERIOD\POINT}           :Specify separate line period or point.
  -DEL[IMITER]=character               :Specify delimiter character.
  -NRO[W]=n                            :Specify lines of name.
  -NV=character                        :Specify nibblev character.
  -NB=character                        :Specify nibble character.
  -STR=character                       :Specify structure/union character.
  -SORT=n                              :Specify sorting symbol.
  -SYM[LENGTH]=n                       :Specify symbol name length.
  -LU=character                        :Specify left up corner character.
  -LD=character                        :Specify left down corner character.
  -RU=character                        :Specify right up corner character.
  -RD=character                        :Specify right down corner character.
  -UC=character                        :Specify up cross character.
  -DC=character                        :Specify down cross character.
  -LC=character                        :Specify left cross character.
  -RC=character                        :Specify right cross character.
  -CR=character                        :Specify cross character.

DEFAULT ASSIGNMENT:  -MAP -ROW=66 -COL=80 -NOT=8 -NOF -HEA -WOR=current -AXI=C
                     -NAM=8 -LIN=H -SEP=PERIOD -DEL = * -NRO=1 -NV=^ -NB=> -STR=&
                     -SORT=1 -SYM=32 -LU=+ -LD=+ -RU=+ -RD=+ -UC=+ -DC=+ -LC=+
                     -RC=+ -CR=+
```

(4) AMAP17K

17K Series Document Processor Vx. xx [DD MM YY]
Copyright (c) NEC Corporation 1994

usage1 : AMAP17K input-file [option[...]]
usage2 : AMAP17K @input-file [option[...]]

input_file : Specify object file name.
@input_file: Specify file name of object file name list.

The option is as follows ([ ] means omissible).

-AMA[P][=file]	:Create all map list [with specified name].
-LNK=file	:Specify linkage file name.
-ROW=n	:Specify row in a page.
-COL[UMN]=n	:Specify column in a line.
-TAB/-NOT[AB][=n]	:Specify tab code or space.
-FOR[M]/-NOF[ORM]	:Specify form feed code or line feed code.
-HEA[D]/-NOHE[AD]	:Specify printing header or not.
-WOR[K]=path_name	:Specify path name for temporary file.
-AXI[S]={C|R}	:Specify axis column or row.
-NAM[ELENGTH]=n	:Specify name length.
-LIN[EKIND]={H|Z}	:Specify line kind.
-SEP[ARATE]={PERIOD|POINT}	:Specify separate line period or point.
-DEL[IMITER]=character	:Specify delimiter character.
-NRO[W]=n	:Specify lines of name.
-NV=character	:Specify nibblev character.
-NB=character	:Specify nibble character.
-STR=character	:Specify structure/union character.
-SOR[T]=n	:Specify sorting symbol.
-SYM[LENGTH]=n	:Specify symbol name length.
-LU=character	:Specify left up corner character.
-LD=character	:Specify left down corner character.
-RU=character	:Specify right up corner character.
-RD=character	:Specify right down corner character.
-UC=character	:Specify up cross character.
-DC=character	:Specify down cross character.
-LC=character	:Specify left cross character.
-RC=character	:Specify right cross character.
-CR=character	:Specify cross character.

DEFAULT ASSIGNMENT: -AMA -ROW=66 -COL=80 -NOT=8 -NOF -HEA -WOR=current -AXI=C -NAM=8 -LIN=H -SEP=PERIOD -DEL = * -NRO=1 -NV=^ -NB=> -STR=& -SOR=1 -SYM=32 -LU=+ -LD=+ -RU=+ -RD=+ -UC=+ -DC=+ -LC=+ -RC=+ -CR=+

(5) XREF17K

```
17K Series Document Processor Vx. xx [DD MM YY]
     Copyright (c) NEC Corporation 1994

usage1   : XREF17K input-file [option[ ...]]
usage2   : XREF17K @input-file [option[ ...]]

input_file   :  Specify object file name.
@input_file:  Specify file name of object file name list.

The option is as follows ([ ] means omissible).

   -XRE[F][=file]                  :Create cross referrence list
                                        [with specified name].
   -LNK=file                       :Specify linkage file name.
   -ROW=n                          :Specify row in a page.
   -COL[UMN]=n                     :Specify column in a line.
   -TAB/-NOT[AB][=n]               :Specify tab code or space.
   -FOR[M]/-NOF[ORM]               :Specify form feed code or line feed code.
   -HEA[D]/-NOHE[AD]               :Specify printing header or not.
   -WOR[K]=path_name               :Specify path name for temporary file.
   -XFO[RM]=n                      :Specify cross referrence list output format.

DEFAULT ASSIGNMENT:  -XRE -ROW=66 -COL=80 -NOT=8 -NOF -HEA -WOR=current -XFO=0
```

(6) PUB17K

```
17K Series Document Processor Vx. xx [DD MM YY]
     Copyright (c) NEC Corporation 1994

usage1   : PUB17K input-file [option[ ...]]
usage2   : PUB17K @input-file [option[ ...]]

input_file   : Specify object file name.
@input_file: Specify file name of object file name list.

The option is as follows ([ ] means omissible).

   -PUB[XREF][=file]              :Create public cross referrence list
                                       [with specified name].
   -LNK=file                      :Specify linkage file name.
   -ROW=n                         :Specify row in a page.
   -COL[UMN]=n                    :Specify column in a line.
   -TAB/-NOT[AB][=n]              :Specify tab code or space.
   -FOR[M]/-NOF[ORM]              :Specify form feed code or line feed code.
   -HEA[D]/-NOHE[AD]              :Specify printing header or not.
   -WOR[K]=path_name              :Specify path name for temporary file.
   -XFO[RM]=n                     :Specify public cross referrence list output format.

DEFAULT ASSIGNMENT:  -PUB -ROW=66 -COL=80 -NOT=8 -NOF -HEA -WOR=current -XFO=0
```

(7) REP17K

17K Series Document Processor Vx. xx [DD MM YY]
Copyright (c) NEC Corporation 1994

usage1 : REP17K input-file [option[...]]
usage2 : REP17K @input-file [option[...]]

input_file : Specify object file name.
@input_file: Specify file name of object file name list.

The option is as follows ([ ] means omissible).

-REP[ORT][=file]	:Create report list [with specified name].
-LNK=file	:Specify linkage file name.
-ROW=n	:Specify row in a page.
-COL[UMN]=n	:Specify column in a line.
-TAB/-NOT[AB][=n]	:Specify tab code or space.
-FOR[M]/-NOF[ORM]	:Specify form feed code or line feed code.
-HEA[D]/-NOHE[AD]	:Specify printing header or not.
-WOR[K]=path_name	:Specify path name for temporary file.
-RFM[T]=file	:Specify format file name.

DEFAULT ASSIGNMENT: -REP -ROW=66 -COL=80 -NOT=8 -NOF -HEA -WOR=current

(8) SPEC17K

17K Series Document Processor Vx. xx [DD MM YY]
Copyright (c) NEC Corporation 1994

usage1 : SPEC17K input-file [option[...]]
usage2 : SPEC17K @input-file [option[...]]

input_file : Specify object file name.
@input_file: Specify file name of object file name list.

The option is as follows ([ ] means omissible).

-SPE[C][=file]	:Create document list [with specified name].
-LNK=file	:Specify linkage file name.
-ROW=n	:Specify row in a page.
-COL[UMN]=n	:Specify column in a line.
-TAB/-NOT[AB][=n]	:Specify tab code or space.
-FOR[M]/-NOF[ORM]	:Specify form feed code or line feed code.
-HEA[D]/-NOHE[AD]	:Specify printing header or not.
-WOR[K]=path_name	:Specify path name for temporary file.
-SUM[MARY]='text', file	:Specify program title and text file name.
-DFM[T]=file	:Specify format file name.

DEFAULT ASSIGNMENT: -SPE -ROW=66 -COL=80 -NOT=8 -NOF -HEA -WOR=current

(9) TREE17K

17K Series Document Processor Vx. xx [DD MM YY]
Copyright (c) NEC Corporation 1994

usage1 : TREE17K input-file [option[...]]
usage2 : TREE17K @input-file [option[...]]

input_file : Specify object file name.
@input_file: Specify file name of object file name list.

The option is as follows ([ ] means omissible).

-TRE[E][=file]	:Create call tree list [with specified name].
-LNK=file	:Specify linkage file name.
-ROW=n	:Specify row in a page.
-COL[UMN]=n	:Specify column in a line.
-TAB/-NOT[AB][=n]	:Specify tab code or space.
-FOR[M]/-NOF[ORM]	:Specify form feed code or line feed code.
-HEA[D]/-NOHE[AD]	:Specify printing header or not.
-WOR[K]=path_name	:Specify path name for temporary file.
-MAI[N]=function_name	:Specify main function name.

DEFAULT ASSIGNMENT: -TRE -ROW=66 -COL=80 -NOT=8 -NOF -HEA -WOR=current

(10) MOD17K

17K Series Document Processor Vx. xx [DD MM YY]
Copyright (c) NEC Corporation 1994

usage1 : MOD17K input-file [option[...]]
usage2 : MOD17K @input-file [option[...]]

input_file : Specify object file name.
@input_file: Specify file name of object file name list.

The option is as follows ([ ] means omissible).

-MOD[ULE][=file]	:Create module list [with specified name].
-LNK=file	:Specify linkage file name.
-ROW=n	:Specify row in a page.
-COL[UMN]=n	:Specify column in a line.
-TAB/-NOT[AB][=n]	:Specify tab code or space.
-FOR[M]/-NOF[ORM]	:Specify form feed code or line feed code.
-HEA[D]/-NOHE[AD]	:Specify printing header or not.
-WOR[K]=path_name	:Specify path name for temporary file.
-MFM[T]=file	:Specify format file name.
-PMA[P]	:Create program map list.
-MLE[N]=n	:Specify module name length.
-RLE[N]=n	:Specify routine name length.

DEFAULT ASSIGNMENT: -MOD -ROW=66 -COL=80 -NOT=8 -NOF -HEA -WOR= current -MLE=32 -RLE=12

2.2 Start and End Messages

The Document Processor issues a normal end message to the standard output when processing ends normally. It issues an abnormal end message if the processing ends abnormally.

(1) Start message

```
17K Series Document Processor Vx.xx [DD MM YY]
   Copyright (c) NEC Corporation 1994
- process-name start hh:mm:ss yy/dd/mm -
```

Remarks 1. Vx.xx : Version number
[DD MM YY] : Date when the process was developed
hh:mm:ss : Time when the program was started
yy/dd/mm : Date on which the program was started

2. process-name

Document Processor	: DOC17K
Absolute address format listing	: LIST17K
Module map listing	: MAP17K
Program map listing	: AMAP17K
Cross-reference listing	: XREF17K
Public cross-reference listing	: PUB17K
Report listing	: REP17K
Document listing	: SPEC17K
Call tree listing	: TREE17K
Module listing	: MOD17K

(2) End messages

<1> Message output upon normal end

```
- process-name end hh:mm:ss yy/dd/mm -
Total error(s) : x Total warning(s) : y
```

Remark hh:mm:ss : Time when the program was ended
yy/dd/mm : Date on which the program was ended
x : Number of errors
y : Number of warnings

<2> Message output upon abnormal end

