

**REAL-TIME EXECUTIVE
SYSTEM CROSS LOADER**

**PROGRAMMING AND
OPERATING MANUAL**



PROGRAMMING AND OPERATING MANUAL

REAL-TIME EXECUTIVE SYSTEM CROSS LOADER

(RTE SXL)

15 AUGUST 1973

— IMPORTANT NOTICE —

This manual contains information on Hewlett-Packard Real-Time Executive software. The reader is assumed to be a programmer familiar with one of the Hewlett-Packard FORTRAN or Assembler programming languages.

Part No. 29103-93001

Microfiche No. 29103-93002

Copyright © 1973

HEWLETT-PACKARD COMPANY

Sunnyvale, California

First Edition

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system (e.g., in memory, disc or core) or be transmitted by any means, electronic, mechanical, photocopy, recording or otherwise, without prior written permission from the publisher.

Printed in U.S.A.

CONTENTS

	Page
GLOSSARY	vii
I GENERAL DESCRIPTION	1-1
Introduction	1-1
Software Requirements	1-1
Hardware Requirements	1-1
Installation	1-1
Description	1-1
Duplicate Symbol Definition	1-1
Use of DEBUG	1-1
II COMMANDS	2-1
Input Formats	2-2
Syntax Punctuation Symbols	2-2
Keywords	2-2
Command Language	2-3
ABORT	2-3
BOUNDS	2-3
DISPLAY	2-4
ECHO	2-5
END	2-5
LINKS IN	2-5
LINKS START AT	2-6
MAP	2-8
Information Contained in the Core Map	2-8
Module Core Bounds Information	2-8
Global Variables Information	2-9
LIBRY	2-9
Link Table	2-9
OUTPUT	2-9
RELOCATE	2-12
SEARCH	2-13
SEGMENT	2-14
SET	2-15
SNAP	2-17
TRANSFER	2-18
III OPERATING PROCEDURE	3-1
ON, SXL	3-1
RTE SXL SCHEDULE	3-1
Phase I. Target Machine Configuration	3-1
Phase II. Relocation Commands	3-2
Order of Command Input	3-4
Distributed System Use	3-4
IV ERROR MESSAGES	4-1
BP MEM OVF	4-1
CHSM ERR.	4-1

RTE SXL

Section	Page
COMMON LNTH ERR	4-1
CREAT ERR	4-2
DUPL EP:	4-2
EVAL. ERR	4-2
FILE READ OR WRITE ERR	4-2
ILL. BP LNTH	4-2
ILL CHAR	4-2
ILL. EXT	4-2
ILL. LINKS STRT ADDR.	4-3
ILL. RCRD TYPE	4-3
MEM OFV	4-3
NAM OUT OF SEQ	4-3
NO CMND ID	4-3
NO LINK ROOM	4-3
NO MAIN PRGM	4-3
NO NAM RCRD	4-4
OPEN ERR	4-4
SXL WAITING FOR DISC TRACKS	4-4
UNRECOGNIZED STMT	4-4
WHAT NAME TO USE FOR ABS. FILE?	4-4
 V NON-RTE TARGET SYSTEMS	5-1
Theory of Operation	5-1
Software Requirements	5-1
Hardware Requirements	5-1
Operating Instructions	5-3
Procedure for Turning on BCSGN	5-3
ON, BCSGN	5-3
Procedure for Turning on TC3GN	5-3
ON, TC3GN	5-3
Procedure to Abort BCSGN or TC3GN	5-3
Operator Dialogue	5-3
Errors and Diagnostics	5-8
Interface Details	5-8
Input Data Format	5-8
Output Data Format	5-9
Prompt Messages	5-9
Examples	5-9
 VI RELOCATING PROGRAMS FOR RTE-C (HP 9600C)	6-1
Snapshot Phase	6-1
RTE-C Snapshot Format	6-1
Core Bounds	6-1
Resident Entry Points	6-1
 VII PRECAUTIONS	7-1
INDEX	Index-1

ILLUSTRATIONS

Figure	Page
1-1. RTE System Cross Loader Operation	1-2
2-1. Current Page Mode Link Areas	2-7
2-2. Typical Core Map	2-10
2-3. Core Map Without Current Page Linking	2-11
3-1. Module 1 Declares Common	3-3
3-2. Module 2 Declares Common	3-3
5-1. BCS or TCE/3 Generation	5-2
5-2. Examples of Setting Constants in the Interrupt Location	5-9
5-3. Batch BCS Generation	5-10
5-4. Interactive BCS Generation	5-11

TABLES

Table	Page
2-1. RTE SXL Commands	2-1
4-1. Summary of RTE SXL Error Messages	4-1

GLOSSARY

COREIMAGE —	The special way of storing programs for high-speed access used in RTE.	Module (Cont)	Assembler or the source, of a single subprogram (which may be the main subprogram). The relocatable binary code consists of an indefinite number of records, beginning with a NAM record and ending with an END record; there may be any number of ENT, EXT, and DBL records in between.
FIXUP —	The information saved by SXL when a forward reference is encountered, such that when the forward-referenced (FR) symbol becomes defined, the FR instruction can be relocated in the proper linking mode.	OVERLAY —	A portion of a program which is brought into core as necessary to perform some process, under control of some "main program" which is core-resident at least during the time the overlay is in core.
FORWARD REFERENCE —	An instruction in relocatable code which references an external symbol for which SXL has not yet obtained a definition, either through relocating an ENT record in an earlier module or a SET statement entered in a user command.	PROGRAM UNIT —	Refers to the relocated code output of RTE SXL. A "program unit" normally contains all submodules referenced by the individual modules which it contains. For example, a module which performs I/O using the formatter is not complete until it has been linked with the FRMTR library routine and relocated into a "program unit." Program units are always executable when moved into core by a primitive core "loader" such as the Basic Binary Loader.
HOST MACHINE or HOST SYSTEM —	The computer and operating system environment in which the various program preparation programs (editor, compiler, loader, etc.) execute.	SEGMENTED MAIN PROGRAM —	A program written in such a way that it does not need all of itself to be in a core at all times; when a portion of code needs to be executed (in order to perform some specific job) it is made core-resident via a "segment load EXEC call" to the monitor
LINK —	A 16-bit address word used for off-page references. This address is stored in the base page (or with the user code, current page linking mode only).		
MAIN —	That portion of a user's program which is core resident during program execution. Control passes from the operating system to the main program to begin execution.		
MODULE —	The resultant, relocatable binary code of a compiler (e.g., FORTRAN or ALGOL) or the		

(Continued)

(Continued on next page)

Glossary (Continued)

Segmented
Main Program
(Cont)

system, and control passed to this portion. The blocks of code which can remain non-core resident are called "segments". It is always necessary to have a "main program area", which is always in core while the program is executing. It contains flags and pointers, receives control when the program is initially turned on, and serves as a main control. It may consist of more than one module, and it often contains modules which are shared by some of the segments. While in core, modules in the main program area and those in the segment area (there may be several of those, too) execute as a single program. That is, control may pass from anywhere in the segment area to anywhere in the main and back again via a JMP or JSB instruction (the operand must be declared external in the segment and an entry point in the main). Likewise, parameters in the main can be accessed from a segment. It is possible, of course, to access parameters in a segment from a main, but it is not generally a good idea, because whenever a segment is made core-resident, a fresh copy of it is brought in.

SEGMENT —

See definition for "Overlay".

SNAPSHOT —

A SNAPSHOT is an ASCII file in the form of command statements that saves the

(Continued)

Snapshot (Cont)

maximum core bounds, entry points, link table, and undefined externals in such a way that an overlay can be generated later. These "OVERLAYS" may be segments or they may be programs. Some operating systems (for example, BCS) can be built and configured entirely from the relocatable modules and operator keyboard commands, without requiring a system-configuration program. The SNAPSHOT is a mechanism by which RTE SXL connects system-dependent symbols (e.g., .IOC., EXEC, etc.) in the particular target (terminal) machine to programs prepared for that system using SXL. It also allows the user to relocate only the segment (in a segmented main program) which he wishes to change.

