Quelo™

68000 ASSEMBLER PACKAGE OVERVIEW MANUAL

Copyright

Disclaimer

Quelo makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, Quelo reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Quelo to notify any person of such revision or changes.

Trademarks

CP/M is a registered trademark of Digital Research. PC-DOS is a registered trademark of IBM.

TABLE OF CONTENTS

68000 Assembler Package Overview Manual

Contents.

Section 1. User License, Warranty and Support.

Section 2. Package Contents.

Section 3. Program Invocation Command Lines.

SECTION 1
User License, Warranty and Support

License:

The Quelo 68000 Assembler Package (PRODUCT) is licensed for single use. This would ordinarily be for use on a single machine, but may be used by a single person on more than one machine (e.g. the software consultant who may need to work in various locations). Multiple use requires the purchase of multiple copies or a multiple use license in the case of a multiuser system.

Copies of the software may be made for backup purposes, but not for any kind of distribution. The documentation may NOT be copied.

Use of the software implies acceptance of these license terms.

Warranty and Disclaimer:

The complexity of this PRODUCT makes it impossible to guarantee that the software is error free. However, Quelo will provide error corrections in a timely manner, given that sufficient information is available to identify the problem.

Quelo cannot be responsible for misuse of the PRODUCT or for the quality of software developed using the PRODUCT. The user is expected to be familiar with the 68000/68010 instruction set, basic assembly language concepts, relocation and linking.

Registration and Support:

The software distribution media come with a registration card. Please fill out the card and return it promptly to Quelo. Support and update notices will only be provided to registered users.

Each package is assigned a registration number which appears on the card and the distribution media. This number and the purchaser's name will be required for support. Names, addresses and phone numbers for support are provided on a sheet of paper separate from this document.

SECTION 2
Package Contents

Programs:

DATE - Set date (for CP/M only).

M68K - The Macro Pre-Processor.
A68K - The 68000/68010 Assembler.

QLIB - The Object Librarian.
QLINK - The Linker/Locator.

QSYM - The Symbol Report Generator.
- Module Summary.
- Symbol Table Listing.
- Cross Reference.
- Memory Map.

IMAGE - The HEX to Memory Image Converter.
SPLIT - The HEX File Even/Odd Splitter.

Documentation:

68000 Assembler Package Overview Manual (this document).
68000 Assembler and Macro Pre-Processor Manual.
Linker and Object Librarian Manual.
Symbol Report Generator Manual.

System Dependent Supplement.

READ.ME file on distribution disk.

Example Files:

This example demonstrates the software configuration tracking facilities of the package. See the Linker and Object Librarian Manual, Section 1, under Running the Linker.

LT30.A68	Main program.
LT31.A68	Configuration list.
LT32.A68	Output utilities.
LT33.LNK	The link specification.
LT34.LNK	Link specification for CP/M-68K.

This next example demonstrates the object librarian. See the Linker and Object Librarian Manual, Section 6, under Object Librarian Demonstration.

QT33.LIB	The library specification.

This example may be used to exercise the macro pre-processor. See the 68000 Assembler and Macro Pre-processor Manual, Section 7.

MACRO.M68	Macros and macro calls.

Changes Since Previous Release 4.2:

- The object librarian has been added to the package.
- The new binary relocatable object format has been implemented. Complex expressions may be passed on to the linker for evaluation. The object format is generalized to be processor independent. Quelo is using it for other processors as well as the 68000. Each object file begins with an ASCII header which can be viewed via the users terminal. Header information includes the information supplied with the assembler IDNT directive.
- Some program names and file extension names have been changed. This serves two purposes, to avoid confusion between the old and new package releases and to emphasize the processor independence of some of the programs. Only the assembler and macro pre-processor are now specific to the 68000.

Program Name:

	Old	New	Description
		A68K.COM	68000/68010 assembler
		DATE.COM	set date
		IMAGE.COM	HEX to binary
		M68K.COM	macro pre-processor
New		QLIB.COM	object librarian
Change	L68K.COM	QLINK.COM	linker
Change	S68K.COM	QSYM.COM	symbol report generator
		SPLIT.COM	split HEX file

File Extension:

	Old	New	Description
		.A68	assembler source file
		.ERR	macro processing error file
		.EVN	even HEX file from SPLIT
		.HEX	linker output file
		.IMG	binary file from IMAGE
New		.LIB	librarian specification file
Change	.L68	.LNK	linker specification file
		.LST	linker and librarian list file
Change	.R68	.LTX	binary object file
		.M68	macro pre-processor source file
		.ODD	odd HEX file from SPLIT
		.PRN	assembler list file
		.RPT	symbol report file
Change	.S68	.SYM	symbol report generator source file

- For CP/M, date access operations are now directed to the currently logged in drive, rather than drive A: as before.

- The linker now accesses library files.

LINK libfile	load entire library
LINK libfile()	search library for needed modules
LINK libfile(mod7,mod4)	load specific modules from library
LINK modfile	load module
LINK modfile()	load module only if needed

- The OPT directive option PCF has been added to force PC relative addressing on forward references. This option is useful for software development for systems that require PC relative addressing.
- "A" or "a" entered at the console will abort the program and close files (formerly control C).
- The memory limitations of SPLIT and IMAGE have been eliminated.
- The assembler supports local symbols defined as an offset from an external symbol. The offset is limited to the range -32768..32767.

```
        XREF    xsym
xoff    EQU     xsym+4
```

- The special lines used for communication of information from M68K to A68K now begin with a tilde, rather than an asterisk. This eliminates the confusion sometimes encountered with user comments.

SECTION 3
Program Invocation Command Lines

Lower Case Considerations:

Unless otherwise instructed, the various programs in the assembler package will convert lower case characters in user defined symbols to upper case characters. If it is desired that lower case be distinct from upper case, the lower case command line option, -L, should be invoked each time the following programs are used: M68K, A68K, QLIB and QLINK. Failure to do this consistently will probably result in undefined symbol error messages.

Even when the lower case option is selected, lower case will still be treated the same as upper case for assembler instruction mnemonics, assembler and linker directives and librarian commands. The option only affects user defined symbols, including module names specified with the assembler and linker IDNT directives and referenced in the linker LINK directive and in the librarian ADD, COPY and DELETE commands.

Command Line Contents:

Program command lines consist of three basic kinds of information. First is the name of the program to execute. This is used only by the operating system. The other two kinds of information may appear in any order.

One kind is the input/output specification. There must be exactly one of these for most programs. None or more than one will result in an error message and program termination.

The other kind selects program options. A leading hyphen (dash or minus sign) distinguishes an option from an input/output specification. There may be any number of options in the command line, including none. With few exceptions, most program options can be run together following a single hyphen.