```
- process-name end hh:mm:ss yy/dd/mm -
Program aborted.
```

Remark hh:mm:ss : Time when the program was ended
yy/dd/mm : Date on which the program was ended

2.3 Cancellation (CTRL + C)

The cancellation key sequence (CTRL + C) returns control to the OS, deleting all temporary files and output files that have been opened.

If BREAK=ON is not defined in the CONFIG.SYS file (the default is BREAK=OFF), the CTRL + C interrupt is enabled only upon message output and a limited number of other instances. Consequently, the CTRL + C interrupt may be rejected during listing.

To enable the CTRL + C interrupt during disk access, for example, specify BREAK=ON in the CONFIG.SYS file. Remember, however, that specifying BREAK=ON has an adverse affect on processing speed due to the constant monitoring of the CTRL + C interrupt.

[MEMO]

Chapter 3

Supported Output Listings and Options

DOC17K outputs the listings indicated in Table 3-1

Some listing utilities support the specification of detailed output options. For more information on these output options, see the description of the options for each listing utility.

Table 3-1. Types of Output Listings and Options

Output listing	Output file extension	DOC17K execution option
Absolute address format listing	.PRN	-LIS[T]
Optional information output		-SEQ
Embedded cross-reference output		-BRA[NCH]
Cross-reference listing	.XRF	-XRE[F]
Public cross-reference listing	.PUB	-PUB[XREF]
Module map listing	.MAP	-MAP
Program map listing	.AMP	-AMA[P]
Report listing	.REP	-REP[ORT]
Document listing	.SPC	-SPE[C]
Module listing	.MOD	-MOD[ULE]
Address map output		-PMA[P]
Call tree listing	.TRE	-TRE[E]

3.1 Options

There are three types of options:

- Options that specify a listing utility
- Options common to all listing utilities
- Listing-specific options

3.1.1 Options That Specify a Listing Utility

Table 3-2. Options That Specify a Listing Utility

Option name	Default	Explanation
-LNK=filename	–**(Note)**	Specifies the link object file (.LNK) produced by the linker (LK17K).
-LIS[T] [=filename] -NOL[IST]	LIS	Produces an absolute address format listing.
-XRE[F] [=filename] -NOX[REF]	NOX	Produces a cross-reference listing.
-PUB[XREF] [=filename] -NOP[UBXREF]	NOP	Produces a public cross-reference listing.
-MAP[=filename] -NOM[AP]	NOM	Produces a module map listing (module-specific data memory map).
-AMA[P] [=filename] NOA[MAP]	NOA	Produces a program map listing (program - data memory map).
-REP[ORT] [=filename] -NOR[EPORT]	NOR	Produces a report listing
-SPE[C] [=filename] -NOSP[EC]	NOSP	Produces a document listing.
-MOD[ULE] [-filename] -NOMO[DULE]	NOMO	Produces a module listing.
-PMA[P]	–	Produces a program address map.
-TRE[E] [=filename] -NOTR[EE]	NOTR	Produces a call tree listing.

(Note) -LNK must be specified to execute DOC17K and the listing utilities. See **Section 2.1**.

(1) Input files

The files listed in Table 3-3 are needed to output listings.

Specify the required file output options when starting RA17K, *emlC-17K*, and LK17K.

For more details on the file output options, refer to the RA17K, *emlC-17K*, and LK17K User's Manuals.

Table 3-3. Input Files Required to Enable Listing Output

Input file / Output listing (option)	Intermediate PRN file	REL file	LNK file	XREF file	Summary file	Log file
Absolute address format listing (-LIS[T])	O	O	O	O**(Note 1)**		
Cross-reference listing (-XRE[F])		O	O	O		
Public cross-reference listing (-PUB[XREF])						
Module map listing (-MAP)	O**(Note 2)**	O	O			
Program map listing (-AMA[P])						
Report listing (-REP[ORT])		O	O			O
Document listing (-SPE[C])	O	O	O	O	O	
Module listing (-MOD[ULE])	O	O				
Call tree listing (-TRE[E])	O**(Note 3)**		O			

(Note 1) Used when -BRA[NCH] is specified.
(Note 2) Used to extract symbol definition comments.
(Note 3) With *emlC-17K*, use a preprocessing file instead of an intermediate PRN file.

(2) Input file extensions

Input files are named using the specified filename, to which one of the following extensions is appended.

Table 3-4. Input File Extensions

Specification / Input file	Module name specification	Parameter filename specification
Intermediate PRN file Preprocessing file	Module name.IPN Module name.PPL	Module name.IPN Module name.PPL
REL file	Module name.REL	Module name.REL
LNK file	Specified filename.LNK	Specified filename.LNK
XREF file	Module name.IXF	Module name.IXF
Summary file	Module name.SUM	Module name.SUM
Log file	Module name.LOG	Module name.LOG

3.1.2 Options Common to All Listing Utilities

The output listing format control options are common to all listing utilities.

Table 3-5 lists the options that are common to all listing utilities.

Table 3-5. Options That Are Common to All Listing Utilities

Option name	Default	Contents
-ROW=n	n = 66	Number of lines per page of the listing
-COL[UMN]=n	n = 80	Number of columns per line of the listing
-TAB -NOT[AB] [=n]	NOT = 8	Tab conversion control
-FOR[M] -NOF[ORM]	NOF	Form feed control
-HEA[D] -NOH[EAD]	HEA	List header control
-WOR[K]=pathname	Current	Work drive and directory specification

(1) -ROW=n

Specifies the number of output lines per page ($50 \leq n \leq 255$). The default is n = 66.

(2) -COL[UMN]=n

Specifies the number of output columns per line (72 ≤ n ≤ 255). The default is n = 80. Any part of the listing that exceeds the specified number of columns is not output. When the specified number of columns per line is not in a multiple of double-byte characters, the listing is truncated one column before the double-byte character.

A sufficiently large value should be specified for COL. Any map listing that exceeds the number of columns per line specified here is directed to the next page. A listing that exceeds the number of columns for two pages is not output.

If the number of columns per line is specified in the format file for document listing, that number is used.

(3) -TAB/-NOT[AB] [=n]

Specifies whether to replace each occurrence of a TAB code with a blank or output it as is. The default is -NOT[AB]=8.

Table 3-6. -TAB Option Settings

Setting	Explanation
-NOT[AB] [=n]	Insert a blank to align the character following the TAB with a column that is an integer multiple of n (1 ≤ n ≤ 255).
-TAB	Output the TAB code as is.

(4) -FOR[M]/-NOF[ORM]

Specifies whether to feed forms with null lines to output a form feed code (FF). The default is -NOF[ORM].

Table 3-7. -FORM Option Settings

Setting	Explanation
-FOR[M]	Feed forms with FF codes.
-NOF[ORM]	Output CR/LF codes until the line specified by -ROW.

(5) -HEA[D]/-NOH[EAD]

Controls header output at the beginning of each page. The default is -HEA[D].

Table 3-8. -HEAD Option Settings

Setting	Explanation
-HEA[D]	Output a header on each page.
-NOH[EAD]	Output a header on the first page only.

When -NOH[EAD] is specified, the following header is omitted:

```
RA17K V1.00 V1 <<******* *** LIST>>...
PROG = *****
```

(6) -WOR[K]=pathname

Specifies the work drive and directory used by the Document Processor. If environmental variable TMP is specified, the TMP path is used when this option is omitted.

If environmental variable TMP is not specified, the current path is used.

3.1.3 Listing-Specific Options

Listing-specific options can be specified.

All options for each listing utility can be specified when DOC17K starts. Once a listing utility has been run independently, however, only those options specific to that listing can be specified. Remember that an option error is likely to result if the parameter file used to start DOC17K is used as the parameter file for running a particular listing utility.

(1) Absolute address format listing options

Table 3-9. Absolute Address Format Listing Options

Option name	Default	Explanation
-GEN -NOG[EN]	GEN	Macro and repeat pseudo instruction expansion control
-CON[D] -NOC[OND]	CON	Conditional statement control
-SEQ -NOS[EQ]	NOS	Option information listing control
-BRA[NCH] -NOB[RANCH]	NOB	Embedded cross-reference listing control

(a) -GEN/-NOG[EN]

Controls the listing of the macro-expanded part of a macro statement. The default is -GEN. With -NOG[EN], output is controlled by the macro expansion control pseudo instructions in the source program.

Table 3-10. -GEN Option Settings

Setting	Explanation
-GEN	Lists all statements.
-NOG[EN]	• SMAC/VMAC/NOMAC/OMAC/LMAC pseudo instructions Controls the listing of the macro-expanded part. • SBMAC/VBMAC/NOBMAC/OBMAC/LBMAC pseudo instructions Controls the listing of the embedded macro-expanded part.

Remark Refer to the RA17K User's Manual for details of the macro expansion control pseudo instructions.

(b) -CON[D]/-NOC[OND]

Controls the listing of an IF, IFCHAR, IFNCHAR, IFSTR, or CASE conditional statement when the is false. The default is -CON[D].

Table 3-11. -COND Option Settings

Setting	Explanation
-CON[D]	Lists all statements regardless of the condition.
-NOC[OND]	Controls the listing of the conditional statement with SFCOND/ LFCOND pseudo instructions in the source program.

Remark Refer to RA17K User's Manual for more details on the conditional listing output control pseudo instructions.

(c) -SEQ/-NOS[EQ]

Controls the listing of option information. The default is -NOS[EQ].

Table 3-12. -SEQ Option Settings

Setting	Explanation
-SEQ	Produces an option information listing.
-NOS[EQ]	Does not produce an option information listing.

(d) -BRA[NCH]/-NOB[RANCH]

Controls the listing of label reference line information (embedded cross-references). The default is -NOB[RANCH].

Table 3-13. -BRANCH Option Settings

Setting	Explanation
-BRA[NCH]	Lists embedded cross-references.
-NOB[RANCH]	Does not list embedded cross-references.

(2) Module map listing and program map listing options

Table 3-14. Module Map Listing and Program Map Listing Options

Option name	Default	Explanation
-AXI[S]={C|R}	C	Vertical and horizontal axis specification
-NAM[ELENGTH]=n	n = 8	number-of-characters-per-box specification
-LIN[EKIND]={H|Z}	H	Line kind specification
-SEP[ARATE]={PERIOD|POINT}	PERIOD	PUBLIC/LOCAL separator line type specification
-DEL[IMITER]=character	*	Deleted character setting
-NRO[W]=n	n = 1	Symbol name number-of-characters specification
-NV=character	^	NIBBLEnV attribute display character setting
-NB=character	>	NIBBLEn attribute display character setting
-STR=character	&	STRUCTURE/UNION attribute display character setting
-SOR[T]=n	n = 1	Symbol order specification
-SYM[LENGTH]=n	n = 32	Symbol name number-of-characters-per-line specification

Remarks 1. Decimal digits may be specified for options.
2. Only single-byte single characters can be used to specify options on PC/AT host machines. Both single- and double-byte single characters are supported by PC-9800 host machines.

(a) -AXI[S]={C|R}

Specifies the vertical and horizontal axes of a map listing in tabular form. The default is C.

Table 3-15. -AXIS Option Settings

Setting	Explanation
R	Sets the vertical address for the row address (0xH to 7xH) and that for the column address (x0H to xFH).
C	Sets the vertical address for the column address (x0H to xFH) and that for the row address (0xH to 7xH).

(b) -NAM[ELENGTH]=n

Specifies the number of characters that can be displayed in a row of a box ($8 \leq n \leq 255$). The default is 8. The number of characters actually displayed equals NAMELENGTH * NROW.

(c) -LIN[EKIND]={H|Z}

Specifies the frame lines to be drawn. The default is H.

Table 3-16. -LINEKIND Option Settings

Setting	Explanation
Z	Double-byte line
H	Single-byte line

Caution **With PC/AT host machines, double-byte lines (-LINEKIND=Z) cannot be specified.**

In addition, the following options can be specified to set a line kind in part:

Table 3-17. -LINEKIND Option Settings (Detailed Settings)

Setting	Explanation
-LU=character	Upper left corner (default: double-byte "┌", single-byte "+")
-LD=character	Lower left corner (default: double-byte "└", single-byte "+")
-RU=character	Upper right corner (default: double-byte "┐", single-byte "+")
-RD=character	Lower right corner (default: double-byte "┘", single-byte "+")
-UC=character	Upper intersection (default: double-byte "┬", single-byte "+")
-DC=character	Lower intersection (default: double-byte "┴", single-byte "+")
-LC=character	Left intersection (default: double-byte "├", single-byte "+")
-RC=character	Right intersection (default: double-byte "┤", single-byte "+")
-CR=character	Cross-intersection (default: double-byte "┼", single-byte "+")

(d) -SEP[ARATE]={PERIOD|POINT}

Specifies the type of public/local separator lines. The default is PERIOD.

Table 3-18. -SEPARATE Option Settings

Setting	Explanation
POINT	Point
PERIOD	Period

Caution With PC/AT host machines, the point (-SEPARATE=POINT) cannot be specified.

(e) -DEL[IMITER]=character

Specifies a delimiter character that is attached to a symbol name when it exceeds the value specified by -NAM[ELENGTH] or -NRO[W]. "*" is assumed by default.

(f) -NRO[W]=n

Specifies the number of lines ($1 \leq n \leq 5$) that are displayed when a symbol name exceeds the value specified by -NAM[ELENGTH]. The default is 1. When a symbol name exceeds the specified number of lines, it is terminated by a delimiter and the subsequent characters are not displayed.

The number of characters actually displayed equals NAMELENGTH * NROW.

(g) -NV=character

Specifies the character used to identify a NIBBLEnV attribute symbol. "^" is assumed by default. This identification character is prefixed to each symbol name when it is displayed.

(h) -NB=character

Specifies the character used to identify a NIBBLEn attribute symbol. ">" is assumed by default. This identification character is prefixed to each symbol name when it is displayed.

(i) -STR=character

Specifies the character used to identify a STRUCTURE (structure or union) attribute symbol. "&" is assumed by default. This identification character is prefixed to each symbol name when it is displayed.

(j) -SOR[T]=n

Specifies the order in which symbol names are output in the symbol name listing. If this option is omitted, symbol names are output in alphabetic order.

Table 3-19. -SORT Option Settings

Setting	Explanation
0	In order of appearance
1	In alphabetic order
2	In address order

(k) -SYM[LENGTH]=n

Specifies the symbol name length of the symbol name listing ($1 \leq n \leq 255$). The default is 32. A symbol name longer than 32 characters is output using two lines, together with symbol information.

(3) Cross-reference listing option

Table 3-20. Cross-Reference Listing Option

Option name	Default	Explanation
-XFO[RM]=n	0	Specifies the output format of a cross-reference listing.

(a) -XFO[RM]=n

Specifies the output format of a cross-reference listing. The default is 0.

Table 3-21. -XFORM Option Settings

Setting	Explanation
0	Outputs definition/reference line numbers on one line.
1	Outputs a filename and a reference line number for each reference in a tag jump format.

[Sample output] -XFO[RM]=0