SXL —

The call name by which the Real-Time Executive System Cross Loader is known to other software or program segments. SXL also serves as a further shortened form of the abbreviated name RTE SXL. Therefore, within the text of this manual, SXL can be considered to be interchangeable with, and synonymous to, the terms RTE SXL and Real-Time Executive System Cross Loader.

TARGET
MACHINE or
TARGET
SYSTEM —

The computer and operating system for which the program is being prepared.

SECTION I

GENERAL DESCRIPTION

INTRODUCTION

The Real-Time Executive System Cross Loader (RTE SXL) provides a means for linking relocatable files produced by compilers or assemblers together with one or more library files and the system core-resident entry points for RTE-C, TCE/3, or BCS target 2100 Series operating system environments, and to format this output into an executable form. The resulting binary output need only be brought into core in order to be run.

RTE SXL also has the following features:

- Current Page linking mode (optional).
- Choice of core image code or absolute binary output for arbitrary target 2100 Series operating system configurations.
- Accepts both RTE and FMP files.
- Load-time driver configuration.
- Accepts multiple types of libraries for different target computer configurations.
- Permits table building and core patching.
- Supports off-line command-file preparation.

RTE SXL accepts relocatable object programs from either an input device or a file created by the HP Assembler or by the RTE ALGOL or FORTRAN compilers. The files may reside on load-and-go tracks or on any other relocatable file.

RTE SXL can be used to relocate programs that are not necessarily intended to be run on the host system. The user makes files known to the system through an RTE File Manager (FMP) Restore Program (RP, *name*) command if they are to be run on the host machine. Figure 1-1 shows the general sequence of steps in the Loader operation.

SOFTWARE REQUIREMENTS

The Real-Time Executive System Cross Loader (RTE SXL) is available in punched-tape relocatable binary format as HP Part Number 29103-60001. RTE SXL contains all of the interface software necessary to execute in a 24K Real-Time Executive (RTE) Operating System with the RTE File Manager.

HARDWARE REQUIREMENTS

RTE SXL operates in a minimum of 6K of core in the background disc-resident (BKDR) area in an RTE system running on either an HP 2100A or an HP 2116B (or later) Computer. A keyboard-input console (teletype (TTY) or HP 2600A Terminal (CRT)) and a photoreader are required. A magnetic tape unit and a tape punch may optionally be added.

INSTALLATION

RTE SXL must be configured into the RTE System during the program input phase of RTGEN. The RTE File Manager and 24K of core are required. No provisions exist for on-line loading. (Refer to "Program Input Phase" in Section VI, RTE System Installation, of the Real-Time Executive Software System Programming and Operating Manual, HP Part Number 02005-90001.)

DESCRIPTION

RTE SXL accepts both relocatable binary code and user commands. The user commands direct the way in which the relocatable binary code is converted to output code. Defaults are the same as existing loaders, except where specifically noted. The user who wishes to perform a simple relocation of a program (simplest case), link up to entry points in the host system, and include routines from the system relocatable utility library need only use three commands: RELOCATE, SEARCH, and END. (These commands are described in detail in Section II, "Commands".) As his requirements become more sophisticated, he may learn the other commands as necessary.

DUPLICATE SYMBOL DEFINITION

RTE SXL normally forbids redefinition of entry point names. However, the user can use the SET . . . UNDEFINED statement before redefining it. A bit is set in the symbol table indicating the symbol has been "undefined."

USE OF DEBUG

DEBUG can be forced to take control of a user program either by including it in a LOAD statement and by changing the transfer address pointer (?XFER) to be DEBUG's entry point, or by relocating the DEBUG module prior to the main.

RTE SXL

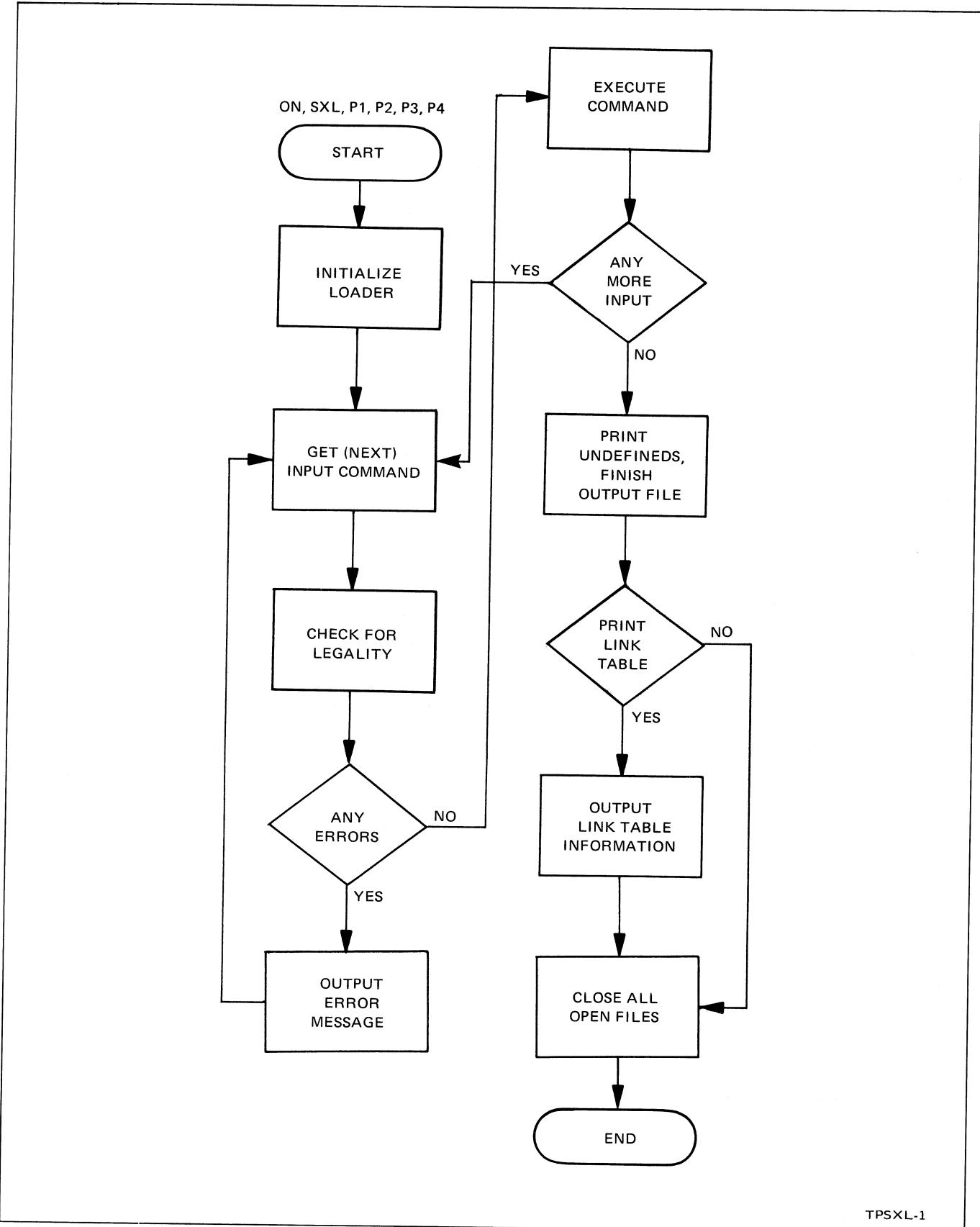


Figure 1-1. RTE System Cross Loader Operation

SECTION II

COMMANDS

Table 2-1 shows a summary of the operator commands which may be used to build an RTE System Cross Loader Command file.

Table 2-1. RTE SXL Commands

Command	Description	Page
ABORT	Terminate SXL execution on all errors	2-3
BOUNDS	Change core bounds for target machine	2-3
DISPLAY	Access contents of current symbol table	2-4
ECHO	Print command input as received	2-5
END	Terminate command input, and go to final processing	2-5
LINKS IN	Define SXL Link option	2-5
LINKS START AT	Set start of base-page Link area and initialize contents	2-6
MAP	Turn on listing flags for core map printout	2-8
OUTPUT	Define output format and file name for relocated code	2-9
RELOCATE	Open files and relocate modules within the files	2-12
SEARCH	Selective-load files or modules ("library load")	2-13
SEGMENT	Begin relocation of a segment	2-14
SET	Create, define entry point names, and initialize values in object code	2-15
SNAP	Dump and store SXL symbol table	2-17
TRANSFER	Perform temporary transfer to a new command input file	2-18

INPUT FORMATS

RTE SXL accepts directives from a "command file" rather than from switch registers or "GO" directives. This file may be created off-line. Command statements resemble a primitive job control language and consist of two types: statements that are immediately executed and statements that merely set flags, modes, and bounds for later action. Any statement which sets a flag, a bound, or a mode leaves that mode in effect until it is specifically changed by later entry of a similar statement within the command file. The command file may be entered either by disc file or through any other standard ASCII input device.