Command line items are delimited by spaces. Therefore, option selections and input/output specifications may not have embedded spaces.

For example: A68K -L -S LT30,LPT: = LT30
"A68K" is the name of the 68000 assembler program for the operating system to load and execute.
"-L" and "-S" are program option selections. "-SL" would have the same effect.
"LT30,LPT: = LT30" is the input/output specification to be described below.

Quelo Input/Output Specification (Three Formats):

Input/output specifications may appear in one of three basic forms. In all forms at least one input must be specified. More than one input usually means the concatenation of files or devices to make up a single input. The exception would be in programs which require more than one distinct input. Note that programs which recognize something in a file such as an END directive will terminate processing even though all entries in an input list have not been processed.

<input 1>,<input 2>,<input 3>...

The "short" form shown above contains no output specification. All required output file names are derived from the first entry in a list of one or more input file or device names.

=<input 1>,<input 2>,<input 3>...

The above "intermediate" form also contains no output specification, but implies that only the first (primary) output is to be derived from the first input entry. The remaining outputs will be directed to the "NUL:" device (bitbucket). The leading equal sign distinguishes the "intermediate" form from the "short" form.

<output 1>,<output 2>... =<input 1>,<input 2>...

The above "full" form includes an explicit list of output files or devices. Those outputs omitted from the list will be directed to the "NUL:" device. When an output file name is not specified, but the drive and/or file type is, the name will be taken from the first entry in the input list.

For both inputs and outputs, the file type will be set to a default if not explicitly specified. The default depends on the particular program being used. For outputs, the drive will be taken from the first entry in the input list if not explicitly specified.

The following formats are suitable for input specification:

d:filename.typ	
d:filename	
filename.typ	
filename	file name alone
	null - ignored
CON:	the console
TTY:	the console

The following formats are suitable for output specification:

d:filename.typ	
d:filename	
filename.typ	
d:.typ	
d:	drive alone
filename	file name alone
.typ	file type alone
	null- treated as NUL:
CON:	the console
TTY:	the console
LST:	the printer
LPT:	the printer
NUL:	the bit bucket

The drive letter may range from "A" to "P" or from "a" to "p". An "@" may also be used to specify the currently logged-in drive.

Program Input/Output Summary:

<output 1>,<output 2>,<output 3> = <input 1>,<input 2>...

	M68K	A68K	QLIB	QLINK	QSYM
output 1 (primary)	assembler source .A68	relocatable object module .LTX	listing .LST	HEX load module .HEX	various symbol reports .RPT
output 2	errors and symbol table .ERR	listing .PRN	symbols and cross reference data .SYM	listing .LST	
output 3		symbols and cross reference data .SYM		symbols and cross reference data .SYM	
input(s)	source with macros .M68	assembler source .A68	library specification .LIB	link specification .LNK	symbols and cross reference data .SYM
"include" input	source with macros .M68	assembler source .A68	library specification .LIB	link specification .LNK	
object input/ output			object module or library .LTX	object module or library .LTX	

Note that "include" inputs are specified with the INCLUDE directives recognized by M68K, A68K, QLIB and QLINK. "object" inputs are specified with the LINK directive recognized by QLINK. "object" inputs and outputs are specified with the CREATE, UPDATE, EXTRACT, ADD and COPY commands recognized by QLIB.

Listing and Report Page Formatting:

The LLEN and PLEN directives may be used in A68K and QLINK to establish listing page format. Page format information is passed from A68K, QLIB and QLINK to QSYM in the .SYM symbol table and cross reference data file. This information may be overridden by using the "-F" option when running QSYM to produce the various symbol reports. Page title information is also passed to QSYM in a similar manner and may be overridden by using the "-H" option.

Page breaks are normally accomplished via the ASCII formfeed character. However, indicating a negative bottom margin in the PLEN directive or the QSYM "-F" option will cause blank lines to be issued for page breaks instead of a formfeed.

Line length (LLEN) is the number of characters allowed on a listing line. Page length (PLEN) is the total number of lines on a page, from perforation to perforation. Top

margin (tm) is the number of lines to space down before printing begins. Bottom margin (bm) is the number of blank lines desired at the bottom of the page. The number of printed lines (including page heading information) will be (PLEN - tm -bm).

Listing and report headings include page numbers and a date-time stamp if the information is available. See the documentation specific to the implementation of A68K for your operating system for date and time availability.

Program Option Summary:

		M68K	A68K	QLIB	QLINK	QSYM
A		global arguments				
B			same effect as -SX	same effect as -SX	same effect as -SX	combined symbol table and cross reference reports
C		define character				compressed map
E			local symbols to .LTX and .SYM		local symbols to .SYM	discard local symbols
F(line length),(page length),(top margin),(bottom margin)						page format
HS HM HI					S-records Mostek HEX Intel HEX	
H(page heading title info)						page title
I				module info to .SYM	module info to .SYM	module summary
L		retain lower case	retain lower case	retain lower case	retain lower case	
M						memory map
S		symbols to .ERR	symbols to .SYM	symbols to .SYM	symbols to .SYM	symbol table
T		truncate symbols	truncate symbols	truncate symbols	truncate symbols	
V		formfeed to .ERR	formfeed to .PRN	formfeed to .LST	formfeed to .LST	formfeed to .RPT
X			cross reference to .SYM	cross reference to .SYM	cross reference to .SYM	cross reference
Z		define symbol table size	define symbol table size	define symbol table size	define symbol table size	define symbol table size
1		debug	debug	debug	debug	debug

Note: The debug option shows the symbol table space to be allocated in hexadecimal at the console. Also, for A68K and QLINK the listing is produced for all passes.

The formfeed option places a formfeed at the start of listing or report files.

M68K Global Parameter Substitution:

The "-A" option to M68K allows the passing of arguments from the command line into the text being processed. When appearing outside of a macro, "\0" through "\9" reference these command line arguments.

For example:

Command line:	-Azero,one,,,four
M68K input line: (not in macro)	* \1 *** \0 *\2*\4*
M68K output line:	* one *** zero **four*

Note that spaces may NOT be included in the arguments and the limit is 10 arguments. This feature can be very handy for controlling conditional assembly without having to edit the source file.

Symbol Table Size Control ("-Z" option):

The various programs will take as much memory as is available (but not more than 32766 bytes) for symbol table space or buffer space in the case of the SPLIT and IMAGE programs.

Some difficulty has been encountered in determining how much space is really available when running on the PC-Jr. For that reason the "-Z" command line option has been implemented. This option allows the user to specify symbol table space directly.

If you get an "ALLOC" error message, run the program with the "-1" option to discover how much space the program is trying to take. The value displayed is the number of bytes given in hexadecimal. Run the program again with the "-Z" option to specify a smaller amount of space. It will take some experimentation to find the maximum value that will work for a given program.