```
SYMBOL                VALUE     ATTR    TYPE    /REF(#DEF)
Symbol name           00000001  Local   DAT     /#1, 2
```

[Sample output] -XFO[RM]=1

```
MODULE(LineNo.)          : VALUE    : ATTR  : TYPE  : SYMBOL
B:\RA17K\SAMPLE.ASM(5)   : 00000001 : Local : MEM   : symbol-name
B:\RA17K\SAMPLE.ASM(8)
```

(4) Report listing option

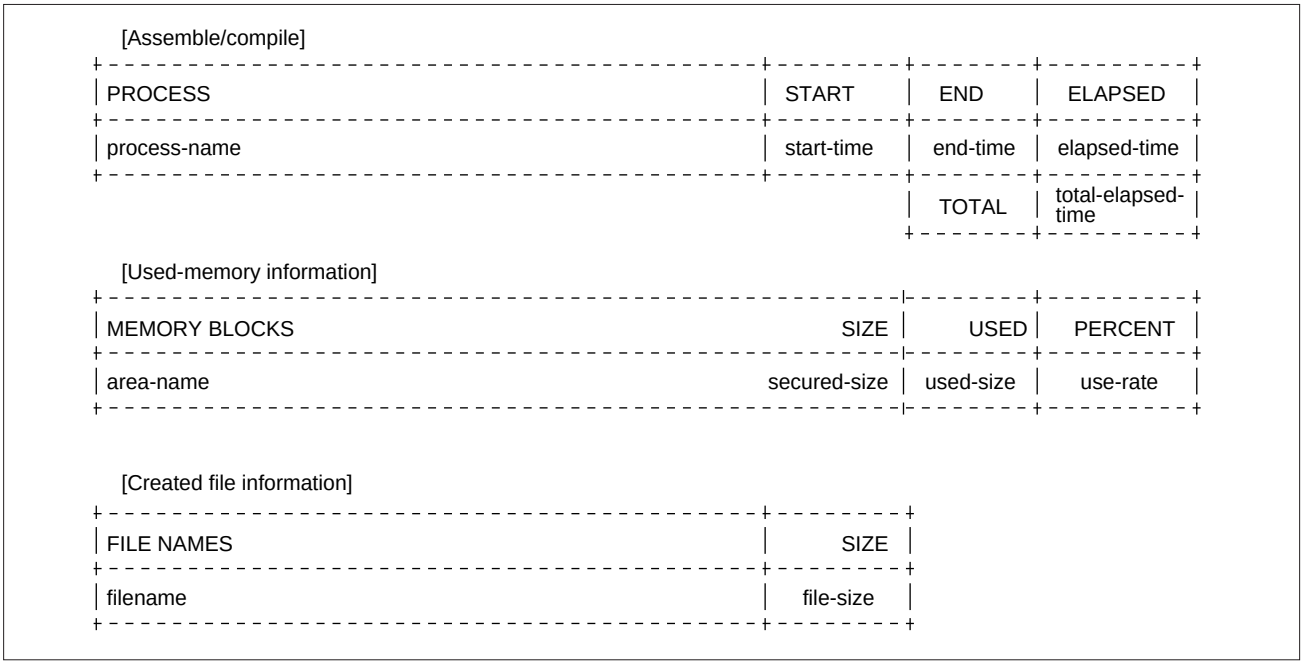
Table 3-22. Report Listing Option

Option name	Default	Explanation
-RFM[T]=filename	–	Names a report listing format file.

(a) -RFM[T]=filename

Names the format file containing the user-specified format in which a report listing is to be produced. If this option is omitted, the output format shown in Figure 3-1 is assumed.

Figure 3-1. Report Listing Output Format (Default)



(5) Document listing options

Table 3-23. Document Listing Options

Option name	Default	Explanation
-SUM[MARY]='program title',filename	–	Output the program summary
-DFM[T]=filename	–	Names a module and routine format file.

(a) -SUM[MARY]='program-title',filename

This option allows program summaries to be included in the document listing as desired. The contents of the file specified by title and filename are output in the document listing.

(b) -DFM[T]=filename

Names the format file containing the user-specified format in which modules and routines are to be listed. The output format shown in Figure 3-2 is assumed when this option is omitted.

Figure 3-2. Document Listing Output Format (Default)

```
[Module summary]
+-------------------------------------------------------------------------------+
| Chapter n.   Module Title                                                     |
+-------------------------------------------------------------------------------+
| [Specifications]                                                              |
|                                                                               |
| Module Summary                                                                |
|                                                                               |
+-------------------------------------------------------------------------------+
| Module Name       : module-name                                               |
| Create            : created-date                                              |
| Size              : file-size                                                 |
| Step              : steps                                                     |
+-------------------------------------------------------------------------------+
| PUBLIC(MEM)       : PUBLIC-symbol-name                                        |
| PUBLIC(FLG)       : PUBLIC-symbol-name                                        |
| PUBLIC(DAT)       : PUBLIC-symbol-name                                        |
| PUBLIC(LAB)       : PUBLIC-symbol-name                                        |
| EXTRN(MEM)        : EXTRN-symbol-name                                         |
| EXTRN(FLG)        : EXTRN-symbol-name                                         |
| EXTRN(DAT)        : EXTRN-symbol-name                                         |
| EXTRN(LAB)        : EXTRN-symbol-name                                         |
+-------------------------------------------------------------------------------+

[Routine summary]
+-------------------------------------------------------------------------------+
| n. m   Routine Title                                                          |
+-------------------------------------------------------------------------------+
| [Specifications]                                                              |
|                                                                               |
| Routine Summary                                                               |
|                                                                               |
+-------------------------------------------------------------------------------+
| [Routine Information]                                                         |
|                                                                               |
| Address Range             : start-address to end-address                      |
| Entrance                  : entry-name                                        |
| MEM Changed               : changed MEM type symbol                           |
| MEM Referred              : referred MEM type symbol                          |
| MEM Manipulated           : manipulated MEM type symbol                       |
| FLG Changed               : changed FLG type symbol                           |
| FLG Referred              : referred FLG type symbol                          |
| DAT Referred              : referred DAT type symbol                          |
| Branch To                 : branch to LAB type symbol                         |
| Subroutines Called        : subroutines called LAB type symbol                |
| LAB Manipulated           : manipulated LAB type symbol                       |
| System Call               : SYSCAL entry name                                 |
+-------------------------------------------------------------------------------+
| TAG information                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
+-------------------------------------------------------------------------------+
```

(6) Call tree listing option

Table 3-24. Call Tree Listing Option

Option name	Default	Explanation
-MAI[N]=main-function-name	MAIN	Names the first function to appear in the tree listing.

(a) -MAI[N]=main-function-name

Names a main function in the tree listing. If this option is omitted, a tree listing is produced from ”MAIN”.

(7) Module listing options

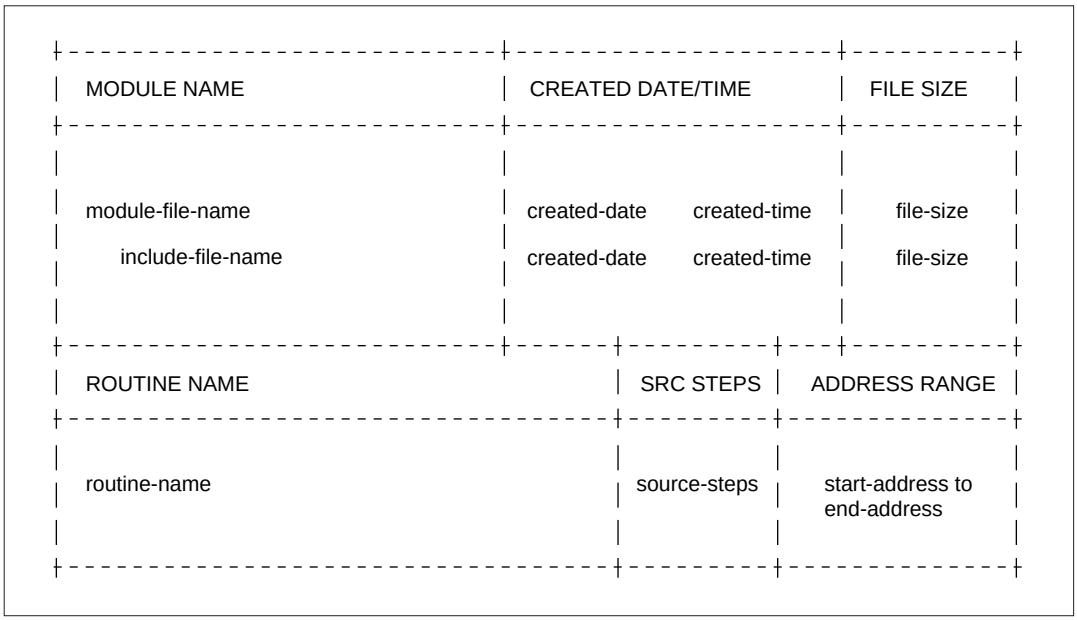
Table 3-25. Module Listing Options

Option name	Default	Explanation
-MFM[T]=filename	–	Names a module listing format file.
-PMA[P]	–	Produces a program address map.
-MLE[N]=n	32	Specifies a module name length.
-RLE[N]=n	12	Specifies a function (routine) name length.

(a) -MFM[T]=filename

Names the format file that contains the user-specified format in which a module listing is to be produced. If this option is omitted, the output format shown in Figure 3-3 is assumed.

Figure 3-3. Module Listing Output Format (Default)

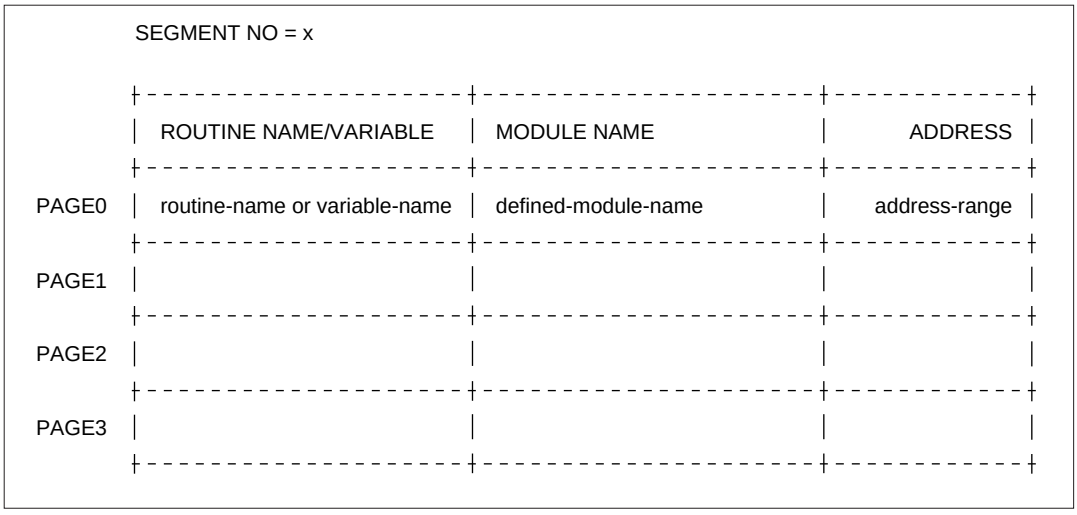


(b) -PMA[P]

Specifies program address map output. Address map output is suppressed if -PMA[P] is omitted.

When -PMA[P] is specified, an address map is appended to a module listing output file (.MOD). An address map is produced page by page (PAGE0 to PAGE3).

Figure 3-4. Address Map Output Format



(c) -MLE[N]=n

Specifies the length of the module name output field ($12 \leq n \leq 64$). The default is 12.

(d) -RLE[N]=n

Specifies the length of the routine name output field ($22 \leq n \leq 128$). The default is 32.

Chapter 4

Output Listing

4.1 Option Information Listing

Information relating to the options specified at the initiation of the assembler (RA17K), compiler *(emIC-17K)*, linker (LK17K), and Document Processor (DOC17K), can be output as a listing.

(1) Option

Specify -SEQ to produce an option information listing. When -NOL[IST] is specified, however, specifying -SEQ does not allow option information listing. An option information listing is output at the beginning of each module-specific absolute address format listing.

(2) Output file

Absolute address format listing (.PRN)

(3) Output details

An option listing is output.

(a) MODULE OPTION

The assembly (or compilation) options specified at the initiation of RA17K (or *emIC-17K*) and the options specified at the initiation of DOC17K are listed.

The options are listed along with their specified values, or with the defaults when no values are specified.

(b) LINK OPTION

The linker options specified at the initiation of LK17K are listed.

The options are listed along with their specified values, or with the defaults when no values are specified.

[Sample output]