SYNTAX PUNCTUATION SYMBOLS

<u>Symbol</u>	<u>Function</u>
<i>UPPER CASE ITALICS</i>	These words are literals and must be specified as shown.
<i>lower case italics</i>	Symbolic representations indicating what type of information is to be supplied. When used in text, the italics distinguish them from other textual words.
[]	The syntax structure enclosed within brackets is optional. Its inclusion implies something semantically; but syntactically, the statement is acceptable in either form.
<i>item 1</i> <i>item 2</i> <i>item 3</i>	One of the vertically stacked syntax structures must be used, but it is optional which one is chosen to be used.
*	An asterisk (*) in column 1 of the input command line begins a comment which continues to the end of the line. The line is echoed, if the ECHO mode is on; but otherwise, it is ignored.

(Continued)

<u>Symbol</u>	<u>Function</u>
&	An ampersand character (&) continues the statement onto the following line. All characters following an ampersand on the same line are ignored. An ampersand can occur anywhere in a statement, except within a symbol.

<u>Symbol</u>	<u>Function</u>
\$	An indirect operator must be used to reference the contents of a location in the program unit. The indirect operator may only specify "writes" into a file. Parentheses must surround the object of the indirect. Legal only in the SET statement.

KEYWORDS

The description of the RTE SXL command language which follows makes use of a number of keyword abbreviations. For readers not familiar with these terms, they have been summarized and listed below. Terms of a more general nature can be found in the Glossary.

<u>Keyword</u>	<u>Meaning</u>
BPLOCC	Base Page Location Counter. Current location counter for base page. (Similar to LOCC.)
FWABP	First Word Available Base Page. This item is used by SXL to contain the first word of the Base Page area for the relocatable code. Defaults to the value of BKDR area of the host system.
FWAC	First Word Available Common. Used by SXL as a pointer to the first word of the Common area in a target computer.
FWAM	First Word Available Memory. Used by SXL as a pointer

(Continued on next page)

Keywords (Continued)

<u>Keyword</u>	<u>Meaning</u>
FWAM (Cont)	to the first word of main memory in a target computer.
LOCC	Current Location Counter for SXL. It is used to determine the starting address for the relocation of each module. If not previously given another value, LOCC is given the current value of FWAM when the first module is relocated. After each module is relocated, LOCC is incremented by the length of the program plus primary and secondary linking areas (if any) to point to the next available core location.
LWABP	Last Word Available Base Page. SXL pointer to the last word of the base page of a target computer.
LWAC	Last Word Available Common. Pointer to the last word of Common area for a target computer.
LWAM	Last Word Available Memory. The last word of main memory for a target computer.
NEXTPAGE	The address of the next page after the current one; the contents of NEXTPAGE = ((LOCC AND 76000 ₈) + 2000 ₈)
?XFER	Program Unit Transfer Address, used by the operating system to jump indirectly to program control. At the end of loading, the contents of this pointer are used to catalog the program unit into the disc system.
LIBRY	A "reserved word" (a word that cannot be used as a file name for any user file) file name which represents the RTE SXL relocatable library.
L.G	A "reserved word" file name which represents the Load-and-Go area of memory.

COMMAND LANGUAGE

The following detailed descriptions of each of the RTE SXL operator commands are presented in alphabetical sequence for ease of reference.

ABORT

Purpose:

Terminate SXL execution if any error occurs. The second form below causes an immediate abort if any undefined symbols exist when an END statement is encountered.

Format:

ABORT IF	ERROR
	UNDEFS

COMMENTS

This statement sets a flag to terminate RTE SXL execution under the conditions stated under "Purpose". The default is both flags "OFF".

This command is used in the case when no operator is monitoring RTE SXL and its prompt withdrawal from the scheduled list, if any errors occur, is highly desirable.

If an ABORT flag is not set and an error occurs, the console device is placed on the command-files stack for corrective dialogue.

When it is undesirable to proceed further, orderly termination of RTE SXL can be accomplished by entering "ABORT IF ERROR", then entering any single character. This technique causes an unrecognized statement error.

If the decision to abort is taken, "SXL ABORTED" is printed following the printing of the command input line and error message.

BOUNDS

Purpose:

Describe the core bounds for relocated program at a target computer.

(Continued on next page)

BOUNDS (Continued)

Format:

BOUNDS FWAM = *number*, LWAM = *number*, &
FWABP = *number*, LWABP = *number*, &
FWAC = *number*, LWAC = *number*

Where:

FWAM is the address of the first word of main memory. The default value is the bound of the host machine's background disc-resident area (BKDR).

LWAM is the address of the last word of main memory. The default value is the bound of the host machine (BKDR).

FWABP is the address of the first word of the base page. The default value is the bound of the host machine (BKDR).

LWABP is the address of the last word of the base page. The default value is the bound of the host machine (BKDR).

FWAC is the address of the first word of the Common area. The default value is the address supplied by the first module loaded which declares a value for FWAC (background Common).

LWAC is the address of the last word of the Common area. The default value is determined by the size of the first module loaded which declares a value for FWAC (BK).

COMMENTS

The default values for this statement describe a background, disc-resident program, except that COMMON defaults to the program's own core area (rather than background COMMON).

Memory overflow errors occur when any part of the relocated program, including base-page data and links, is forced outside the limits set in the BOUNDS statement parameters. Code may reference symbols outside the bounds and symbols may be defined to be outside of these bounds without causing errors.

EXAMPLE

The following are typical bounds for 8K of memory:

FWAM = 2000, LWAM = 17677, FWABP = 2, &
LWABP = 1777

DISPLAY

Purpose:

Allows on-line user to obtain symbol table information.

Format:

DISPLAY TABLE
UNDEFS
symbol name [ON *logical unit number*]
bounds keyword

Where:

TABLE = displays the entire contents of the symbol table of all currently defined entry points and their values in the relocation sequence.

UNDEFS = displays the list of currently undefined symbols in the symbol table and the module name that first referenced each.

symbol name = if this name is a currently defined entry point, its value is printed. Otherwise, "UNDEFINED" is printed.

bounds keyword = displays keywords and their values.

logical unit number = An optional logic unit number may be specified for the print device to be used. The default for this operand (logical unit number = 1) is the console device.

ECHO**Purpose:**

To print or suppress echo of RTE SXL commands.

Format:

ECHO	ON <i>logical unit number</i>
	OFF

Where:

ON = Turns on the echo mode. If the echo mode is "ON," command lines are printed on the echo logical unit as they are encountered. Echo commands are printed preceded by the command identifier character.

OFF = Turns the echo mode flag off.

logical unit number = Commands will be echoed on this logical unit.

COMMENTS

Any remaining undefined symbols are listed on the core map device. End-of-loading is "forced," which sets undefined names to address zero.

The relocatable library and the Load-and-Go area are not automatically scanned. They must be specifically referenced in RELOCATE or SEARCH statements under the special, reserve file names LIBRY and L.G, respectively.

LINKS IN**Purpose:**

Define the SXL linking mode.

Format:

LINKS IN	BASE	[,DIRECT]
	CURRENT	

Where:

BASE = The Base-Page linking mode is the default mode. All links are placed in the base page.

CURRENT = The Current-Page linking mode causes SXL to place entry point links in the base page until this area becomes full, then current page areas are used. Page crossings within a single module (i.e., non-external reference instructions) generate links on the current page, if possible; otherwise, they are generated on the base page.

DIRECT = If this option is specified, reference instructions to external symbols are made directly whenever possible (i.e., if a link is not required, it is not used). If not given, then a link will be used for each instruction (or DEF) referencing as external symbol. This is true for DEF's as well as memory reference instructions.