-Z ⟨constant⟩ where ⟨constant⟩ may have several forms:

Leading $ indicates HEX
trailing K indicates multiply by 1024

The following examples all represent the same value:

-Z16384	deciamal
-Z$4000	hex
-Z16K	16 * 1024
-Z$10K	16 * 1024

SECTION 4
Listing Date-Time Stamping

Introduction:

When available, date and time information is included in listing page headings. The source of this information will vary from one system to another. Some operating systems provide time-of-day information and others do not. For those that do not, a program is provided to create and maintain a special date file.

CP/M:

CP/M releases prior to 3.0 do not have a date-time facility. For those versions, a "date" file is stored on the currently logged in drive under user 0. If the CP/M release is prior to 2.0, the user number does not apply.

Each program which uses the date file selects the currently logged in drive and user 0 prior to searching for the file. Each program may be patched to change the date drive and user number. Contact Quelo for patch information.

The date file contains no information, as the file name is the date. The name format is "mm/dd/yy.DAT". The maintenance program is called "DATE". If the date file already exists, DATE will display the date and issue a verify prompt. A "N" or "n" response will obtain the prompt for entering a new date. When only the day is to be changed, that is all that has to be entered.

If the date file does not exist, DATE immediately issues the prompt for entering the date. The program terminates with a "Y" or "y" response to the verify prompt. Invalid responses result in a repeated prompt.

MP/M:

MP/M has a built-in date-time facility, so the "date" file does not apply.

MS-DOS, PC-DOS:

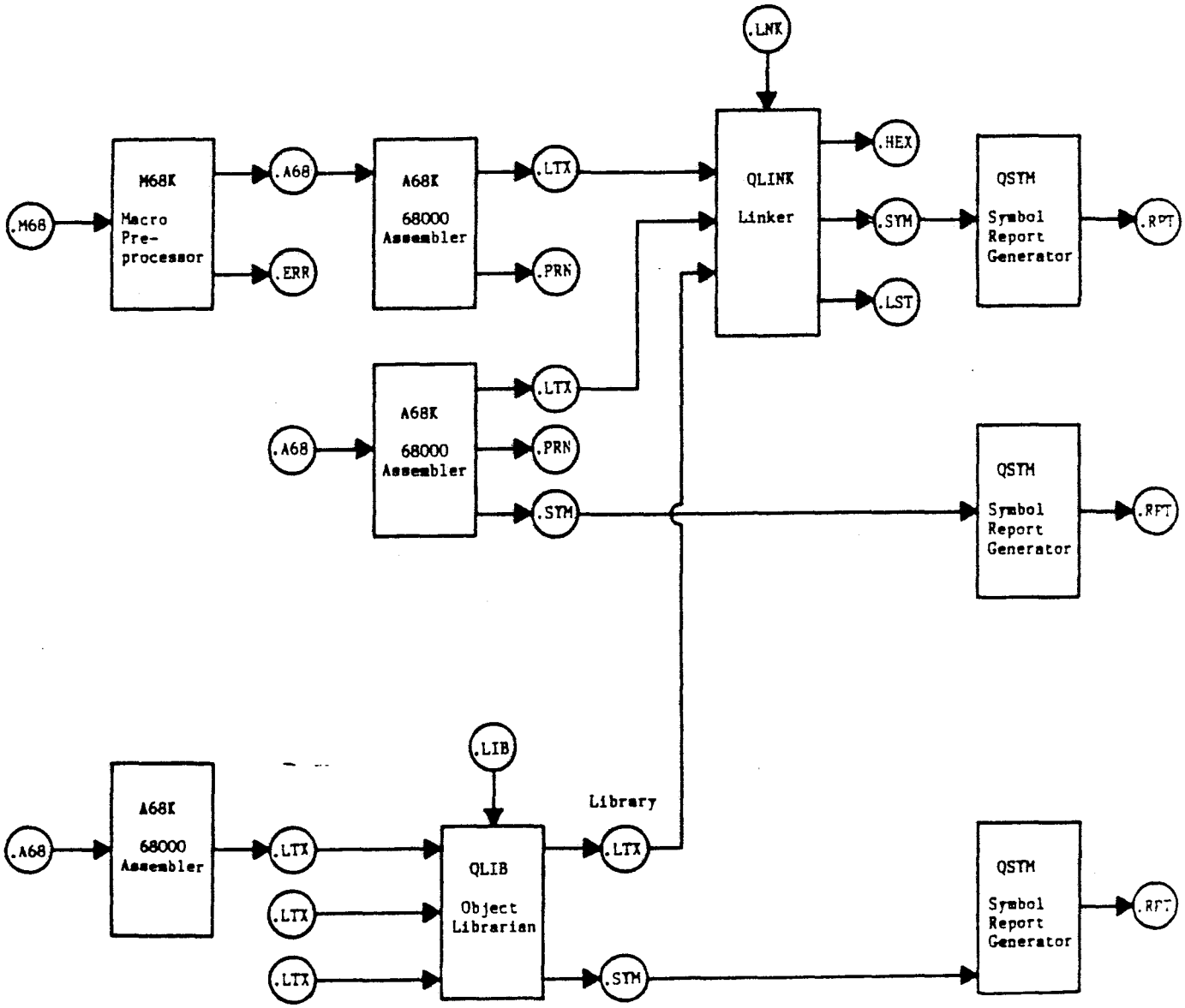
MS-DOS has a built-in date-time facility, so the "date" file does not apply.