```
RA17K        V1.00 V1 << D17XXX    ASSEMBLE LIST >>  HH:MM:SS MM/DD/YY  PAGE 001

PROG = SAMPLE

<MODULE OPTION>

 NOOBJ/OBJ                : OBJ=A:\RA17K\SAMPLE. REL
 NOUND/UND                : NOUND
 INCLUDE                  : INCLUDE=
 NOLIST/LIST              : LIST=A:\RA17K\SAMPLE. IPN
 NOXREF/XREF              : XREF=A:\RA17K\SAMPLE. IXF
 ZZZ0                     : ZZZ0=0
 ZZZ1                     : ZZZ1=0
 ZZZ2                     : ZZZ2=0
 ZZZ3                     : ZZZ3=0
 ZZZ4                     : ZZZ4=0
 ZZZ5                     : ZZZ5=0
 ZZZ6                     : ZZZ6=0
 ZZZ7                     : ZZZ7=0
 ZZZ8                     : ZZZ8=0
 ZZZ9                     : ZZZ9=0
 ZZZ10                    : ZZZ10=0
 ZZZ11                    : ZZZ11=0
 ZZZ12                    : ZZZ12=0
 ZZZ13                    : ZZZ13=0
 ZZZ14                    : ZZZ14=0
 ZZZ15                    : ZZZ15=0
 WARNING                  : WARNING=4
 ABSOLUTE/RELOCATABLE     : RELOCATABLE
 SUMMARY/NOSUMMARY        : NOSUMMARY
 HOST/NOHOST              : NOHOST
 WORK                     : WORK=D:\
 TAGSTART                 : TAGSTART=.
 TAGEND                   : TAGEND=..
 NOLIST/LIST              : LIST=SAMPLE. PRN
 SEQ/NOSEQ                : SEQ
 BRANCH/NOBRANCH          : NOBRANCH
 GEN/NOGEN                : GEN
 COND/NOCOND              : COND
 NOTAB/TAB                : NOTAB=8
 FORM/NOFORM              : NOFORM
 ROW                      : ROW=66
 COLUMN                   : COLUMN=80
 WORK                     : WORK=D:\
 LNK                      : LNK=SAMPLE.LNK
 HEAD/NOHEAD              : HEAD
```

```
< LINK OPTION >

 NOOBJ/OBJ                  : OBJ=A:\RA17K\SAMPLE. LNK
 NOICE/ICE                  : ICE=A:\RA17K\SAMPLE.ICE
 NOPROM/PROM                : NOPROM
 HOST/NOHOST                : NOHOST
 WORK                       : WORK=D:\
 WARNING                    : WARNING=0
 INCREMENT/NOINCREMENT      : NOINCREMENT
 DIRECTIVE                  : DIRECTIVE=
 PARAMETER                  : PARAMETER=
 PROG                       : PROG=
 INDIRECT                   : INDIRECT=
 JUNK/NOJUNK                : NOJUNK
 NOL/LMAP                   : LMAP=A:\RA17K\SAMPLE.LMP

RA17K       V1.00 V1  << D17XXX    ASSEMBLE LIST >>   HH:MM:SS MM/DD/YY PAGE 002

PROG =

 ML/NOML                    : NOML
 MP/NOMP                    : NOMP
```

4.2 Absolute Address Format Listing

The results of source program assembly and linkage are converted to absolute addresses for assembly listing.

(1) Option

Specify -LIST to produce an absolute address format listing.

(2) Output file

Absolute address format listing (.PRN)

(3) Output details

(a) Line format

```
EEEE STNO LOC.OBJECT M I SOURCE STATEMENT
```

- EEEE : Error code (A four-character string indicating the type of an error as it occurs.)
- STNO : Source line number.
- LOC. : Location address (absolute address). Address 1xxxx for EPA. For a branch table, addresses and object codes are listed with section names on the comment lines.
- OBJECT : Object code.
- M : Macro nesting level.
- I : Include nesting level.
- SOURCE STATEMENT : Source program contents.

Remark Refer to the RA17K User's Manual for details of the line format.

(b) Listing of statements expanded in the macro, repeat, and include parts

Statements expanded in the macro, repeat, and include parts are listed according to the following:

- STNO : A "+" sign and the line number of each expanded statement are listed until the expansion ends. The expanded statement line number is incremented each time expansion is performed.
- M : The nesting levels of macros, repeat instructions, and built-in macros are listed.
- I : The nesting level of INCLUDE pseudo instructions is listed.

(c) Error statement listing

If error statements have been encountered, error codes and error numbers are listed in place of EEEE. Refer to the RA17K or *emIC-17K* User's Manual for details of the error codes and error numbers.

(d) End of listing

The end of a listing is identified by the output of the following information:

- TOTAL ERRORS : Total number of errors that occurred during source module assembly.
- TOTAL WARNINGS : Total number of warnings generated during source module assembly.

(e) Related pseudo instructions

- TITLE : Whenever a TITLE pseudo instruction is encountered, a form feed is performed and a title is output as a header on the new page. Title output continues until the next TITLE pseudo instruction is encountered.

- EJECT : A form feed is performed whenever an EJECT pseudo instruction is encountered.
- LIST/NOLIST : Statement listing is controlled.
- SFCOND/LFCOND : Conditional statement listing is controlled.
- C14344/C4444 : The object code format is controlled.
- SMAC/VMAC/NOMAC/OMAC/LMAC
  : Listing of the macro-expanded part is controlled.
- SBMAC/VBMAC/NOBMAC/OBMAC/LBMAC
  : Listing of the embedded macro-expanded part is controlled.

Remark Refer to the RA17K User's Manual for details of the listing control instructions.

[Sample output]

```
RA17K V1.00 V1   << D17XXX ASSEMBLE LIST >>   HH:MM:SS MM/DD/YY PAGE 001

PROG = SAMPLE

SOURCE = SAMPLE. ASM

EEEE STNO   LOC.   OBJECT   M  I      SOURCE STATEMENT
       225                             ;*****************************************
       226                             ;*
       227                             ;*  RESET  POWER ON RESET PROCESS
       228                             ;*
       229                             ;*****************************************
       230  00000D                     IRESET:
       231  00000D  071F0                 DI                          INTERRUPT DISABLE
       232
       233                             ;*****************************************
       234                             ;*  PORT INITIALIZE
       235                             ;*****************************************
       236
       237  00000E  1D790                 MOV    BANK, #0
       238  00000F  1D700                 MOV    RP0A, #CP0AINI ; PORT0A
       239  000010  1D710                 MOV    RP0B, #CP0BINI ; PORT0B
       240  000011  1D720                 MOV    RP0C, #CP0CINI ; PORT0C KEY STB
       241  000012  1D730                 MOV    RP0D, #CP0DINI ; PORT0D KEY IN
       242
       243  000013  1D791                 MOV    BANK, #1
       244  000014  1D700                 MOV    RP1A, #CP1AINI ; PORT1A
       245  000015  1D710                 MOV    RP1B, #CP1BINI ; PORT1B
       246  000016  1D720                 MOV    RP1C, #CP1CINI ; PORT1C
       247  000017  1D730                 MOV    RP1D, #CP1DINI ; PORT1D
       248
       249  000018  1D792                 MOV    BANK, #2
       250  000019  1D700                 MOV    RP2A, #CP1AINI ; PORT2A
       251  00001A  1D710                 MOV    RP2B, #CP1BINI ; PORT2B
       252  00001B  1D720                 MOV    RP2C, #CP1CINI ; PORT2C
       253
Using the EPA area
       254  10001C  1D790                 MOV    BANK, #0
       255  10001D  1D7E0                 MOV    RPL, #BRBANK0
       256
       257                             ;*****************************************
       258                             ;*  REGISTER FILE SET
       259                             ;*****************************************
       260                 1           INITFLG   P2ABIO3, P2ABIO2, P2ABIO1, P2ABIO0
  +      1                 2             SET4      P2ABIO3, P2ABIO2, P2ABIO1, P2ABIO0
  +      2 10001E 1D78F    2               MOV     WR, #1111B
  +      3 10001F 07324    2               POKE    .MF.P2ABIO3 SHR 4. WR
  Using the JUMP table                   ; Jump Table:?INIT
           00000   0C000

TOTAL ERRORS = 0
TOTAL WARNINGS = 0

END OF LIST
```

4.3 Cross-Reference Listing

4.3.1 Embedded Cross-References

LIST17K lists the reference line numbers of the labels appearing in source programs in an absolute address format listing.

(1) Option

Specify -BRA[NCH] to list embedded cross-references. When -NOL[IST] is specified, however, the specification of -BRA[NCH] does not permit embedded cross-reference listing.

(2) Output file

Absolute address format listing (.PRN)

(3) Output details

Those addresses for referencing labels and the types of instructions are listed.

- P-xxxx : A public label referenced from another module. Used at address xxxx.
- B-xxxx : Used by the BR instruction at address xxxx.
- C-xxxx : Used by the CALL instruction at address xxxx.

[Sample output]

```
RA17K V1.00 V1   << D17XXX ASSEMBLE LIST >>   HH:MM:SS MM/DD/YY PAGE 001

PROG = SAMPLE

SOURCE = SAMPLE. ASM

EEEE STNO   LOC.   OBJECT   M  I      SOURCE STATEMENT
          5                           PUBLIC   IRESET
           :
        123                           ;*****************************************
        124                           ;*  RESET    POWER ON RESET PROCESS
        125                           ;*****************************************
                                      P-23F, P-242, B-124, C-128
        126  00000D                   IRESET:
        127  00000D  071F0               DI                      : INTERRUPT DISABLE
        128
        129                           ;*****************************************
        130                           ;*  PORT INITIALIZE
        131                           ;*****************************************
        132
        133  00000E  1D790              MOV  BANK, #0
        134  00000F  1D700              MOV  RP0A, #CP0AINI ; PORT0A
        135  000010  1D710              MOV  RP0B, #CP0BINI ; PORT0B
        136  000011  1D720              MOV  RP0C, #CP0CINI ; PORT0C KEY STB
        137  000012  1D730              MOV  RP0D, #CP0DINI ; PORT0D KEY IN
        138
           :
Reference within the same module
           :
        234  000124  0C00D              BR    IRESET            ; BR RESET
           :
        238  000128  1C00D              CALL  IRESET            ; CALL RESET

Reference from another module
          4                           EXTRN      LAB:IRSET
           :
         84  00023F  0C00D              BR    IRESET            ; BR RESET
           :
         98  000242  1C00D              CALL  IRESET            ; CALL RESET
```

4.3.2 Table Cross-References

(1) Local cross-references

The definitions and reference positions of symbols referenced within a source module are listed.

(a) Option

Specify -XRE[F] to list local cross-references.

(b) Output file

Cross-reference listing (.XRF)

(c) Output details

(i) -XFO[RM]=0

```
symbol-name evaluation-value attribute type reference/definition-line-number(#definition-line-number)
```

(ii) -XFO[RM]=1

```
Definition line: source-module-name(nn):evaluation-value:attribute:
type :symbol-name
Reference line: source-module-name(nn)
```

The following table details the line format.

Legend	Output value	Explanation
Definition line number	nn	Indicates the number of the line on which a symbol is defined.
Reference line number		Indicates the number of the line from which the symbol is referenced
Evaluation value (VALUE)	32 bits (hexadecimal)	Refer to the RA17K User's Manual for details.
Attribute (ATTR)	Public	A symbol that can be referenced from other modules
	Local	A symbol that can be referenced only within the module in which it appears.
Type (TYPE)	DAT	Data type
	MEM	Memory type
	Nn	Horizontal nibble memory type
	NnV	Vertical nibble memory type
	LAB	Label type
	FLG	Flag type
	MAC	Macro
	STR	Structure
	UNI	Union
	ARY	Array

Remark Up to about 8,000 symbols can be contained within a cross-reference listing.

[Sample output] -XFO[RM]=0

```
RA17K V1.00 V1 << D17XXX    XREF LIST >>  HH:MM:SS MM/DD/YY PAGE 01-001
PROG = SAMPLE
SOURCE = SAMPLE. ASM
  SYMBOL                    VALUE   ATTR TYPE   /REF(#DEF)
  Symbol name               00000001 Local  DAT    /#1, 2
```

[Sample output] -XFO[RM]=1