COMMENTS

The ECHO mode is initially set as parameter *p2* when RTE SXL is turned on:

ON, SXL, *p1*, *p2*, *p3*, *p4*

If *p2* is zero or is not given, the echo mode is off. Otherwise, the mode is on and *p2* is used as the logical unit number.

The ECHO command is provided as a separate RTE SXL statement, so that the mode can be changed at any time. Echo mode provides documentation of the relocation procedure. Liberal use of comments can describe each step.

EXAMPLES

```
ECHO ON 6
ECHO OFF
```

END

Purpose:

Terminate command input phase of the relocation process, format the output file as specified in the OUTPUT statement.

Format:

END

COMMENTS

The DIRECT option can be specified at any time. RTE SXL links forward references directly or indirectly,

RTE SXL

according to whichever mode is specified when the forward reference is encountered. All subsequent references to a symbol already defined are linked according to the most recently specified mode.

NOTE

If there are any privileged routines in the RTE system core-resident library (including EXEC, \$LIBR, and \$LIBX) on the same page as any instruction which references them, do **not** use DIRECT. When LOCC is not on the same page as any core-resident system routine, the DIRECT option may be used freely.

When a reference is encountered to a symbol as yet undefined, a link is "allocated" in anticipation. If the link is not actually needed, it is returned for use by other page crossings.

Primary and Secondary linking areas are created only for current page linking mode.

A "primary links area" is a block of core allocated before attempting to load a module which appears to cross a page boundary. A "guess" area is created, starting at the current value of LOCC (see "c" in Figure 2-1). The initial size of the Primary Links area is derived from the size of the module by a "guess," calculated as follows: $(\text{MIN}(A2,B))4$. The size of the primary links area may be specified in a RELOCATE or SEARCH statement instead of being calculated. ALGOL programs do not have length information in the NAM record, and so are not given Primary Link areas unless the area is specified in a RELOCATE or SEARCH statement.

The location counter is increased by the size of the primary area, and the module is relocated using the updated value of the location counter (see *cl* in Figure 2-1). In this way, two areas are created, one on either side of the page boundary, from which links can be allocated.

The "primary" links area is used for off-page references on Page *x* in Figure 2-1. The secondary area is used for off-page references on Page *y*.

RTE SXL makes a second pass over any module which does not completely use its primary links area for linkage, shortening the area by the number of unused words in this area.

The secondary links area is allocated after the last data area of the module. References from Page *y* (in Figure 2-1) to any other page generate links in the secondary area. LOCC is incremented by the size of the program plus the size of the secondary areas, before the next module is relocated.

LINKS START AT

Purpose:

This statement is used to predetermine the start of the Base-Page link area and/or initialize values in it.

Format:

LINKS START AT *starting address* [,*link value list*]

Where:

starting address =

The octal value of this address must be greater than or equal to the contents of BPLOCC.

link value list =

This list is composed of integers or entry point names separated by commas.

COMMENTS

The link value list has two distinct formats: integers and symbols. Integers and symbols cannot be mixed in the same statement. The form with constants tells RTE SXL it may assume a link exists in the base page at the specified value, at the address given (if more values are given, they are assigned at successive address). RTE SXL will not place these values in the output file. This saves generating extra links in the output file.

The form when the list is composed of symbol name is used to tell RTE SXL that it is to store a link to that symbol (which is entered in the undefined list) at the specified starting address. If more symbols are in the list, links to them will be stored at successive locations. If any of these symbols are still undefined when a SEGMENT or END statement is entered, a zero will be stored in the link location. This allows the preassignment of base page links, and the generation of modular software.

EXAMPLES

```
LINKS START AT 100, IOC., DMAC1, DMAC2  
LINKS START AT 45, 2000, 3000
```

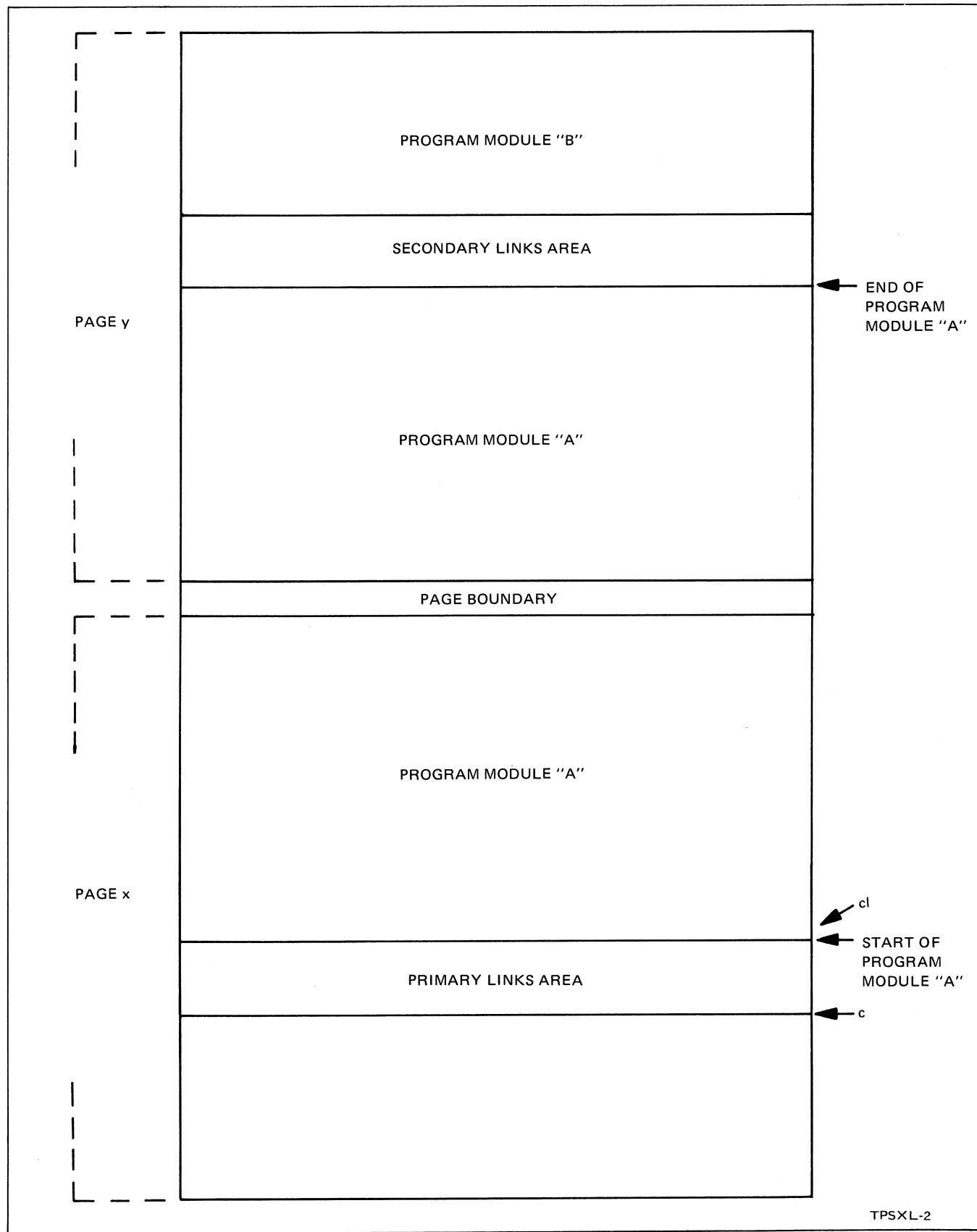


Figure 2-1. Current Page Mode Link Areas

MAP**Purpose:**

This statement turns on the listing flags for the core map printout.

Format:

MAP	MODULES
	FILES
	GLOBAL [ON <i>logical unit number</i>]
	LINKS
	ALL
	OFF

Where:

MODULES	Module names and core bounds are printed under the heading for "MODULES".
FILES	File names are printed in the column for "FILES".
Globals	Entry point names and core addresses are printed under "GLOBAL VARS". If there was a forward reference to this entry point, the name of the module which first referenced it is printed under "REF. BY".
LINKS	In current page linking mode, the core used by the primary area (or a blank line if none) is printed, followed by the secondary area bounds (if any). These are the areas actually used, and may differ from the space allocated due to the algorithm used. The link table is also printed at end-of-loading.
ALL	Turns on all options.
OFF	Turns off all options.
<i>logical unit number</i>	An output file name may be specified. The default value is the standard list output device (logical unit = 6).