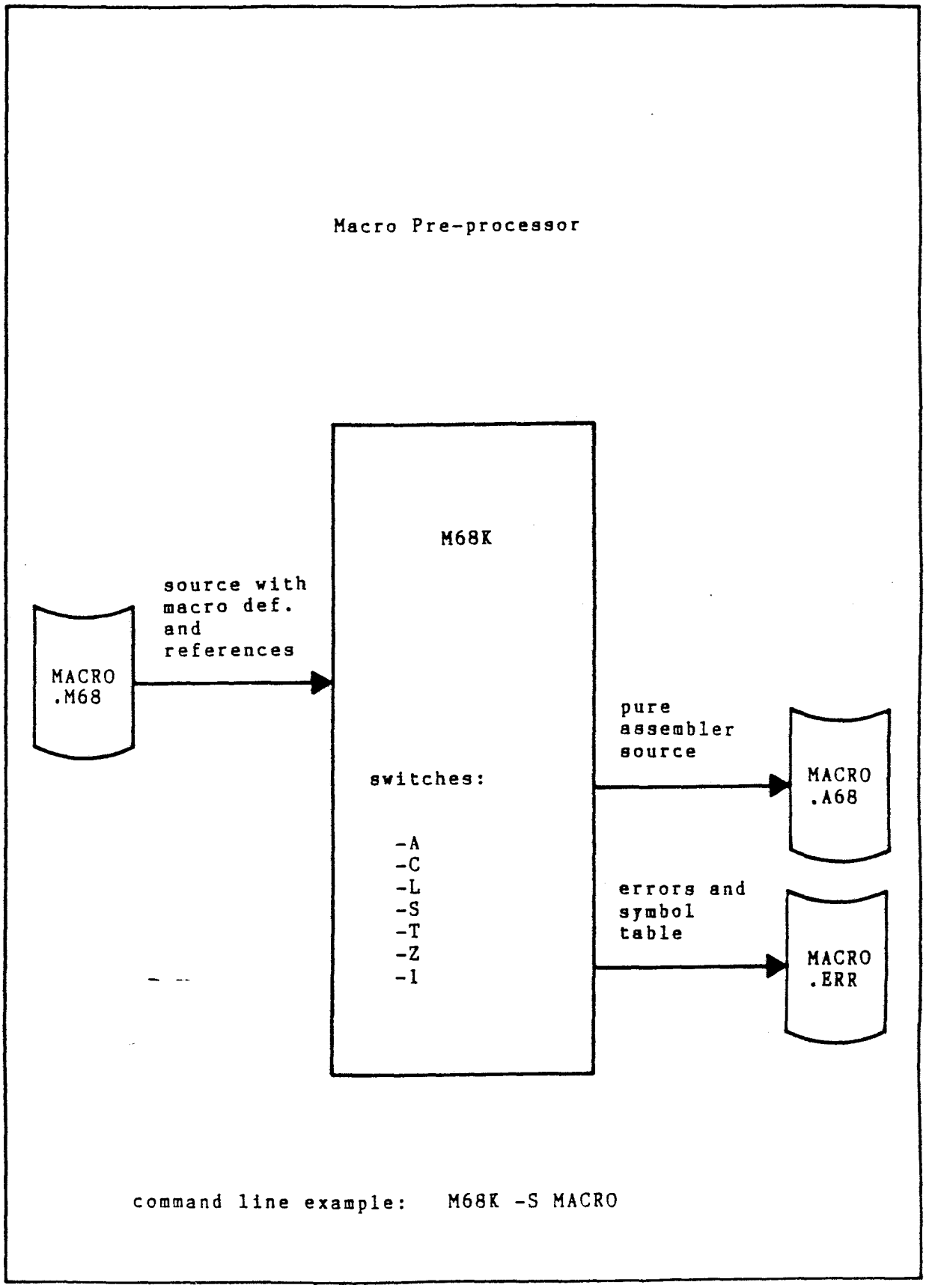
Portable Assembler Package:

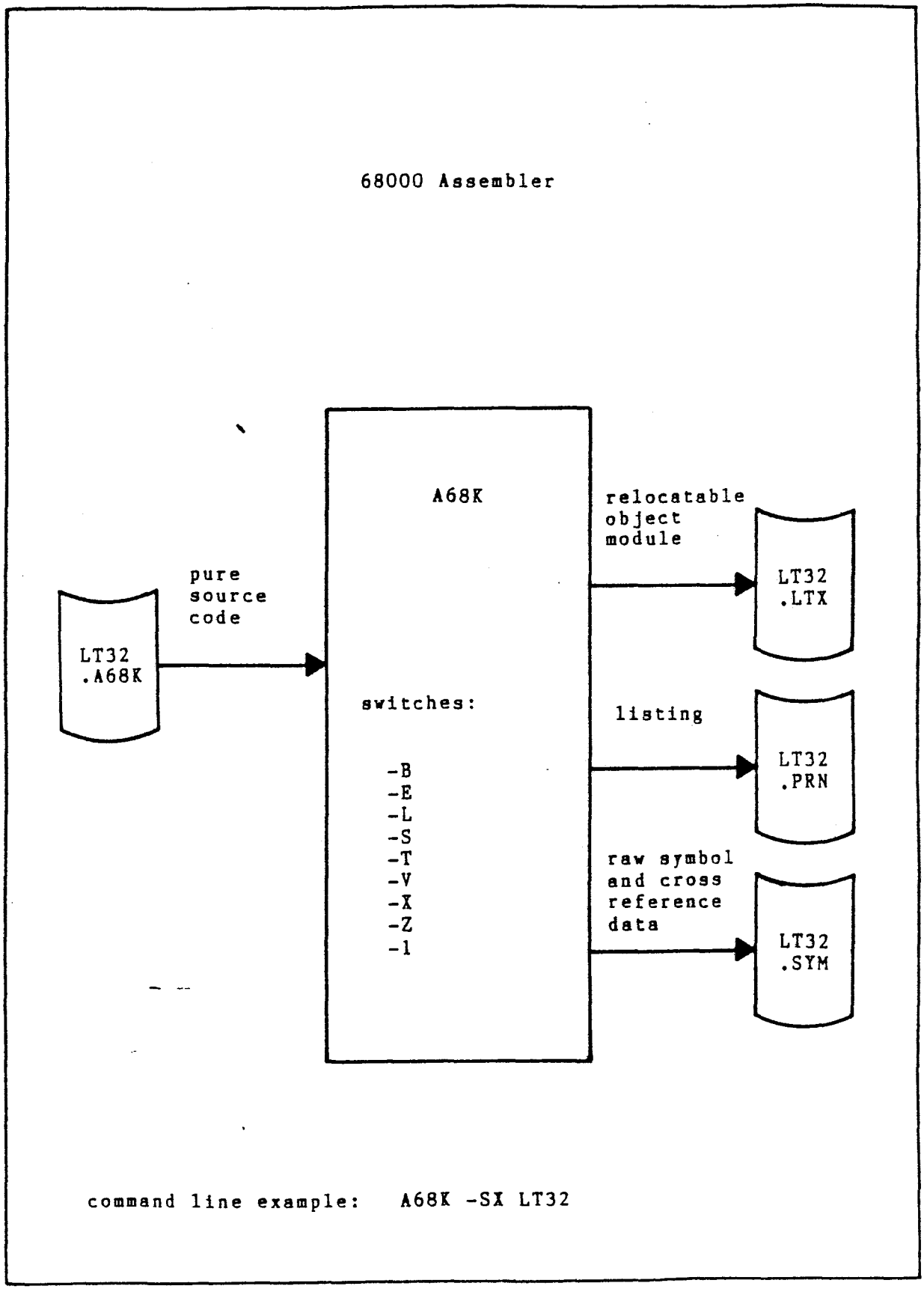
Date-time access is up to the implementor.