```
RA17K V1.00 V1 << D17XXX    XREF LIST >>  HH:MM:SS MM/DD/YY PAGE 01-001
PROG = SAMPLE
SOURCE = SAMPLE. ASM
MODULE(LineNo.)              : VALUE    : ATTR : TYPE : SYMBOL
B:\RA17K\SAMPLE. ASM(5) : 00000001 : Local  : MEM  : symbol-name
  B:\RA17K\SAMPLE. ASM(8)
  B:\RA17K\SAMPLE. ASM(12)
B:\RA17K\SAMPLE1.MLC(7)  : 00000000 : Local : STR : structured-symbol-name
  B:\RA17K\SAMPLE1.MLC(18)
  B:\RA17K\SAMPLE1.MLC(22)
```

(2) Public cross-references

The definitions and reference positions of those symbols referenced from external modules are listed.

(a) Option

Specify -PUB[XREF] to list public cross-references.

(b) Output file

Public cross-reference listing (.PUB)

(c) Output details

(i) -XFO[RM]=0

symbol-name evaluation-value attribute type [m] reference/definition-line-number (#definition-line-number)

(ii) -XFO[RM]=1

Definition line: source-module-name(nn):evaluation-value:attribute:type:symbol-name Reference line: source-module-name(nn)

The table below details the line format.

Legend	Output value	Explanation
Definition line number	nn	Indicates the number of the line on which a symbol is defined.
Reference line number		Indicates the number of the line from which the symbol is referenced
m	m	Indicates the module number (sequence number of the REL file specified at entry).
Evaluation value (VALUE)	32 bits (hexadecimal)	Refer to the RA17K User's Manual for details.
Attribute (ATTR)	Public	Symbol that can be referenced from other modules
	Local	Symbol that can be referenced only within the module in which it appears.
Type (TYPE)	DAT	Data type
	MEM	Memory type
	Nn	Horizontal nibble memory type
	NnV	Vertical nibble memory type
	LAB	Label type
	FLG	Flag type
	MAC	Macro
	STR	Structure
	UNI	Union
	ARY	Array

Remark Up to about 8,000 symbols can be contained within a cross-reference listing.

[Sample output] -XFO[RM]=0

```
RA17K V1.00 V1 << D17XXX    PUBXREF LIST >>  HH:MM:SS MM/DD/YY PAGE 01-001
PROG = SAMPLE
SOURCE = SAMPLE.ASM
   SYMBOL                    VALUE   ATTR TYPE     /REF(#DEF)
   Symbol name               00000001 Local DAT     /[1]# 1
```

[Sample output] -XFO[RM]=1

```
RA17K V1.00 V1 << D17XXX    PUBXREF LIST >>  HH:MM:SS MM/DD/YY PAGE 001
PROG = SAMPLE
MODULE(LineNo.)         : VALUE     : TYPE : SYMBOL
B:\RA17K\MAIN. ASM(5) : 00000001 : MEM  : PUBLIC-symbol-name
        B:\RA17K\SAMPLE. ASM(24)
        B:\RA17K\SUB. ASM(32)
B:\RA17K\SAMPLE1. MLC(7)  :  00000000 :  STR :  structured-symbol-name
        B:\RA17K\SAMPLE2. MLC(18)
        B:\RA17K\SAMPLE3. MLC(22)
```

4.4 Module Map and Program Map Listings

4.4.1 Map Listings

A map listing presents memory definitions as a table.

(1) Options

Specify -MAP to output a module map listing or -AMA[P] to output a program map listing.

(2) Output file

- Module map listing (.MAP)

  -MAP : REL filename.MAP

  -MAP=<filename> : <filename>.MAP
  If more than one module is named, a map listing is output for each of the named modules.

- Program map listing (.AMP)

  -AMA[P] : Parameter filename.AMP

  -AMA[P]=<filename> : <filename>.AMP
  A map listing in which the symbols of multiple modules are merged is output.

(3) Output details

A map listing consists of a bank-specific listing of memory attribute (MEM, NIBBLEn, and NIBBLEnV) symbols and flat attribute symbols. A map of the members of *emIC-17K* structures and unions is included.

Caution To output a structure and union map, specify a debug option at *emIC-17K* compile time. No structure or union map is output if this debug option is omitted.

(a) Delimiters

The kinds of lines used to represent the height, width, corners, intersections and other aspects of a memory map can be represented with specified characters. Delimiters may also be specified for public and local variables.

(b) Symbol name display

When a variable name is too large to be displayed in a specified box, a specified delimiter is displayed. For the NIBBLEn, NIBBLEnV, and STRUCTURE attributes, the variable name is padded with a specified character to enable the display of more than one box.

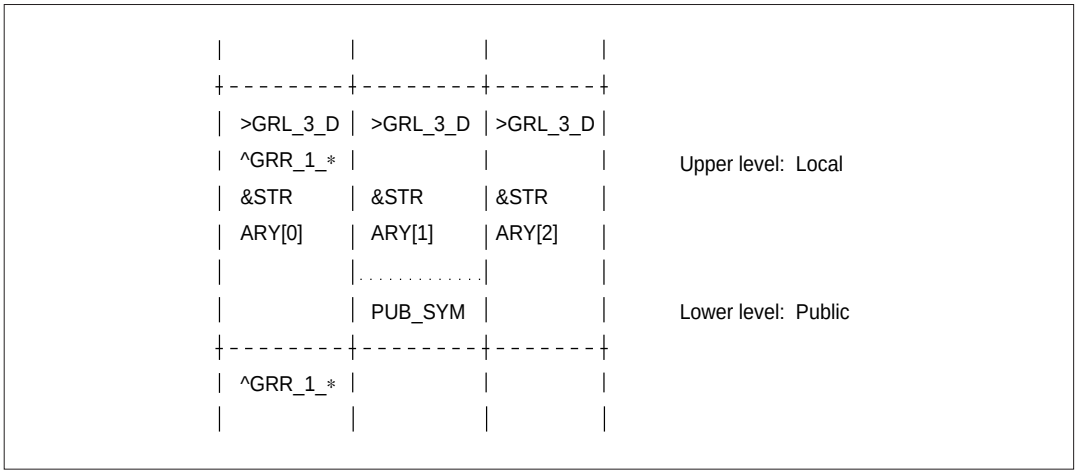
(c) Array display

Arrays are represented by the name [ ]....

Example 1 BYTE abc[2] [3]: Listing of six arrays
abc[0] [0] abc[0] [1] abc[0] [2] abc[1] [0] abc[1] [1] abc[1] [2]

Example 2 Array representation of the structure STRUCTURE
struct A a[2];
&a[0] &a[0] &a[1] &a[1]

Figure 4-1. Sample Memory Map Listing



Remark Double-byte lines, 8 characters, and delimiter "*".

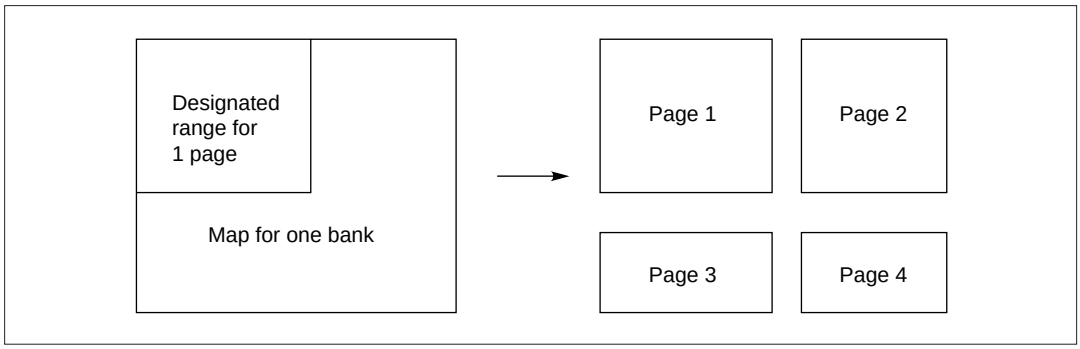
Caution While the number of symbols displayed in each box is unlimited, the execution of symbol processing requires that a work area be secured in the host machine's memory. If the available memory space is limited, it may prove impossible to display all the symbols. For a 17K Series device, do not assign more than 6,000 symbols per bank of data memory.

(d) Height and width as a number of characters

The height and width of a page are set to the option values of ROW (height) and COLUMN (width). A data memory map which exceeds the capacity of one page is continued on the next page.

If, however, a map is greater than two pages in length, that portion exceeding two pages is truncated.

Figure 4-2. Representation of a Data Memory Map Greater than One Page in Length



[Sample output 1]

RA17K V1.00 V1 << D17XXX DATA MEMORY MAP >> HH:MM:SS MM/DD/YY PAGE 01-001

PROG = SAMPLE

SOURCE = SAMPLE.ASM

BANK = 0

ROW	COL 0	COL 1	COL 2	COL 3	COL 4	COL 5	COL 6	COL 7
0	^GRL_1_D	LCDD0 >GRL_1_D	>GRL_1_D		DATA TOP		KEY_COU*	RG0
1	^GRL_2_D &struct	LCDD1 >GRL_2_D &struct	>GRL_2_D &struct	&struct	VOL_DAT &abc[0]	&abc[0]	&abc[1]	RG1 &abc[1]
2	^GRL_3_D	LCDD2 >GRL_3_D	>GRL_3_D		..... ^PNV_SYM	DELAY_W*		RG2
3	K_DAT	LCDD3	..... P_SYM	DELAY_FL	BAL_DAT ..... ^PNV_SYM	DLY_COU*		RG3
4	AR3	LCDD4		..... >PN_SYM	..... >PN_SYM ^PNV_SYM	DLY_IXL		^K_DAT_X
5	AR2	LCDD5 ^GRR_1_*	^GRL_7_D		FAD_DAT	LEVEL_D*	LCD_DA_3	^K_DAT_Y ^K_DAT_X
6	AR1	LCDD6 ^GRR_1_*	^GRL_7_D			DTM_IXM2 ^GRL_6_D	^GRL_5_D	^K_DAT_X ^K_DAT_Y
7	AR0	LCDD7 ^GRR_1_*	^GRL_7_D	DELAY_FR	SUR_M_D*	^GRL_6_D	^GRL_5_D	

Symbols with the NIBBLEn or NIBBLEnV attribute, for which multiple nibbles are defined, are output unconditionally. Symbols with the NIBBLEn type are preceded with ">". Those with the NIBBLEnV type are preceded with "^". Structured symbols are preceded with "&"; arrays are terminated by array-number[n]. Symbols exceeding the value of -NAMELENGTH are terminated by "*".

[Sample output 2]

```
RA17K V1.00 V1  << D17XXX FLAG MAP >>  HH:MM:SS MM/DD/YY PAGE 01-001
PROG = SAMPLE
SOURCE = SAMPLE.ASM

                                BANK = 0

     MSB       3         2         1         0       LSB
      ┼--------┼--------┼--------┼--------┼
   63 | LCD_DWN | LCD_UP |        |        |
      |         |        |........|        |
      |         |        | P_FLG1 |        |
      ┼--------┼--------┼--------┼--------┼
   66 |         |        |        |        |
      ┼--------┼--------┼--------┼--------┼
   68 |         |........|        |        |
      |         | P_FLG2 |        |        |
      ┼--------┼--------┼--------┼--------┼
   70 | P2A3    | P2A2   | P2A1   | P2A0   |
      ┼--------┼--------┼--------┼--------┼
```

[Sample output 3]

A sample listing of the structure definition shown below is presented on the following pages.

```
struct          puba
{
        BYTE    a1;
        BYTE    a2[2];
        struct  pubb
        {
                BYTE    b1;
                BYTE    b2[2];
        }       PubB1;
        struct  pubb    Pb2[2];
};
```

RA17K V1.00 V1 << D17XXX STRUCTURE MAP >> HH:MM:SS MM/DD/YY PAGE 01-001

PROG = SAMPLE

SOURCE = SAMPLE. MLC

STRUCT = pubb

	COL 0	1	2	3	4	5	6	7
R O W 0	^b1	^b1	^b2[0]	^b2[0]	^b2[1]	^b2[1]		

RA17K V1.00 V1 <<D17XXX STRUCTURE MAP>> HH:MM:SS MM/DD/YY PAGE 01-001

PROG = SAMPLE

SOURCE = SAMPLE.MLC

STRUCT = puba

	COL 0	1	2	3	4	5	6	7
R O W 0	^a1	^a1	^a2[0]	^a2[0]	^a2[1]	^a2[1]	&PubB1	&PubB1
1	&PubB1	&PubB1	&PubB1	&PubB1	&Pb2[0]	&Pb2[0]	&Pb2[0]	&Pb2[0]
2	&Pb2[0]	&Pb2[0]	&Pb2[1]	&Pb2[1]	&Pb2[1]	&Pb2[1]	&Pb2[1]	&Pb2[1]

4.4.2 Symbol Listing

The map listings are followed by a symbol listing that consists of memory and flag symbol information.

(1) Option

Specify -MAP to produce a module-specific symbol listing, or -AMA[P] to produce a symbol listing for an entire program.

(2) Output files

- Module map list (.MAP)

  -MAP : REL filename.MAP

  -MAP=<filename> : <filename>.MAP
  If more than one module is named, a map listing is produced for each of the named modules.

- Program map listing (.AMP)

  -AMA[P] : Parameter filename.AMP

  -AMA[P]=<filename> : <filename>.AMP
  A map listing consisting of symbols from multiple modules is produced.

(3) Output details

SYMBOL VALUE TYPE BIT INFORMATION

The table below details the line format.

Legend	Output value	Explanation
SYMBOL	Symbol name	Symbol name. When a symbol name exceeds the value of -SYMLENGTH, (one double-byte character or two single-byte characters), only the symbol name is output on the first line. All the information following VALUE is directed to the subsequent lines.
Evaluation value (VALUE)	32 bits (hexadecimal)	Refer to the RA17K User's Manual for details.
Type (TYPE)	DAT	Data type
	MEM	Memory type
	Nn	Horizontal nibble memory type
	NnV	Vertical nibble memory type
	LAB	Label type
	FLG	Flag type
	BIT	Bit position
	1	...1
	2	..1.
	4	.1..
	8	1...
	STR	Structure
	UNI	Union
	ARY	Array
INFORMATION	–	Contents of the comment on the symbol definition line

[Sample output]