COMMENTS

When processing for this statement begins, all flags are turned off, so that only those flags requested will be on. The first MAP statement causes a page eject and header to be printed.

INFORMATION CONTAINED IN THE CORE MAP

The core map heading appears at the top (see Figures 2-2 and 2-3). "File name" refers to the relocatable binary file. "Program module" refers to the name of the module, and appears for each module. "GLOBAL VARS" refers to "global" symbols (symbols known to a relocating loader, as opposed to "local" symbols which are those variables whose names have meaning within a module but are unknown outside that module) and to which references may be made in other modules by declaring the symbol EXTERNAL (FORTRAN) or EXT (Assembly). These names consist of subroutine and function names in FORTRAN modules, and names declared as ENT (entry) symbols in Assembly modules. "Entry point" is a misnomer, since in the general case, global symbols can be constants, arrays, real or integer variables, or address constants (DEFS), as well as subroutine and function names.

MODULE CORE BOUNDS INFORMATION

There are four core areas of interest for each program module, printed only if the "MODULES" core map flag is set:

1. Memory allocated to the module's instruction and data area (exclusive of base page), including BSS area
2. Base page area, including ORB data and instructions and link area
3. Primary links area (printed only if in current page linking mode)
4. Secondary links area (printed only if in current page linking mode)

Item 1 is printed under the headings "FIRST ADDRS" and "LAST ADDRS".

Item 2 is printed under the heading "BASE PAGE".

Items 3 and 4 are printed on separate lines under the heading "FIRST ADDRS." and "LAST ADDRS.". The primary area is printed first, unless there is none. When there is no primary area, one line is left blank, followed by the secondary area. If neither area exists, neither line is printed.

Current page linking area information is of particular interest when it is desirable to conserve memory. (See the discussion on current page linking, under "LINKS IN . . ." command.) The values printed for each area represent the area actually used. Wasted core in the primary area is indicated by the difference between the start of the module and the end of the guess area; secondary area waste is indicated by the start of the secondary area and the end of the program module. This latter waste occurs exclusively in FORTRAN modules which don't cross a page boundary (hence, no second pass).

GLOBAL VARIABLES INFORMATION

Each global symbol is given a value either by appearing in an ENT-type relocatable binary record, and relocated by SXL or by the user via the SET statement. Its value is printed (if the "GLOBALS" map flag is set) under the heading "GLOBAL VARS" as it is defined. If the symbol was in the undefineds list, the name of the module which first put it there is printed under the heading "REF. BY". This is particularly useful when adapting software to another application (for example, to remove the formatter to save core).

LIBRY

When the system relocatable utility library (LIBRY) is searched, names of core-resident library routines are defined, as are EXEC, \$LIBR, and \$LIBX. These names are printed (if the GLOBALS flag is set) before opening the library file. When the library search is over, the list of core-resident entry points is searched again for matches in the undefineds list. \$LIBR and \$LIBX often are required by library routines.

LINK TABLE

At end-of-loading, or after SEGMENT statements, SXL prints the address of base page links and the name of the symbol they correspond to (if the Links flag is on). This is of use when debugging a program; it is helpful to make patches if the link address is known.

Following the link table (if any), a composite table is printed (if any map option flag is set) containing core used (including base page), free core, the value of the transfer address and the name of the module which declared it, and the name of the output file. The message "\$END SXL" is printed on the console (regardless of the map flags conditions) to record this fact. If any map option is set, it is printed on the map file as well.

EXAMPLES

```
MAP ALL ON 6
MAP MODULES
MAP MODULES, Globals
MAP GLOBALS, MODULES, LINKS ON 6
```

OUTPUT

Purpose:

This statement defines an OUTPUT format for the relocated code, and may optionally direct RTE SXL to store the code in a particular file, on a particular disc (or device). This request also allows the user access to program parameters and to force programs into the foreground.

Format:

OUTPUT	ABS CI [ON <i>file name</i> [<i>p1</i>] , <i>p2</i> , <i>p3</i> , & FGR <i>p4</i> , <i>p5</i> , <i>p7</i> , <i>p8</i> , <i>p9</i> , <i>p10</i>]]
--------	--

Where:

p1 through *p10* are optional integer parameters.

ABS = A b s o l u t e C o d e f o r m a t .
BBL-compatible absolute format is necessary in distributed systems, and allows convenient backup to be made for other systems. This format must be used for programs that are to run on terminal computers.

CI = Core Image Output format. The default format in RTE, it is used for programs which are to run in RTE computers. Core image files are written to the output file in blocks of 128 words. No checksum or parity bits are possible. Core bounds and start addresses for the program unit are included in the catalogue entry array passed to the File Manager. The disc area is zeroed before loading, so all BSS's initially contain zero (except first and last BSS's in the program).

FGR = Foreground format. Its use implies that the program is to be core image and that it will be located in the foreground real-time area of the central computer. The initial values for FWAM, LWAM, FWABP, LWABP, FWAC, and LWAC may be overridden by a BOUNDS statement entered later. The type parameter (*p3*), if entered, is ignored (type 2 is assumed). Parameters not given will be set to their default values.

(Continued on next page)

RTE SXL

FILE PROGRAM GLOBAL FIRST LAST BASE PAGE REF., BY
 NAME MODULE VARS. ADDRS. ADDRS.

 --OUTPUT CI ON LOADRC CR 2 SCODE 0
 --LINKS IN CURRENT,DIRECT
 --SET LSTRF TO FWAM
 LSTRF 040000
 --RELOCATE LOADR,
 LOADR.
 LOADR 040046 046616 000400 001015
 040000 040043
 046617 046632
 --SEARCH LIBRY
 LIBRY
 EXEC 002000 LOADR
 \$LIBR 002200 LOADR
 \$LIBX 002307 LOADR
 LIBRY
 --SET \$(LSTRF+220) TO LOCC
 --SET \$(LSTRF+221) TO LOCC
 --SET \$(LSTRF+222) TO LOCC
 --SET \$(LSTRF+223) TO LOCC
 ***** THE ABOVE FOUR STATEMENTS RE-DEFINE POINTERS
 ** WHICH THE LOADER USES TO FIND FREE MEMORY.
 - I
 -TR
 - *****
 -TR
 --DISPLAY UNDEFS
 --END
 NO UNDEFS
 LINKS TABLE
 TRANSFER TO LOADR 040272
 CORE USED 040000 046632 000400 001015
 FREE CORE 046633 077677 001016 001647
 OUTPUT FILE NAME:LOADRC
 END SXL
 -ABORT IF ERROR
 -ERROR
 -TR,CMD
 -
 --THIS COMMAND FILE RELOCATES THE RTE RELOCATING LOADER
 --(NOT SXL) IN CURRENT PAGE LINKING MODE.
 -- THIS IS NOT NORMALLY POSSIBLE WITH ANY PROGRAM
 -- WHICH HAS BEEN WRITTEN TO USE THE MEMORY BETWEEN
 -- ITS LAST ADDRESS AND THE END OF AVAILABLE MEMORY.
 -- HOWEVER, BY OVERLAYING POINTERS WITHIN THE CODE OF
 -- THE RELOCATED PROGRAM, ONE CAN CHANGE THESE POINTERS
 -- TO "TRUE" FREE MEMORY.
 --

 --MAP ALL

Figure 2-2. Typical Core Map

```
FILE PROGRAM GLOBAL FIRST LAST      BASE PAGE    REF.BY
NAME MODULE VARS, ADDRS, ADDRS,
-----
** THIS EXAMPLE RELOCATES THE SAME RTE LOADER, WITHOUT CURRENT
** PAGE LINKING MODE.
** NOTE THAT THERE IS VERY LITTLE SAVINGS IN CURRENT PAGE LINKING
** FOR LARGE PROGRAM MODULES THAT CROSS SEVERAL PAGE BOUNDARIES.
**
-OUTPUT CI ON LOADERE
-RELOCATE LOADR.
LOADR,
      LOADR      040000  046550  000400  001071
-END
UNDEFS
  EXEC   LOADR
  $LIBR  LOADR
  $LIBX  LOADR
LINKS TABLE
TRANSFER TO LOADR  040224
CORE USED    040064  046550  000400  001071
FREE CORE    046551  077677  001072  001647
OUTPUT FILE NAME:LOADER
END SXL
```