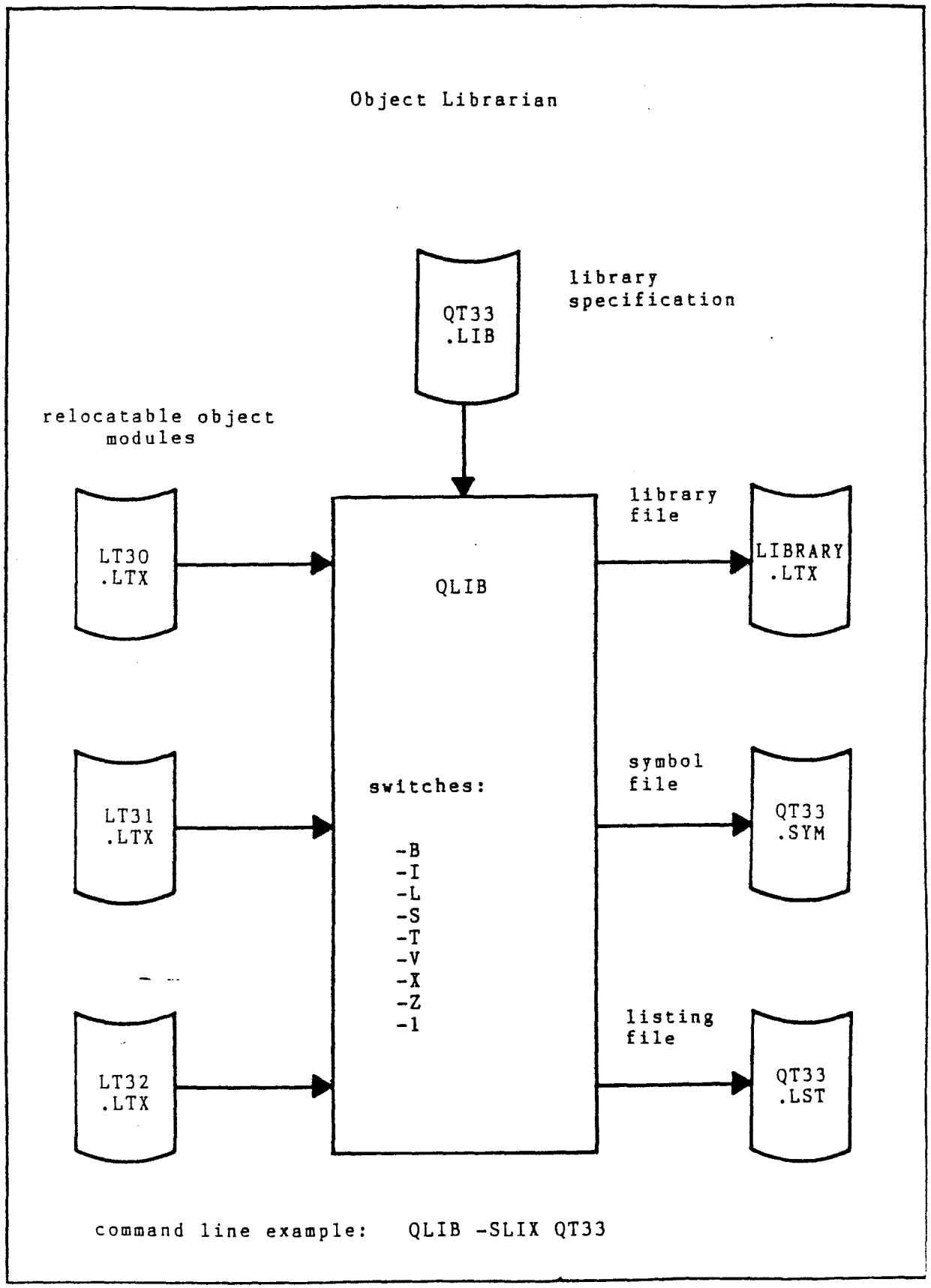
SECTION 5
Flow Diagrams

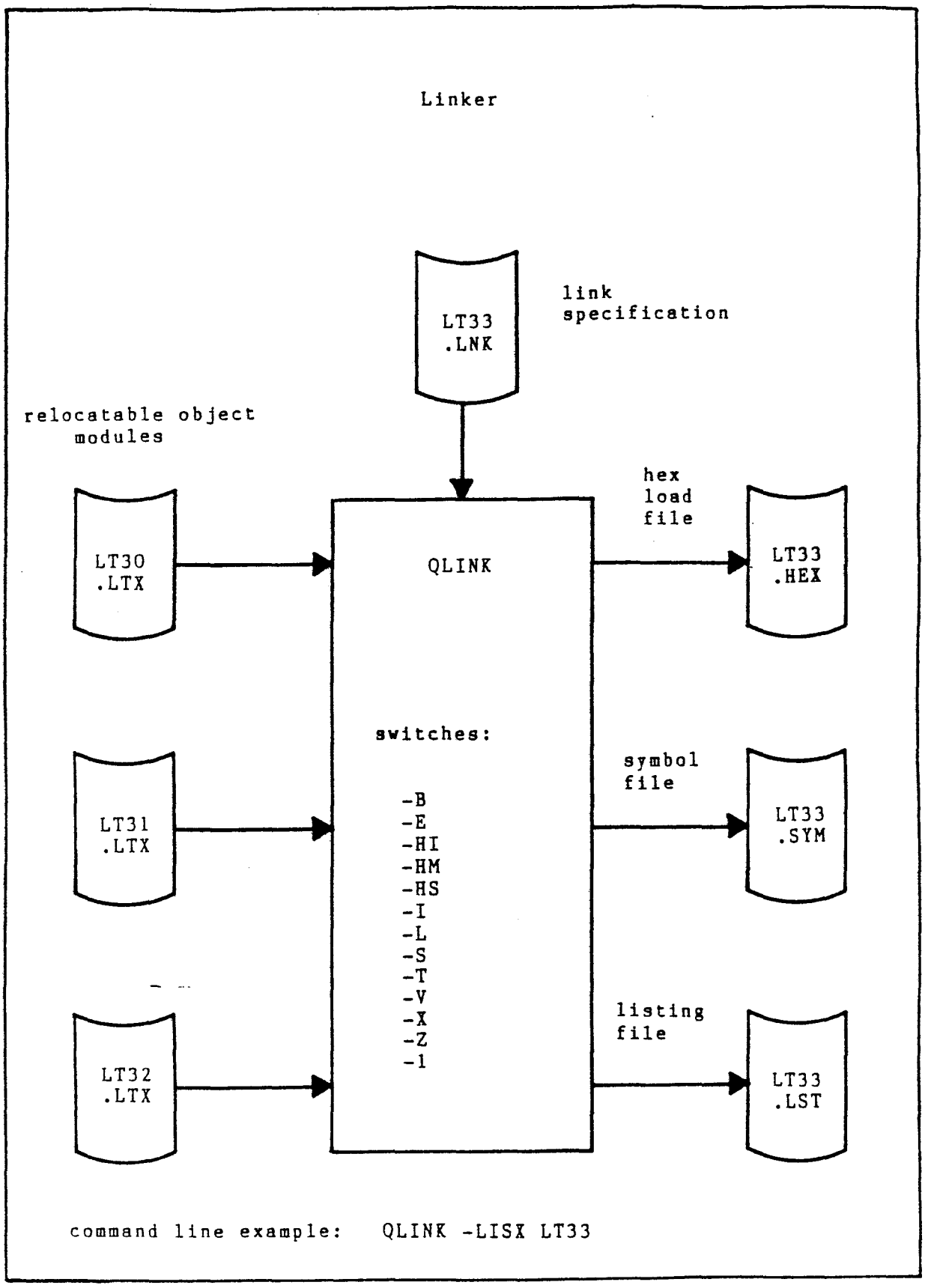
Inter-Program Information Flow

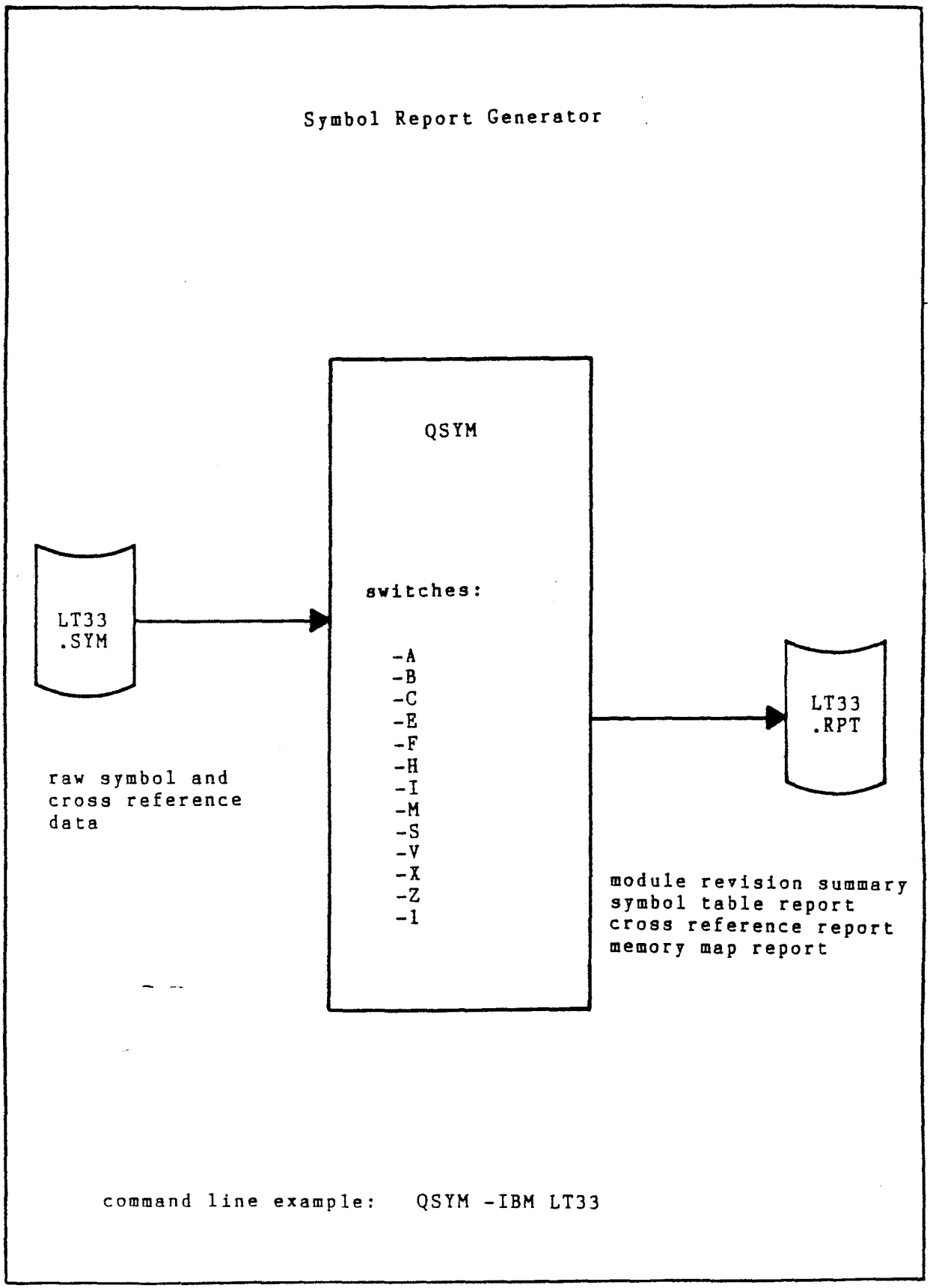



Macro Pre-processor
M68K
source with
macro def.
and
references
MACRO
.M68
switches:
-A
-C
-L
-S
-T
-Z
-1
pure
assembler
source
MACRO
.A68
errors and
symbol
table
MACRO
.ERR
command line example: M68K -S MACRO