```
RA17K V1.00 V1   << D17XXX    SYMBOL MAP >>   HH:MM:SS MM/DD/YY PAGE 01-001
PROG = SAMPLE
SOURCE = SAMPLE. ASM
SYMBOL                  VALUE    TYPE BIT  INFORMATION
ABC_MEM                 00000010  MEM       ; Public variable
WORK                    00000021  FLG  ...1  ; Local variable
ABC_NIB                 00001003  N2        ; NIBBLE 2
XYZ_NIBV                00009004  N2V       ; NIBBLE 2 V
STRUCT                  00000000  STR       ; Structure
UNION                   00000000  UNI       ; Union
ARRAY                   00000000  ARY       ; Array
```

4.5 Report Listing

A report listing provides a record of the status of assembly execution, including the assembly time spent, the status of memory utilization, and file information.

(1) Option

Specify -REP[ORT] to output a report listing.

(2) Output file

Report listing (.REP)

(3) Output details

(a) Run durations

The time at which assembly started, the time at which it ended, and the elapsed time. If more than one module is named, run durations are listed in sequence.

(b) Status of memory utilization

The memory utilization status for the assembler. If more than one module is named, the memory utilization status is output for each of the named modules.

(c) File information

The sizes of the files used by the assembler. If more than one module is named, file information is output for each of the named modules.

[Sample output]

```
RA17K V1.00 V1   << D17XXX    REPORT LIST >>   HH:MM:SS MM/DD/YY PAGE 001
PROG = SAMPLE
<ELAPSED TIME>
```

PROCESS	START	END	ELAPSED
ASSEMBLER(TEST1. ASM)	13:02:05	13:03:17	00:01:12
ASSEMBLER(TEST2. ASM)	13:04:00	13:05:17	00:01:17
COMPILER(SAMPLE3. MLC)	13:05:20	13:06:03	00:00:43
COMPILER(SAMPLE4. MLC)	13:06:04	13:06:17	00:00:13
		TOTAL	00:03:25

RA17K V1.00 V1 << D17XXX REPORT LIST >> 10:34:40 04/22/93 PAGE 002

PROG = SAMPLE

SOURCE = TEST1. ASM

< USED MEMORY >

MEMORY BLOCKS		SIZE	USED	PERCENT
REL	TMP	4317	259	6.0
REL	TMP	2269	669	29.1
REL	TMP	4317	515	11.9
REL	FILE	4341	1145	26.4
INTERM	TMP	8413	242	2.9
XREF	FILE	8437	245	2.9
SYMBOL	MAIN	16384	16384	100.0
SYMBOL	EMS	0	0	0.0
LIST	TMP	4341	393	9.1
ETC		5223	520	10.0

< USED FILES >

FILE NAMES	SIZE
C:\RA17K\SYSTEM\REP17K\TEST1. REL	900
C:\RA17K\SYSTEM\REP17K\TEST1. IXF	0
C:\RA17K\SYSTEM\REP17K\TEST1. IPN	148

(4) Coding a format definition file

A format definition file can be specified with the -RFMT option to output a report listing in a user-specified format.

A format definition file is coded by preceding each line with a line control character. Normal lines are marked by ”N,” the last line by ”P.” The last line is mandatory.

Item coding formats are described below.

(a) %D#item-name [,number-of-characters]%

Item-name may be either a reserved item name or a user-defined item name. If number-of-characters is omitted, the item is output using the number of characters of item-name.

If one item name is specified per line when number-of-characters is omitted, the column length can be adjusted by coding "%S%". If more than one item name is specified per line, however, the column length cannot be adjusted.

Each occurrence of "%D#item-name" in the intermediate PRN file is replaced with the specified item upon detection.

(b) %S%

Code "%S%" to adjust the column length. Pad each line with blanks to align the last character with the end of the line.

(c) %R-character%

Code "%R-character%" to specify a repetition character. The character immediately after %R%" is output repeatedly with column length adjustment.

(d) Other characters

All other characters and lines are output as is.

[Sample coding]

	Definition		Output	
N	%D#TITLE, 8%	→	8 columns	Adjust the column length
N	%D#TITLE%%S%	→	Number of characters for an item	Adjust the column length with %S%
N	%R—%	→		

Caution **When a fixed character and an item appear on the same line of the format definition file, the fixed character is output repeatedly, the specified number of items. To output a fixed character once only, code it on a different line.**

[Sample coding for a format definition file]

```
$PROCESSFORM
N< ELAPSED TIME >
N
N ┌─%R─%─┬──────────┬──────────┬──────────┐
N │ PROCESS%S% │  START   │   END    │ ELAPSED │
N ├─%R─%─┼──────────┼──────────┼──────────┤
N │ %D#PROC%%S% │ %D#START, 8% │ %D#FINISH, 8% │ %D#ELAPSE, 8% │
N └─%R─%─┴──────────┴──────────┴──────────┘
N%S% │   TOTAL    │ %D#TOTAL, 8% │
P%S% └────────────┴──────────────┘
$MEMORYFORM
N< USED MEMORY >
N
N ┌─%R─%─┬──────────┬──────────┬──────────┐
N │ MEMORY BLOCKS%S% │   SIZE   │   USED   │ PERCENT │
N ├─%R─%─┼──────────┼──────────┼──────────┤
N │ %D#MEM%%S% │ %D#MSIZE, 8% │ %D#MUSED, 8% │ %D#PER, 8% │
P └─%R─%─┴──────────┴──────────┴──────────┘
$FILESFORM
N
N< USED FILES >
N
N ┌─%R─%─┬──────────┐
N │ FILE NAMES%S% │   SIZE   │
N ├─%R─%─┼──────────┤
N │ %D#FILE%%S% │ %D#FSIZE, 8% │
P └─%R─%─┴──────────┘
```

4.6 Document Listing

4.6.1 Document Listing

A document listing is a listing output according to a format definition, on the basis of the -SUM[MARY] and module information.

The general format of a document listing is shown below.

(1) Contents

(2) Program Specifications
  (a) Program summary
  (b) Module summary
  (c) Routine summary

(1) Option

Specify -SPE[C] to output a document listing.

(2) Output file

Document listing (.SPC)

(3) Contents

(a) Contents

The title assigned to each block is output as a table of contents. Each block can be divided into units such as chapters, sections, items, and subitems.

Table 4-1. Relationship Between Components of Contents and Program Units

Component	Corresponding program unit
Chapter	Module file
Section	Routine (function)
Item	Tag
Subitem	Tag (hierarchical level)

A typical contents presentation format is as follows:

Chapter : Chapter 1, Chapter 2, ..., Chapter n

Section : 1.1. 1.2, ..., n.n

Item level 1 : 1.1.1, 1.1.2, ..., n.n.n

Item level 2 : (1), (2) ..., (n)

Item level 3 : (a), (b), ... (z)

Item level 4 : **<1>**, **<2>**, ..., **<20>**

[Sample output]

<Header statement> The #toc character string specified in the format definition file is output.

Chapter 1 Module Title 1

1.1 Section Title 2

1.1.1 Routine Title 3

(1) Tag title 1 3

(2) Tag title 2 3

(3) Tag title 3 3

(b) Program Specifications

Program specification text, directed by a format definition, is created by program, module, and routine according to the -SUM[MARY] and module information.

(i) Program summary

The contents of the file specified by -SUM[MARY] are output as a program summary as is. The file contents can be coded freely at the user desires.

[Sample coding]

- **Option**

```
-SPEC -SUMMARY='Program Title',SUM.TXT
```

- **Contents of SUM.TXT**

```
Program name:  SAMPLE.EXE
    Program summary file.  The information contained in this file is
    output as a program file.
```

[Program summary: Sample output]

Program Title

```
Program name:  SAMPLE.EXE
    Program summary file.  The information contained in this file is output
    as a program file.
```

(ii) Module summary

The information written in the module block is edited and output according to the format definition. The output format is the same for both RA17K and *emIC-17K*.

A document listing, output as explained in **[Sample coding of a format definition file]** in **Section 4.6.3(3)**, is given on the following pages.

[Module summary: Sample output]

RA17K V1.00 V1 << D17XXX DOCUMENT LIST >> HH:MM:SS MM/DD/YY PAGE 001

PROG = SAMPLE

Chapter 1. Initial Institution	
[specifications] Module Summary	
Module Name	: INITIAL. ASM
Create	: 92/07/01 13:12:11
Size	: 4096
Step	: 20
PUBLIC (MEM)	: PUB_MEM
PUBLIC (FLG)	: PUB_FLG
PUBLIC (DAT)	: PUB_DAT
PUBLIC (LAB)	: PUB_LAB
EXTRN (MEM)	: EXT_MEM
EXTRN (FLG)	: EXT_FLG
EXTRN (DAT)	: EXT_DAT
EXTRN (LAB)	: EXT_LAB

(iii) Routine summary

The information written in the routine block is edited and output according to the format definition. The output format is the same for both RA17K and *emIC-17K*.

For RA17K, however, input/output information is not output.

[Routine summary: Sample output]

RA17K V1.00 V1 <<D17XXX DOCUMENT LIST>> HH:MM:SS MM/DD/YY PAGE 002

PROG = SAMPLE

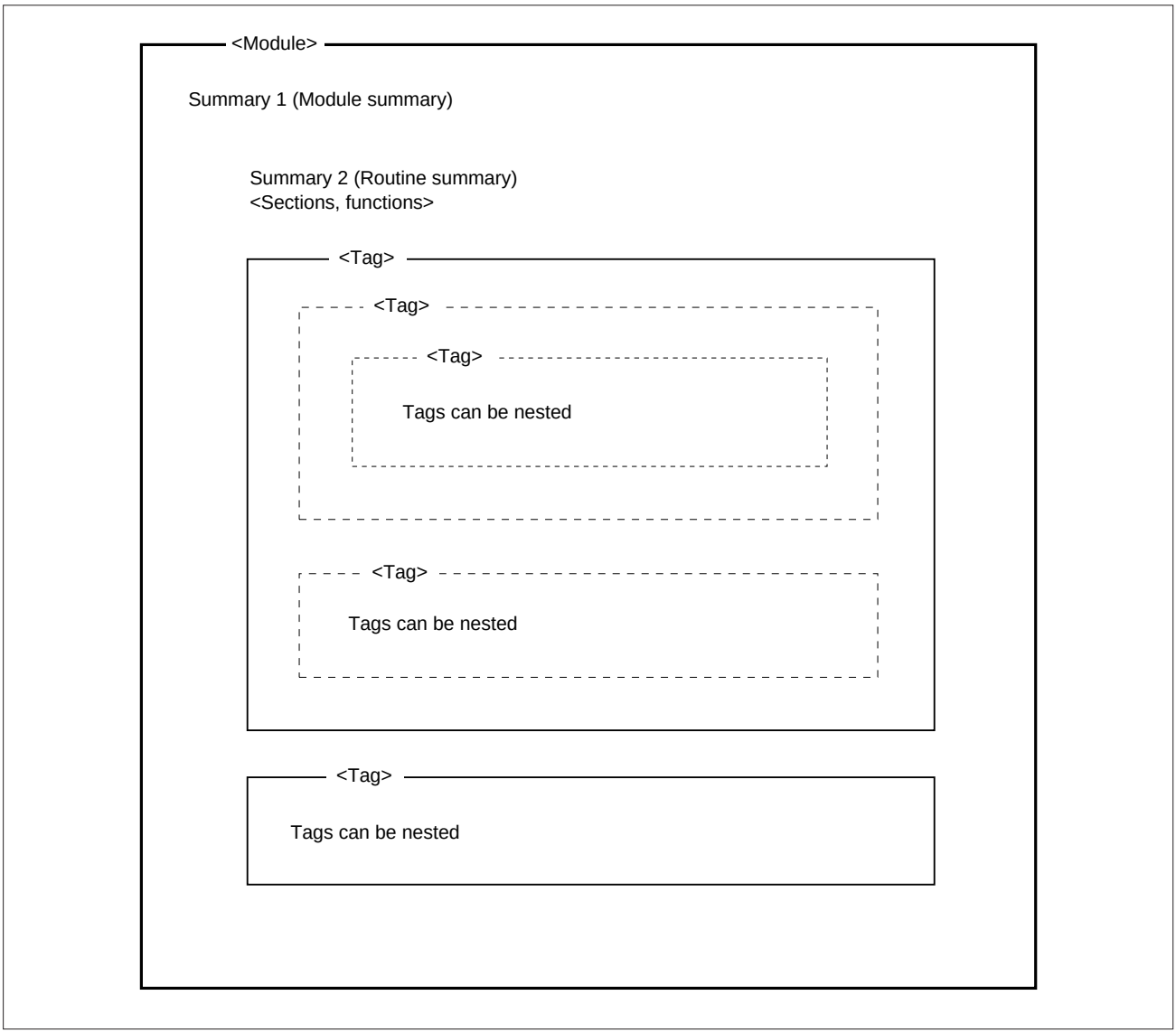
2.1 Volume Control		
[specifications] Routine Summary		
[Routine Information]		
Input	:	nibble a
OutPut	:	dword b
Address Range	:	0 - 1F
EPA Address Range	:	10020 - 1002F
JumpT Address Range	:	0 - 0
Entrances	:	INIT
MEM Changed	:	M_CHG
MEM Referred	:	M_REF
MEM Manipulatd	:	
FLG Changed	:	F_CHG
FLG Referred	:	F_REF
DAT Referred	:	
Branch To	:	
Subroutines Called	:	D17005_INIT
LAB Manipulated	:	
System Call	:	
2.1.1 MODE_KEY Delay change Delay key Delay institution		

4.6.2 Summary Rules

The summary rules described in each module (source file) are presented. A comment symbol is not needed between the SUMMARY and ENDSUM statements. #pragma summary is presented in a comment format. Summaries cannot be nested by themselves.

Summary statements appearing in tags are directed to the tag table of the program specifications.

Figure 4-3. Summary Layout



(1) Program summary

(a) Title

The information specified with -SUM[MARY].

(b) Summary statement

Defined in the separate file for program summary statements, as named by -SUM[MARY]. See **Section 4.6.1(3)(b)(i)**.

(2) Module summary

(a) Title

The title of a module summary is specified by pseudo instructions SUMMARY to ENDSUM (#pragma summary to #pragma end_summary) at the beginning of the corresponding module. The module filename is assumed as the title when no title is explicitly specified.

(b) Summary statement

A summary statement is specified by pseudo instructions SUMMARY to ENDSUM (#pragma summary to #pragma end_summary) at the beginning of the corresponding module.

(3) Routine summary (section or function)

(a) Title

The title of a routine summary is specified by pseudo instructions SUMMARY to ENDSUM (#pragma summary to #pragma end_summary) that precede the section or function definition statement. The section or function name is assumed as a title when no title is explicitly specified.

(b) Summary statement

A summary statement is specified by pseudo instructions SUMMARY to ENDSUM (#pragma summary to #pragma end_summary) immediately before the section or function definition statement.

(4) Tag

(a) Title

Tag titles are specified with ";." or ";.V" for RA17K, or with "//." for *emIC-17K*. This hierarchical level is terminated by ";.." or "//..". Titles can be nested.

The character strings (. and ..), following a comment symbol, can be altered by using RA17K options -TAGS[TART] and -TAGE[ND].

(b) Summary statement

Any comment statement appearing after a tag line is handled as a summary statement. A summary statement is terminated by a non-comment line.

[Sample summary coding 1: RA17K]