Figure 2-3. Core Map Without Current Page Linking

RTE SXL

OUTPUT (Continued)

file name = Used to direct SXL to store the relocated code at a particular file, on a particular disc (if the cartridge number is given), or on a particular device. The default value of this parameter is the file name of the main module.

p1 = Cartridge Identifier (must be an integer). If the cartridge number is given, it is used as a cartridge identifier (RTE File Manager) number to reference the appropriate cartridge. The default for the cartridge identifier (0) is the first disc in the File Manager's list.

p2 = Output File Security Code. Legal values for this parameter are the standard File Manager Security Codes. The default value is 0, for no security code.

p3 = Program type. Values are 2 for real-time disc-resident programs and 3 for background disc-resident programs. Other types may be given as required for programs which are to execute in another computer. Default value = 3.

p4 = Priority (1-99). Default value = 99.

p5 = Resolution code is the unit in time to be multiplied by the multiple execution interval value (MPT) to get the total time interval.

- 1 - tens of milliseconds
- 2 - seconds
- 3 - minutes
- 4 - hours

p6 = Execution multiple. This is a number from 0 to 999 which is used with *p5* to give the actual time interval.

OUTPUT (Continued)

p7 = Hours
p8 = Minutes
p9 = Seconds
p10 = Tens of milliseconds

} Sets absolute start time

NOTE

See the IT command in the RTE manual for details on how parameters *p5* through *p10* are used.

COMMENTS

A cartridge number cannot be used in place of a logical unit number for a non-disc file. The file name in this case should be the desired logical unit (LU). If the disc is not mounted, SXL prints an error.

Only one output file is allowed. If the Output statement is entered more than once, the first file entered is used. SXL does not create copies of object code files. To create backup copies, a disc file should be used for the initial SXL output and copies made from it later.

RELOCATE

Purpose:

Direct SXL to open each named file, and unconditionally load each relocatable module in the file.

Format:

RELOCATE

REL *file name* [[EXCEPT] *module name list*] [,AT *target load address*] [,LINKS] = *number*.

Where:

REL = Acceptable abbreviation for RELOCATE.

(Continued)

(Continued on next page)

RELOCATE (Continued)

<i>file name</i> =	SXL RELOCATEs only these files. Integers are acceptable as file names and are considered to be logical unit numbers.
EXCEPT <i>module name</i>	
<i>list</i> =	The module names appearing after EXCEPT are excluded from relocation. If EXCEPT is not used but module names are given, only those modules will be relocated.
target load <i>address</i> =	If a target load address is given, SXL forces the first module loaded within each file to start at the indicated address.
LINKS <i>number</i> =	If the keyword LINKS appears, preceded by a non-negative integer, the current-page linking mode is implied. A primary links area is created, which is the given integer number of words long.

COMMENTS

There are no default RELOCATE files.

The examples show the kind of identifiers that can be used for module names and the sample format of the RELOCATE statement.

The reserved file name L.G is assumed to mean the Load-and-Go area. LIBRY is reserved to mean the system relocatable library.

EXAMPLES

Names
.IOI.
A3
LIBRY
?EN1

RELOCATE Statements

RELOCATE L.G (MOD1, MOD2, & MOD2,) AT 2000, LINKS = 30
RELOCATE MOD 44
RELOCATE L.G (EXCEPT MOD9)
REL 5, 5, 5

SEARCH

Purpose:

Selective loading of files, or modules within files, on a "library load" basis. Modules are loaded only if they declare an entry point name which matches a previous "forward reference" name.

Format:

SEARCH *file name* [EXCEPT *module name list*] [AT *target address*] [, LINKS] = *number*.

Where:

file name = If specific file names are given, SXL SEARCHes only these files.

EXCEPT
module name

list = The module names appearing after EXCEPT (if any) are skipped.

target
address =

If a target address is given, the module is relocated starting at that address. If no module is relocated, the location counter is the target address (if given), or its value before the statement was entered.

LINKS
number =

If this parameter is specified, current page linking mode is implied, and the specified number is used as the size of the guess area for the first module relocated.

COMMENTS

RTE SXL does not rescan files mentioned in SEARCH statements when undefineds are left at the end of the relocatable input phase. If undefineds still exist which are defined in one of the files already SEARCHed, that file may be mentioned again in another SEARCH statement; or it may be specifically handled for unconditional loading. The following example shows the SEARCH statement used to load a large relocatable file containing several modules:

EXAMPLE

In the following example, user control file input is in italics; SXL output is in capitals.

```
REL MOD1
SEARCH MOD2
DISPLAY UNDEFS ON 1
UNDEFS
E1
E2
E3
SEARCH MOD2
END
NO UNDEFS
```

SEGMENT

Purpose:

To allow the creation of segment overlays.

Format:

```
SEGMENT
TR, name
```

Where:

name is the name of the snapshot file.

COMMENTS

To relocate a segmented main program, the user must first relocate the main program. When this process is completed, the user may optionally save the state of the symbol table by "taking a snapshot" of it (i.e., by using a SNAP statement). The relocation of each segment overlay must

then be separated by the two statements shown under "Format" above. After completion of the last segment, an END statement is entered. Thereafter, if a change is required to one segment, only that segment need be relocated.

Any OUTPUT statement appearing within the scope of a SEGMENT statement (which lasts until the next SEGMENT statement is entered) defines the output file which is to be used for the segment's object code. If none is given, the file is defaulted to be the name of the segment. If a logical unit is specified, it remains the same for the entire run (segments and main).

The loading of each segment proceeds independently of the others, and relatively independently of the main, except that references are permitted from the main into a segment, and vice versa.

Undefined externals may exist when a SEGMENT statement is entered. SXL assumes they are to be defined in subsequent statements. They are listed.

A SEGMENT statement must be entered before loading each segment for the following reasons:

- To allow for the case where several segments have a copy of a particular routine, which is load at different addresses in each segment.
- To allow the main to reference names in the segment overlay area when the routines are contained in type 5 (Segment Main) modules. It also allows the reference of type 7 (subprogram) modules with forcing the modules to be loaded in the main area.
- To allow changes in a segment without reloading the main as well. This is contingent upon the symbol table for the main being saved via the SNAP statement. The symbol table is restored via the TR statement after each SEGMENT statement (no additional SNAP statements need be used here).

EXAMPLE

The following example illustrates how a segmented main is initially constructed.

A main program to be called XTEST is stored in relocatable form in file FTEST and is to be relocated in current-page mode. The main is to include subroutines TEST1, TEST2, and TEST3. There are references in the main to TEST1, TEST2, and SEG1. Two segments are to be loaded: SEG1 and SEG2, from file FSEG1 and FSEG2, respectively. The main is to be loaded into a file called XTEST; the segments are loaded into XSEG1 and XSEG2.

The command file for this operation is shown below:

<u>Statement</u>	<u>Meaning</u>
-ECHO ON 1	Commands will be echoed on logical unit number 1.
-LINKS IN CURRENT	Set linking mode to use current page links.
-OUTPUT CI ON XTEST	Define main's output file.
-RELOCATE FTEST	Relocate file FTEST.
-SEARCH LIBRY	Satisfy externals from the RTE system library.
-SNAP ON XSNAP	Save symbol table, to be restored for each segment.
-SEGMENT	Begin relocation of segment SEG1 and clear symbol table.
-TR, XSNAP	Restore symbol table.
-OUTPUT CI ON XSEG1	Define output file for SEG1.
-SET	
-RELOCATE FSEG1	Relocate segment SEG1's main.
-SEARCH LIBRY	Satisfy externals from system relocatable library.
-SEGMENT	Begin relocation of SEG2 and clear symbol table.
-TR, XNAP	Restore symbol table
-OUTPUT CI ON XSEG2	Define output file of segment.
-RELOCATE FSEG2	Relocate segment SEG2's main.
-SEARCH LIBRY	Satisfy externals.
-END	

SET**Purpose:**

Defines entry point names, places values in the object code, defines keyword values, or deletes a symbol.