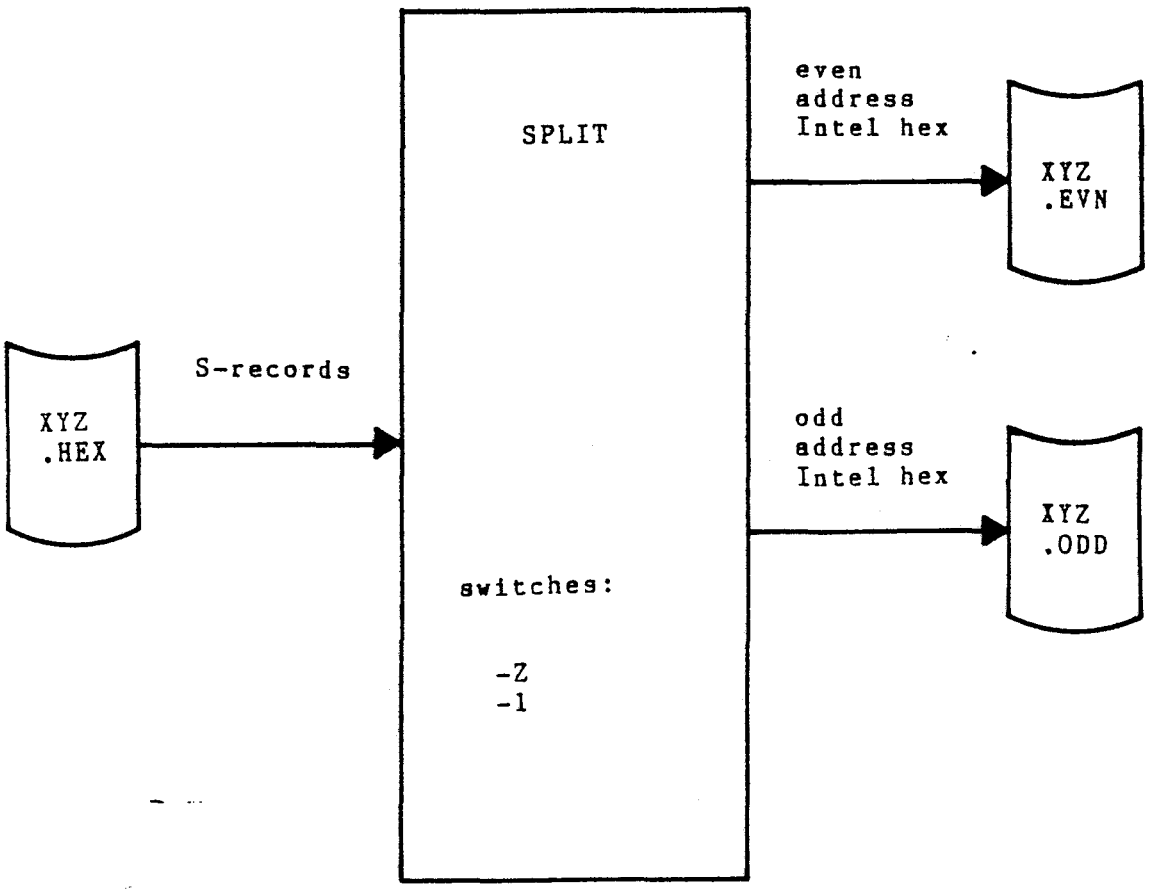



Object Librarian
QT33
.LIB
library
specification
relocatable object
modules
LT30
.LTX
LT31
.LTX
LT32
.LTX
QLIB
switches:
-B
-I
-L
-S
-T
-V
-X
-Z
-1
library
file
LIBRARY
.LTX
symbol
file
QT33
.SYM
listing
file
QT33
.LST
command line example: QLIB -SLIX QT33

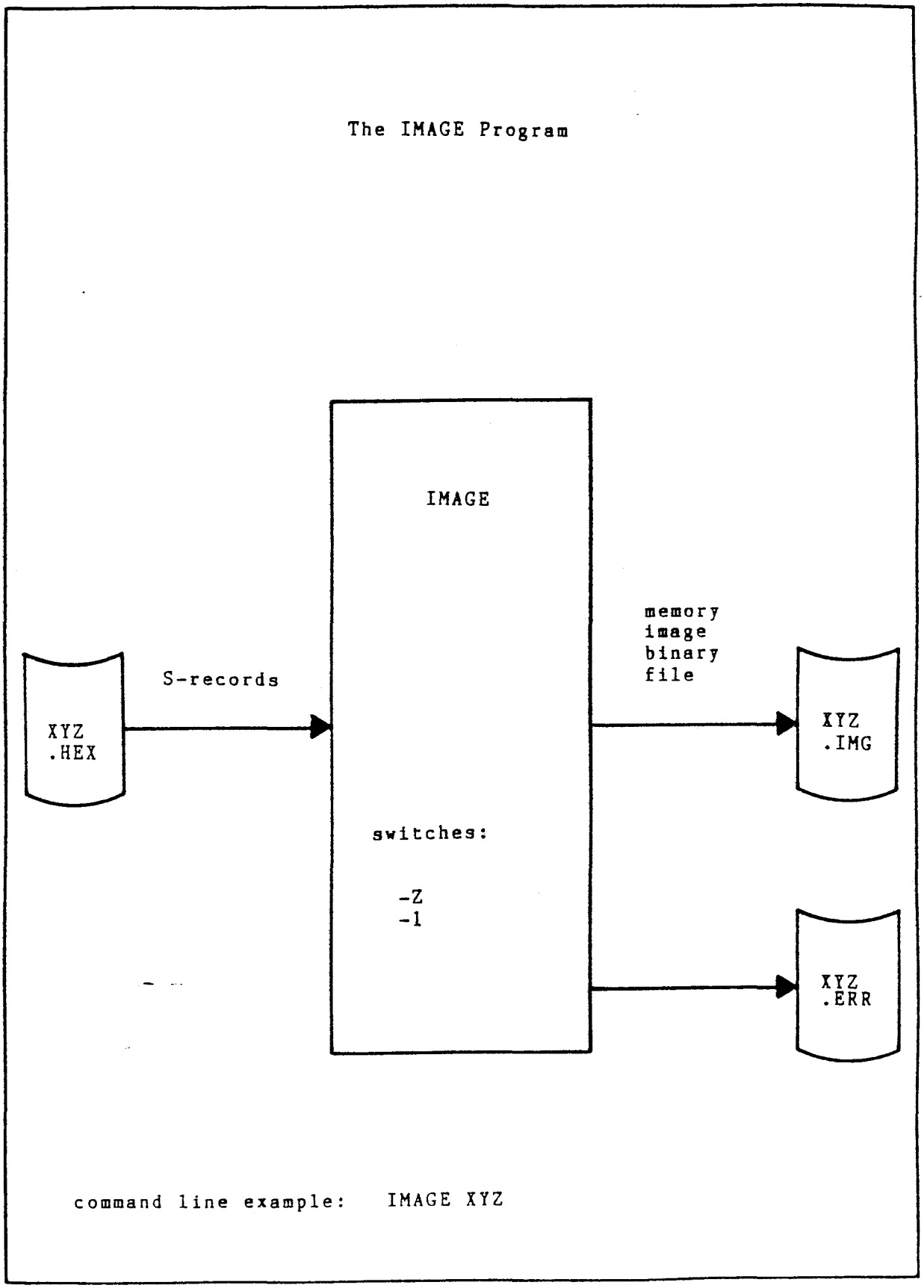

Linker
LT33
.LNK
link
specification
relocatable object
modules
LT30
.LTX
QLINK
hex
load
file
LT33
.HEX
switches:
-B
-E
-HI
-HM
-HS
-I
-L
-S
-T
-V
-X
-Z
-1
LT31
.LTX
symbol
file
LT33
.SYM
LT32
.LTX
listing
file
LT33
.LST
command line example: QLINK -LISX LT33