```
;  Sample Summary
SUMMARY 'Module name title'
   Provides a summary of the module (may be omitted).
ENDSUM
   SUMMARY 'Routine title'
        Provides a summary of the routine (may be omitted).
ENDSUM
PROG1 CSEG
<Program>
;.  Tag title
;   The comment following the tag is assumed to be a summary statement
;   The summary statement is terminated by other than a comment statement

;.  V
;   The immediately subsequent label is registered as a tag
;   A comment following a tag is regarded as being a summary statement
;   The summary statement is terminated by other than a comment statement
LABEL1:
<Program>
SUMMARY <- The section name is assumed as the title when no title is explicitly specified.
        Provides a summary of the routine (may be omitted).
ENDSUM
PROG2 CSEG
<Program>
END
```

[Sample summary coding 2: *emIC-17K*]

```
// Sample Summary
#pragma summary   'Module name title'
//      Provides a summary of the module (may be omitted).
#pragma end_summary
#pragma summary   'Routine title'
//      Provides a summary of the routine (may be omitted).
#pragma end_summary
void  prog1(void)
{
      < Program >
      //.   Tag title 1
      //    A comment following the tag is assumed to be a summary statement
      //    The summary statement is terminated by other than a comment statement
      < Program >
      //.   Tag title 2
      //    Tags may be nested
      //..
      //..
      < Program >
}
#pragma summary <– The function name is assumed as the title when no title is explicitly specified.
//      Provides a summary of the routine (may be omitted).
#pragma end_summary
void  prog2(void)
{
      < Program >
}
```

4.6.3 Format Definition

A format definition file can be specified with the -DFMT option to produce a document listing in a user-specified format. The document listing utility produces a document file according to the information defined in the format definition file.

The format definition file contains a definition of the item names and item functions, the output file format and so on. Users can perform customization by modifying the format definition file.

Because item name replacement begins at character position "%", a sufficiently large space must be allowed between occurrences of %, or between % and any other delimiter (such as a line). When a replacement character string is too long to be inserted, the excess characters are truncated. If more than one replacement character string is specified, the strings are output at the same position on the next line once the item names have been exhausted for the current line.

(1) Format definition file format

The table below describes the coding format for the format definition file.

#item	<item-name>:<alias>:<summary-level>
#row	<number-of-lines-per-page>
#column	<number-of-columns>
#toc	<contents>
#idn	<item-number-format> n: level number**(Note)**
#moduleform	Module specification format
#routineform	Routine specification format

(Note) Item number formats are as follows:

1 = Chapter n	(double-byte characters)	7 = (n)	(single-byte characters)
2 = n	(double-byte characters)	8 = (a)	(double-byte letters)
3 = n	(single-byte characters)	9 = (a)	(single-byte letters)
4 = n. n ...	(double-byte characters)	10 = **<1>** to **<20>**	(double-byte characters>
5 = n.n ...	(single-byte characters)	11 = .	(double-byte characters)
6 = (n)	(double-byte characters)	12 = .	(single-byte characters)

(2) Item name

The contents of the item specified by an item name in the format definition file are retrieved from the source module to replace each occurrence of the item name.

(a) Reserved item name

The table below lists reserved item names.

Table 4-2. Reserved Item Names in the Format Definition File

Item name	Explanation	Item name	Explanation
FileName	Module filename	JMPTEndAddr	Branch table ending address
FileSize	Module file size	RoutineName	Routine name
FileDate	File creation date	Public	Public symbol
FileTime	File creation time	Extrn	External symbol
Title	Title	Entry	Entry point
Step	Number of steps in the module	MemRef	Memory reference
Input	Function input interface	MemChg	Memory change
Output	Function output interface	FlgRef	Flag reference
TopAddr	Starting address	FlgChg	Flag change
EndAddr	Ending address	LabRef	Label reference
EPATopAddr	EPA starting address	LabChg	Label change
EPAEndAddr	EPA ending address	Call	Routine call
JMPTTopAddr	Branch table starting address	Br	Branch

(b) User-defined item name

Users can register user-defined items. When an item name is omitted, the item contents for the preceding item name are assumed. Item contents having the same item name are linked together.

Continuation of the item contents terminates when the next item name or a blank line is encountered.

[Item name sample coding: RA17K]

```
;     <item-name>     :     <item-contents>
;                     :     <continuation-of-item-contents>
;
```

[Item name sample coding: *emIC-17K*]

```
//    <item-name>     :     <item-contents>
//                    :     <continuation-of-item-contents>
//
```

[Examples of user-defined item names]

User-defined item name	Item contents	Coding position in a source module
System	System (program) description	Module summary
Program	Program name	
Module	Module description	
Revision	Revision number	
Manager	Manager name	
Routine	Routine description	Routine summary
Spec	Routine specifications	
Author	Author	
Note	Memo	

(3) Coding a format definition file

A line that begins with ";" is a comment line. Comment lines can be specified anywhere other than in moduleform and routineform.

A format definition file containing a specification of user-defined item names, the number of lines per page in the document list, the number of columns, the character string in the contents, and so on.

Unless otherwise specified, the values listed below are assumed.

#item	No user-defined item
#row	Value specified by -ROW, or the default
#column	Value specified by -COL[UMN], or the default
#toc	Character string "Contents"
#id1	Item format 1 (Chapter 1)

The format definition file then proceeds with moduleform and routineform, which are coded the same way. A format definition file is coded by preceding each line with a line control character. Normal lines are marked by "N," the last line by "P." The last line is mandatory.

Item coding formats are described below.

(a) %D#item-name[,number-of-characters]%

Item-name may be either a reserved item name or a user-defined item name. If number-of-characters is omitted, the number of characters of item-name is assumed.

If one item name is specified per line when number-of-characters is omitted, the column length can be adjusted by coding "%S%". If more than one item name is specified per line, however, the column length is not adjustable.

Each occurrence of "%D#item-name" in the intermediate PRN file is replaced with the specified item as it is encountered.

(b) %S%

Code "%S%" to adjust the column length. Pad each line with blanks to align the last character with the end of the line.

(c) %R-character%

Code "%R-character%" to specify a repetition character. The character following %R%" is output repeatedly with column length adjustment.

(e) Other characters

All other characters and lines are output as is.

[Sample coding]

	Definition		Output	
N	%D#TITLE, 8%	→	8 columns	Adjust the column length
N	%D#TITLE%%S%	→	Number of characters for an item	Adjust the column length with %S%
N	%R—%	→		

Caution **When a fixed character and an item appear on the same line of the form definition file, the fixed character is output as many times as the number of items specified. To output a fixed character once only, code it on a different line.**

[Sample coding for a format definition file]

```
#item
#row=66
#column=250
#toc= 'Contents'
#id1=1                    ; Chapter 1
#id2=4                    ; 1. 1
#id3=4                    ; 1. 1. 1
#id4=6                    ; ( 1 )
#id5=8                    ; ( a )
#id6=10                   ; <1>
$moduleform
N ┌─%R—% ┐
N │ %D#ID1, 10% %D#TITLE, 10%%S% │
N ├─%R—%─┤
N │ [Specifications]%S% │
N │  %D#Contents%%S% │
N ├─%R—%─┤
N │ Module Name       : %D#FileName%%S% │
N │ Create            : %D#FileDate% %D#FileTime%%S% │
N │ Size              : %D#Size%%S% │
N │ Step              : %D#Step%%S% │
N ├─%R—%─┤
N │ PUBLIC(MEM)       : %D#PubMem%%S% │
N │ PUBLIC(FLG)       : %D#PubFlg%%S% │
N │ PUBLIC(DAT)       : %D#PubDat%%S% │
N │ PUBLIC(LAB)       : %D#PubLab%%S% │
N │ EXTRN(MEN)        : %D#ExtMem%%S% │
N │ EXTRN(FLG)        : %D#ExtFlg%%S% │
N │ EXTRN(DAT)        : %D#ExtDat%%S% │
N │ EXTRN(LAB)        : %D#ExtLab%%S% │
P └─%R—%─┘
```

```
$routineform
N ┌%R—%┐
N │%D#ID2, 10%  %D#ROUTINE, 10%%S%│
N ├%R—%┤
N │[Specifications]%S%│
N │ %D#Contents%%S%│
N ├%R—%┤
N │[Routine Information]%S%│
N │Input                  : %D#Input%%S% │
N │Output                 : %D#Output%%S% │
N │Address Range          : %D#TopAddr% - %D#EndAddr%%S% │
N │EPA Address Range      : %D#EPATopAddr% - %D#EPAEndAddr%%S% │
N │JumpT Address Range : %D#JMPTTopAddr% - %D#JMPTEndAddr%%S% │
N │Entrances              : %D#Entry%%S%      │
N │MEM Changed            : %D#MemChg%%S% │
N │MEM Referred           : %D#MemRef%%S% │
N │MEM Manipulated        : %D#MemMan%%S% │
N │FLG Changed            : %D#FlgChg%%S% │
N │FLG Referred           : %D#FlgRef%%S% │
N │DAT Referred           : %D#DatRef%%S% │
N │Branch To              : %D#Br%%S% │
N │Subroutines Called     : %D#SubCall%%S% │
N │LAB Manipulated        : %D#LabMan%%S% │
N │System Call            : %D#SysCall%%S% │
N ├%R— %┤
N │     %D#ID2%  %D#TAG%%S%│
P └%R—% ┘
```

4.7 Call Tree Listing

A call tree listing consists of a hierarchical listing of modules/routines (call tree format).

The characters following the function names have the following meanings:

? : Undefined routine (undefined external reference)

... : Tree already displayed

[n] : n in [n] indicates the ID number with which the tree is displayed.

Function names that are defined but not referenced are labeled <UNREFERENCE>.

Caution **A call tree listing is produced only for the *emIC-17K* source program.**

[Sample output]