Format:

SET *symbol* UNDEFINED
 SET\$(*expression*) TO *expression*
 SET *symbol* TO *expression*

Where:

SET *symbol*
 UNDEFINED = The symbol becomes undefined and is free for later redefinition.

SET (Continued)

SET = Creates or defines entry points, or initializes values in object code. SET will allow the redefinition of any symbol, including any bounds keyword.

\$ = Indirect operator symbol. This symbol must be used to reference the contents of a location. The object of the indirect must always be surrounded by parentheses to avoid ambiguity with symbol names that begin with a dollar sign.

expression = An expression may consist of any combination of symbols, integers, and keywords, separated by a plus sign (+) or a minus sign (-).

(Continued)

(Continued on next page)

SET (Continued)

symbol = Except for the keywords listed below, symbols represent their addresses rather than the contents of the addresses.

When the following keywords are used as expressions or symbols with the SET command, their meanings are as follows:

LOCC = Current Location Counter, used to determine the start-load address for each module as it is loaded. If not previously given another value, LOCC is given the current value of FWAM when the first module is loaded. After relocating each module, LOCC is incremented by the length of the program, plus primary and secondary areas (if any), to point to the next available core location.

BPLOCC = Base Page Location Counter, used for loading base page relocatable data into the base page (operates similarly to LOCC). Base-page links are allocated and intermixed with ORB data, starting with BPLOCC and working toward LWABP. Unless previously given some other value, BPLOCC is given the current value of FWABP when the first module is relocated.

?XFER = Program Unit Transfer Address, used by the operating system to jump indirectly to program control. At the end of loading, the contents of this pointer are used to catalog the program unit into the disc system. If, for example, SXL is running in RTE to create a coreload to a remote terminal and it is desired to allow execution of the relocated program to begin at location 100 (octal), then the following statements may be used:

(Continued)

SET (Continued)

SET \$(100) TO 124101
SET \$(101) TO ?XFER

If no main program module is found during relocation (i.e., no transfer address is given in any END record) then the user can define ?XFER by use of the SET statement. If no address is found when END is entered, RTE SXL will specifically request definition for ?XFER and will reject the END statement. The user can set ?XFER or relocate the main module, as desired.

NEXTPAGE = The address of the next page after the current one; the contents of NEXTPAGE = ((LOCC AND 76000 (octal)) + 2000 (octal)). This allows the user greater control over the generation of links due to page crossings within a module. He may, for instance, set LOCC to NEXTPAGE to avoid page-crossing references for a particular module.

COMMENTS

The SET statement provides the user with the facility to create entry point names and define them, or to place data in the object code. Thus, configuration routines can build tables in the target machine's absolute output file, without requiring extra code to convert this information to relocatable.

If the SET UNDEFINED form is used, then the symbol becomes undefined and is free for later redefinition. This provides an error-check to prevent duplicate entry points from causing errors in the object code. "Undefining" a symbol effectively erases the symbol. It does not enter the undefined symbols list.

Code produced via the SET setatment can force data outside of these boundaries without causing errors.

The SET statement can be used to define BOUNDS keywords as expressions.

SET allows the redefinition of any symbol, including the target machine bounds keywords, as a convenience to a certain class of advanced users. Users should be aware that once the first module has been loaded, LOCC and BPLOCC operate independently of FWAM and FWABP, respectively. The current values of target machine bounds will be used to determine if any data is being relocated outside of available core (an error condition).

CAUTION

The SET statement has the capability of creating object code to be stored in addresses not necessarily within the bounds of the target machine. Configuration routines can make use of this fact to build various tables and protect them from overlay by the relocated code.

EXAMPLES

The following examples show the multiple uses of the SET command and their corresponding formats:

Example 1.

```
*LEAVE A SPACE OF 5 WORDS AS "COMMON."
SET LOCC TO 2005
SET FWAC TO 2000
SET LWAC TO 2004
```

Example 2.

```
*BEGIN NEXT MODULE 3 WORDS INTO THE
NEXT PAGE
SET LOCC TO NEXTPAGE+3
```

Example 3.

```
*STORE A JMP 3, I IN 2, AND STORE TRANSFER
ADDRESS.
```

```
SET $ (3) TO .XFER OR: SET $ (2) TO 124003
SET $ (2) TO 124003 SET $ (3) TO ?XFER
```

NOTE

When the output format is absolute binary, a JMP 3, I is stored in location 2 and the transfer address is stored in location 3. This is done automatically.

SNAP

Purpose:

This statement allows a portion of the SXL symbol table to be dumped and later restored for another loading operation through the regular command input file.

Format:

```
SNAP [ON file name]
```

Where:

The default file is the standard punch. Information in the symbol table is dumped in ASCII in a format acceptable as command file input. Only the current values of the BOUNDS keywords, LOCC, BPLOCC, the base-page links, strings, and entry points are dumped.

COMMENTS

The SNAP statement may be used to dump a symbol table. Once a symbol table is "dumped" it can be "restored" by transferring to the file which contains the output of the SNAP statement.

The symbol table is dumped immediately upon receipt of the SNAP command. Only those entry points currently defined are dumped, which allows dumping of a partial symbol table. Unwanted entry points may be deleted by preceding SNAP with SET ... UNDEFINED.

Duplicate entry points errors can occur if a name appears both in a SNAP file and in an ENT record. Names may, however, be deleted by the SET *name* UNDEFINED command.

The SNAP statement may be used by an RTE system at a host computer to relocate program units for execution in a remote terminal (target machine). The entry points, target-machine boundaries, and base page links used by the resident are then known to SXL for the program unit being loaded.

The use of the snapshot file is illustrated in the preparation of BCS and RTE-C programs which are covered in Sections V and VI.

EXAMPLE

```

BOUNDS FWAM=2000, LWAM=37677 &
FWABP=24, LWABP=1777, FWAC=0, &
LWAC=0
SET LOCC TO 14700
SET BPLOCC TO 30
SET .MEM. TO 14333
SET HALT TO 14333
SET XEQT TO 14603
SET XSQT TO 14602
LINKS START AT 24, 14333, 14602, 114602

```

All numbers are in octal. Entry point names include those defined in relocatable ENT records, as well as those defined in SET statements.

TRANSFER (Continued)

Format:

TRANSFER [, file name]
TR

Where:

TR = Acceptable abbreviation for TRANSFER.

file name = Name of the file to be transferred.

COMMENTS

The new file is used until it is exhausted or until an END statement is encountered.

Each TRANSFER statement “stacks” the last old file (last in, first out) before opening a new file. When an EOF/EOT is encountered, or if no file name is given in the statement, the current file is closed and the previous file is obtained by “unstacking.”

The TRANSFER statement can be used to save repeatedly entering the same SXL directions each time a program is to be relocated. This statement can also be used to restore the symbol table from a snapshot file.

TRANSFER

Purpose:

Performs a temporary transfer to a new command input file.

(Continued)

SECTION III OPERATING PROCEDURE

The Real-Time Executive System Cross Loader (RTE SXL) contains the necessary interface software to function with the Real-Time Executive (RTE) disc-based operating system. RTE SXL has the capability of functioning either as a replacement for the RTE Relocating Loader (the loader which is a standard part of the RTE package) or in conjunction with the RTE Relocating Loader. In other words, the RTE Relocating Loader (HP Part No. 20792-60001) may either be left in the RTE package or removed from the RTE package when RTE SXL is added.

RTE SXL can be turned on either by using the RTE system command, "ON, SXL", or by internal scheduling with the user's program.

ON, SXL

Purpose:

To turn on RTE SXL by RTE system command.

Format:

ON, SXL, *p1, p2, p3, p4*

Where:

p1 = The command file logical unit number. If *p1* = 1, a prompt character is printed on the system TTY, and a bell is rung (to get the operator's attention), when SXL is ready for the next source line. (Default is 5.)

p2 = Echo output Logical Unit. (If zero or not given, echo mode is turned off.)

p3 = List output Logical Unit. (Default is 6.)

p4 = Punch output Logical Unit. (Default is 4.)