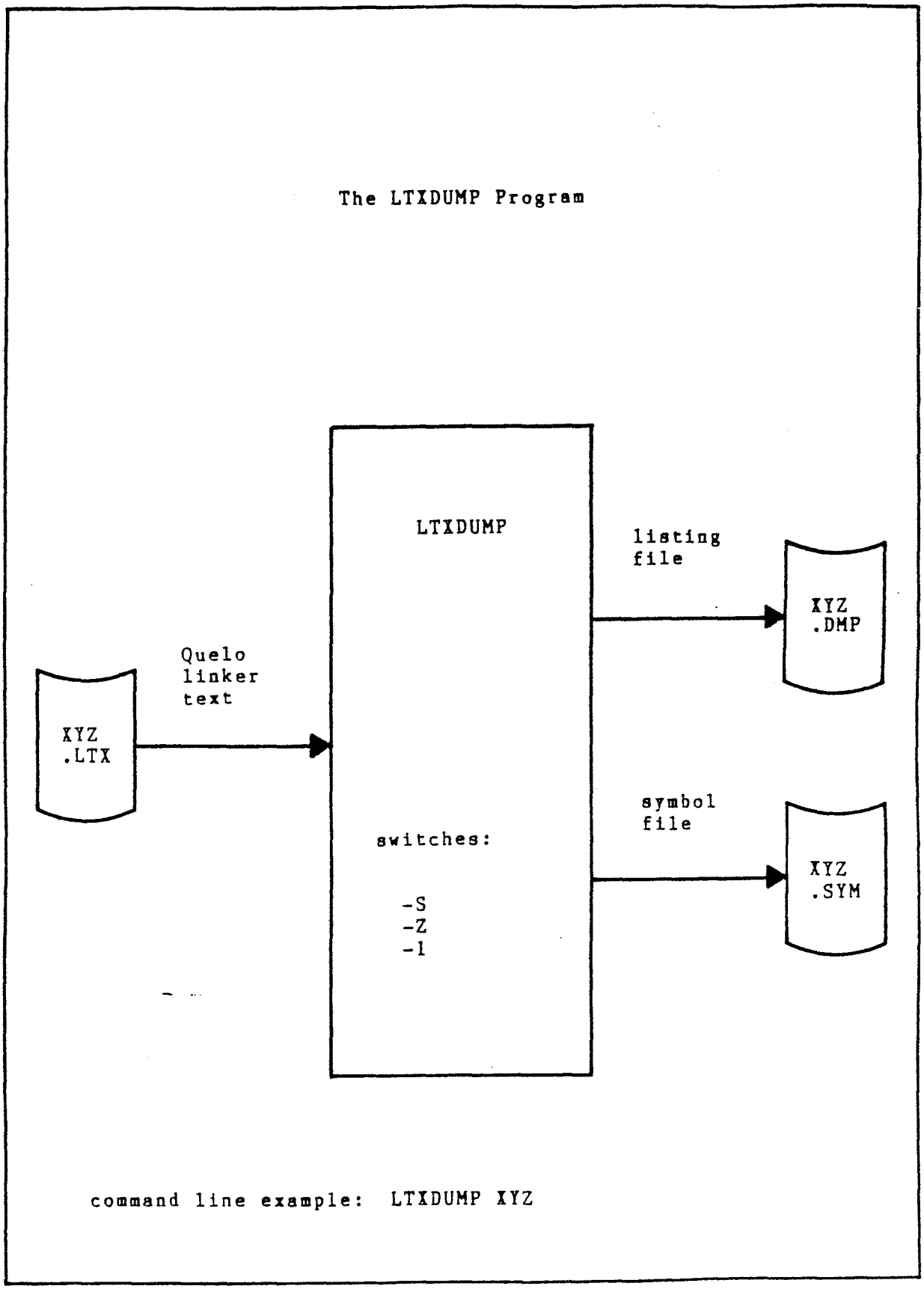

Symbol Report Generator
QSYM
switches:
-A
-B
-C
-E
-F
-H
-I
-M
-S
-V
-X
-Z
-1
LT33
.SYM
raw symbol and
cross reference
data
LT33
.RPT
module revision summary
symbol table report
cross reference report
memory map report
command line example: QSYM -IBM LT33

The SPLIT Program



command line example: SPLIT XYZ


The IMAGE Program
IMAGE
memory
image
binary
file
S-records
XYZ
.HEX
XYZ
.IMG
switches:
-Z
-1
XYZ
.ERR
command line example: IMAGE XYZ



A P P E N D I X A
CP/M-68K Executable File Generation

A link specification can be set up to generate header information for CP/M-68K executable files. As the linker only produces HEX output files, it will still be necessary to convert the HEX file into a binary image file by means of the IMAGE program.

The following link specification assumes that program segments are assembled under section 0, initialized data is assembled under section 1 and uninitialized data is placed in section 2. Note that neither the assembler nor the linker will complain if there is code generated for section 2. It is up to the user to make sure that the bss area really has no initialized data.

```
cpm. base   equ $400
cpm. tpa    equ cpm. base + $100

tseq    equ     0       text section number
dseq    equ     1       data section number
bss     equ     2       uninitialized data

        org     cpm. tpa-$1c
header:
        dc.w    $601a
        dc.l    data.seg-text.seg
        dc.l    bss.seg-data.seg
        dc.l    prog.end-bss.seg
        dc.l    0               no symbols
        dc.l    0               always zero
        dc.l    text.seg        starting point
        dc.w    $ffff           no relocation
text.seg:
        section tseg
data.seg:
        section tseg
bss.seg:
        section bss
prog.end:
        end     text.seg
```

APPENDIX E
Error Messages Common To All Programs

Error Message Explanations:

Bad channel. (ABORT)

Please report this to Quelo. (all programs)

Disk write. (ABORT)

Probably means file or directory space is full. (all programs)

Invalid label.

Label contains invalid characters. (assembler and linker)

Label missing.

A label is required with the directive. (assembler and linker)

Label not permitted.

A label is not permitted with the directive. (assembler and linker)

Object header error. (ABORT)

A file being read as Quelo linker text does not begin with a valid header. Make sure the file specified by the linker LINK directive or the librarian ADD directive is really a linker text file. (linker and librarian)

Parser fault. (ABORT)

Please report this to Quelo. (all programs)

Parser stack not empty.

Might be caused by syntax errors, especially when expressions are being processed. (macro pre-processor, assembler, linker)

Parser stack overflow.

Might be caused by attempting to assemble a very long string. (all programs)

QPL mismatch. (ABORT)

Please report this to Quelo. (all programs)

QPL re-entry. (ABORT)

Please report this to Quelo. (all programs)

QPL ufl. (ABORT)

Please report this to Quelo. (all programs)

QPL unknown function. (ABORT)

Please report this to Quelo. (all programs)

Read from closed channel. (ABORT)

Please report this to Quelo. (all programs)

Unexpected object EOF. (ABORT)

An unexpected end-of-file condition was detected during the reading of a Quelo linker text file. (linker and librarian)

Write to closed channel. (ABORT)

Please report this to Quelo. (all programs)