```
MLC17K V1.00 V1   << D17XXX    TREE LIST >>   HH:MM:SS  MM/DD/YY PAGE 001

PROG = SAMPLE


 main
 |1
 +--initialize_table
 |    |1
 |    +--clear_buffer
 |    |2
 |    +--set_table
 |2
 +--initialize_key
 |    |1
 |    +--key_xxxx ?
 |    |2
 |    +--key_yyyy
 |3
 +--display
 |    |1
 |    +--menu
 |    |    |1
 |    |    +--function_one
 |    |    |    |1
 |    |    |    +--set_func [3.1.1.1]
 |    |    |         |1
 |    |    |         +--exec_key
 |    |    |2
 |    |    +--function_two
 |    |    |    |1
 |    |    |    +--set_func. . . [3.1.1.1]
 |    |    |3
 |    |    +--function_three
 |    |    |    |1
 |    |    |    +--set_func. . . [3.1.1.1]
 |    |    |    |2
 |    |    |    +--modify_mode
```

```
MLC17K V1.00 V1   << D17XXX    TREE LIST >>   HH:MM:SS  MM/DD/YY PAGE 002
PROG = SAMPLE

    |    |    | 4
    |    |    +--function_four
    |    |    |    | 1
    |    |    |    +--set_func. . . [3.1.1.1]
    |    |    |    | 2
    |    |    |    +--change_case
    |    |    |         | 1
    |    |    |         +--attriv
    |    |    |              | 1
    |    |    |              +--convert_mode
    |    |    | 5
    |    |    +--function_end
    |    |         | 1
    |    |         +--set_func. . . [3.1.1.1]
    |    | 2
    |    +--perform
    | 4
    +--exit
```

```
MLC17K V1.00 V1   << D17XXX    TREE LIST >>   HH:MM:SS  MM/DD/YY PAGE 003
PROG = SAMPLE
< UNREFERENCE >

  unref_func1
  unref_func2
```

4.8 Module Listing

A module listing is a routine list specific to a module file. Its output format is specified in a format definition file.

Caution **When source programs have been assembled using RA17K, a section name is output for each routine name field.**

(1) Option

Specify -MOD[ULE] to output a module listing.

(2) Output file

Module listing (.MOD)

(3) Output details

A sample listing is shown below.

(a) Module filename, creation date and time, and file size (includes the filename, creation date and time, and file size)

(b) Routine name, number of source steps, and address range

The address range includes an EP area (headed by @), and a branch table (headed by ?) if any.

[Sample module listing]

```
MLC17K V1.00 V1   << D17XXX    MODULE LIST >>   HH:MM:SS  MM/DD/YY PAGE 01-001
PROG = SAMPLE
SOURCE = SAMPLE1.MLC

+-----------------------------+---------------------+-----------+
| MODULE NAME                 | CREATED DATE/TIME   | FILE SIZE |
+-----------------------------+---------------------+-----------+
|                             |                     |           |
| c:program\source\main.mlc   | 1991/12/16  10:00:00|      1234 |
|                             |                     |           |
|   c:\mll\include\stdio. h   | 1991/10/30  00:00:00|       324 |
|   c:\mll\devfile\dl7006. h  | 1991/11/01  00:00:00|       549 |
|   c:program\source\sub      | 1991/12/16  12:00:00|      1021 |
|                             |                     |           |
+-----------------------------+---------------------+-----------+

+-----------------------------------+----------+---------------+
| ROUTINE NAME                      | SRC STEPS| ADDRESS RANGE |
+-----------------------------------+----------+---------------+
|                                   |          |               |
| initialize_table (void)           |       33 | 00100 - 00123 |
| static clear_buffer (nibble *buffer)|     54 | 00124 - 00129 |
| static set_table (void)           |      109 | 0012A - 0014F |
| @_set_table                       |        - | 10150 - 1015F |
| ?_set_table                       |        - | 00000 - 00000 |
+-----------------------------------+----------+---------------+
```

```
RA17K V1.00 V1   << D17XXX    MODULE LIST >>   HH:MM:SS MM/DD/YY PAGE 02-001
PROG = SAMPLE
SOURCE = SAMPLE2. ASM

+-----------------------------+---------------------+-----------+
| MODULE NAME                 | CREATED DATE/TIME   | FILE SIZE |
+-----------------------------+---------------------+-----------+
|                             |                     |           |
|  c:program\source\keyfunc. asm |  1991/12/16 10:03:00 |  1932  |
|                             |                     |           |
|    c:\include\keyfunc1.asm  |  1991/10/30 00:00:00 |   324  |
|    c:\include\keyfunc2. asm |  1991/11/01 00:00:00 |   549  |
|                             |                     |           |
+-----------------------------+-------+----------+---+-----------+
| ROUTINE NAME                        | SRC STEPS1| ADDRESS RANGE |
+-------------------------------------+----------+---------------+
|                                     |          |               |
|  initialize_key                     |    133   | 00150 - 00153 |
|  clear_key_buffer                   |     98   | 00154 - 00179 |
|  set_key_table                      |    202   | 0017A - 0017F |
|                                     |          |               |
+-------------------------------------+----------+---------------+
```

4.9 Address Map

An address map file contains routines (functions) and variables assigned in ROM. An address map is output page by page (PAGE0 to PAGE3).

Caution **When source programs have been assembled using RA17K, a section name is output in each routine name field.**

(1) Option

Specify -PMA[P] to produce an address map.

(2) Output file

Module file (.MOD)

(3) Output details

A sample listing is shown below.

[Sample address map]

RA17K V1.00 V1 <<D17XXX ADDRESS MAP>> HH:MM:SS MM/DD/YY PAGE 001

PROG = SAMPLE

SEGMENT NO = x

	ROUTINE NAME/VARIABLE	MODULE NAME	ADDRESS
PAGE0	?_initialize	main. asm	00000 - 00000
	initialize	main. asm	00100 - 00123
	@_initialize	main. asm	10124 - 1012F
PAGE1	clear	sub . asm	00800 - 0012F
PAGE2	input	sub2. asm	01000 - 0130E
PAGE3	output	sub3. asm	01800 - 0004F

[MEMO]

Chapter 5

Error Messages

5.1 Error Handling

If an error is detected in any of the options specified in the execution of a listing utility, control immediately returns to DOS. The listing utility is not run. Correct the option before attempting to rerun the utility.

If an error is detected while a listing utility is running, control again returns to the calling process by canceling the execution of the listing utility.

If the parent process is DOS, the DOS prompt is displayed to prompt the user to enter a command. If the parent process is DOC17K, the listing utility having the next highest priority is run. Control returns to DOS if there is no other listing utility to run.

Once the error has been corrected, the listing utility may be rerun to output the desired listing.

5.2 Error Messages

Error messages are output upon the detection of an error during the execution of a listing utility. Table 5-1 lists the error messages that may be output.

[Message format]

```
error CNNN:  error-message 'error-information'
```

Remarks 1. C designates an error code.
A: Abort error F: Fatal error W: Warning error
2. NNN: Error number (three decimal digits)

[Sample output] Error found in an option

```
error A003:  Invalid option 'option-name'
```

Table 5-1. Error Messages (1/5)

Message number	Explanation	
A003	Message	Invalid option
	Cause	There is an error in the option.
	Action	Correct the option.
A004	Message	No Input File
	Cause	An input filename is not specified.
	Action	Specify a REL filename.
A005	Message	Too many files
	Cause	More than one input filename has been specified.
	Action	Specify only one input filename (REL file or parameter file).
A006	Message	Couldn't set SIGINT
	Cause	The CTRL + C interrupt handler cannot be set.
	Action	Rerun.
A007	Message	Illegal character
	Cause	The filename contains an illegal character.
	Action	Specify a valid filename.
A008	Message	No LNK option
	Cause	The -LNK option is not specified.
	Action	Specify the -LNK option.
A009	Message	Invalid file format
	Cause	The format of the option file to be passed to the process is invalid.
	Action	Delete the ".$$$" files and rerun.
A010	Message	Invalid REL format
	Cause	The format of the REL file is invalid.
	Action	Reassemble, or recompile and relink.
A011	Message	Invalid LNK format
	Cause	The LNK file has an invalid format.
	Action	Reassemble, or recompile and relink.

Table 5-1. Error Messages (2/5)

Message number	Explanation	
A012	Message	Not enough memory
	Cause	The size of the work area is insufficient.
	Action	Create a sufficiently large work area.
A013	Message	No section/function name
	Cause	The section or function name is not defined in the LNK file.
	Action	Reassemble, or recompile and relink.
A014	Message	No object code
	Cause	The object code is not defined in the LNK file
	Action	Reassemble, or recompile and relink.
A015	Message	Invalid XREF data
	Cause	There is no correspondence between the XREF data and the REF file.
	Action	Reassemble, or recompile and relink.
A016	Message	Invalid Input file extension
	Cause	The input filename does not have the REL extension.
	Action	Specify a REL file as an input file.
A017	Message	No get section addr
	Cause	The section address cannot be obtained from the LNK file.
	Action	Reassemble, or recompile and relink.
A018	Message	No get section name
	Cause	The section name cannot be obtained from the REL file in the ABS mode.
	Action	Reassemble, or recompile and relink.
A019	Message	Source file name not found in REL
	Cause	The source filename cannot be found in the REL file.
	Action	Reassemble, or recompile and relink.
A020	Message	List file name not found in REL
	Cause	The list filename cannot be found in the REL file.
	Action	Specify the list output option and reassemble, or recompile and relink.

Table 5-1. Error Messages (3/5)

Message number	Explanation	
A021	Message	XREF file name not found in REL
	Cause	The XREF filename cannot be found in the REF file.
	Action	Specify the XREF output option, then reassemble, or recompile and relink.
A022	Message	Summary file name not found in REL
	Cause	The summary filename cannot be found in the REL file.
	Action	Specify the summary output option, then reassemble, or recompile and relink.
A023**(Note 1)**	Message	Invalid Source Number
	Cause	The source line number cannot be found in the intermediate PRN file.
	Action	Reassemble, or recompile and relink.
A024**(Note 2)**	Message	Invalid Nest Number
	Cause	The expansion line number cannot be found in the intermediate PRN file.
	Action	Reassemble, or recompile and relink.
A025	Message	Internal error (xxx)
	Cause	An error was detected during internal processing.
	Action	Contact your dealer or NEC.
A026	Message	Invalid symbol number
	Cause	The symbol number is invalid.
	Action	Reassemble, or recompile and relink.
A027	Message	Format data item error
	Cause	There is an error in a data item name specified in the format definition file.
	Action	Correct the data item name specified in the format definition file.

(Note 1) W023 for MAP.EXE and AMAP.EXE.
(Note 2) W024 for MAP.EXE and AMAP.EXE.

Table 5-1. Error Messages (4/5)

Message number	Explanation	
A028	Message	Format data error
	Cause	There is an invalid entry in the specified format definition file. • A character other than ”N,” ”P,” or ”;” was encountered at the beginning of a line. • The correspondence of ’%’ is unmatched. • An invalid format control code was found. • An extra or lacking entry is found between separate occurrences of %. • The column length exceeds the default, or the value specified as an option. • There is more than one column length adjustment on one line. • The last line specification is missing. • An odd number of columns must be adjusted when a double-byte character is specified as a repeat character.
	Action	Correct the format definition file.
A029	Message	Invalid line format
	Cause	The coding format of the title character string is invalid.
	Action	Correct the format of the title character string.
A030	Message	Invalid command format
	Cause	The coding format of the output format specification command is invalid.
	Action	Correct the command format.
A031	Message	Not Found Line Number
	Cause	A line number specified in the summary file cannot be found in the list file.
	Action	Reassemble, or recompile and relink.
A032	Message	Log file name not found in REL
	Cause	The log filename is not found in the REL file.
	Action	Reassemble, or recompile and relink.
A033	Message	Invalid Log file format
	Cause	The format of the log file is invalid.
	Action	Reassemble, or recompile and relink.
A034	Message	Invalid summary file format
	Cause	The format of the summary file is invalid.
	Action	Reassemble, or recompile and relink.

Table 5-1. Error Messages (5/5)

Message number	Explanation	
A035	Message	Invalid symbol file format
	Cause	The format of the symbol file is invalid.
	Action	Reassemble, or recompile and relink.
A036	Message	Invalid PPL file format
	Cause	The format of the preprocessing file (.PPL) is invalid.
	Action	Reassemble, or recompile and relink.
A037	Message	Function multi defined
	Cause	A function name is defined more than once.
	Action	Correct the function name then rerun.
A038	Message	Function nest overflow
	Cause	The function call nesting level exceeds 40.
	Action	Reduce the function call nesting level to 40 or less.
A039	Message	Symbol file name not found in REL
	Cause	The symbol filename cannot be found in the REL file.
	Action	Reassemble, or recompile and relink.
A040	Message	Option not implemented
	Cause	An illegal option has been specified.
	Action	Remove the illegal option.

5.3 System Error Messages

In addition to the error messages covered in the previous section, system error messages may also be output. A system error message might be output, for example, if a file access request is issued to the OS or if an error is posted from the OS.

[Message format]

```
Filename:  system-error-message
```

[Sample output] A file cannot be found.

SAMPLE.REL: A file or directory cannot be found