NOTE

P1 through *p3* can be a command file name; so another program can schedule SXL with a command file. (In this case, *p4* is not used.)

RTE SXL SCHEDULE

Purpose:

To schedule RTE SXL for execution from a dormant status.

Format:

```
DIMENSION INAME (3)
DATA INAME /2HSX, 2HLA, 2HAA/
ICODE = 9 (wait) or 10 (no wait)
CALL EXEC (ICODE, INAME, IPRM1 . . . IPRM4)
```

COMMENTS

The format shown above for scheduling RTE SXL from a dormant status is coded in FORTRAN. IPRM1 . . . IPRM4 are the parameters equivalent to *p1* through *p4* in the RTE system command format. For more detailed information on the use of the program scheduling technique, refer to "Program Schedule" in Section III, EXEC Calls, in the Real-Time Executive Software System (RTE) Programming and Operating Manual (Part Number 02005-90001).

At this point, RTE SXL is ready to accept command file input. This input should be introduced in two parts or phases: (1) Target Machine Configuration and (2) Relocation Commands.

PHASE I. TARGET MACHINE CONFIGURATION

RTE SXL must be told certain things about the configuration of the computer in which the relocated code is to be run, unless it may assume that the host computer is to be the target computer. In the absence of data to the contrary, it will make this assumption for all decisions. SET or BOUNDS statements must be used to establish definitions that are to differ from the standard assumptions.

If the program is not to be run in one of the host computer's disc-resident areas, the absolute-binary output format should be used, specified by an OUTPUT statement.

The following is a list of the assumptions made by RTE SXL in the absence of other specifications:

Core bounds FWAM, LWAM, FWABP, LWABP, assumed to be the values for the Background Disc Resident (BKDR) area.

FWAM— First Word Available Memory; relocation of modules normally begins here.

FWABP— First Word Available Base Page. Assumed to be the first word available base page location for the BKDR area.

LWAM— Last Word Available Memory, assumed to be the last available address for BKDR.

LWABP— Last Word Available Base Page – assumed to be 1647 (octal).

COMMON The COMMON area pointers, FWAC and LWAC, default to the contents of LOCC when the first module declaring COMMON is relocated. Note that this means programs are each assumed to require a private area for Common. In order for a program to use the RTE Foreground or Background Common areas, the user must specifically set these pointers (SET or BOUNDS) to the proper value (unless the output format is "FGR", in which case the Real-Time COMMON area is used). BKDR programs which attempt to store data into foreground COMMON will cause a memory protect error; they may "read" data from there, however.

LOCC is moved up if COMMON is assumed. Figure 3-1 illustrates the situation for a program declaring 1024 words of COMMON in Module 1. Figure 3-2 shows what happens if only Module 2 first declares the same COMMON. (Core addresses are arbitrarily assumed for this example, program lengths are consistent in the two figures.)

Output File Format= defaults to core image.

Output File Name= defaults to the name of the first main program module relocated (i.e., one which contains a transfer address). Program parameters (type, priority, execution multiples, etc.) are taken from the first main program module relocated. Programs other than type 3 do not force other core boundaries than the default. For example, a type 2 (FGDR) will default to using the BKDR area, which will cause memory-protect violations when run.

If the output file is a core image disc file, the "RP" File Manager command must be used to make the program become known to the system. This allows programs to be packed on the disc, and allows better disc space recovery when they are purged.

Program Parameter Defaults: Type = 3
Priority = 99
Time = 0

Other Defaults: Core map option = OFF
Echo option = OFF
Linking mode = base page only (indirect: EXT/ENT matches always use a base page link).

PHASE II RELOCATION COMMANDS

This phase relocates program modules according to the options set (or defaulted) during Phase I.

Example 1 is an example of the commands necessary to relocate a program. The relocatables exist in four files, TESTR (the main program module), POWRR, FOURRR, and RFTR. In addition, the system library for the host machine is to be searched to resolve remaining undefined externals.

Example 1.

```
*ON, SXL, 1
-RELOCATE TESTR, POWRR, FOURRR, RFTR
-DISPLAY UNDEFS
UNDEFS                                         Computer prints
```

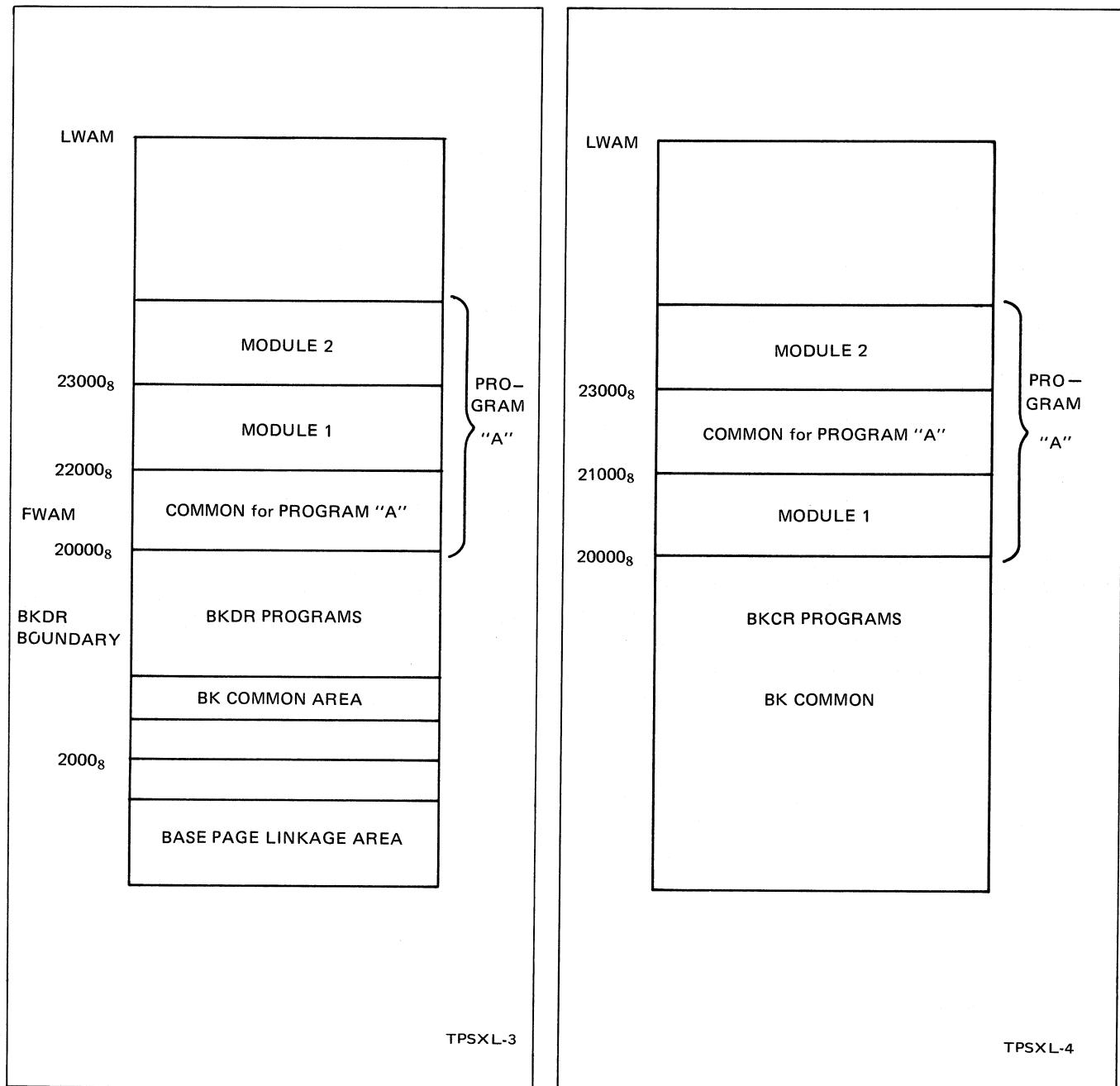


Figure 3-1. Module 1 Declares Common

Figure 3-2. Module 2 Declares Common

RTE SXL

Example 1. (Continued)

SIN TEST	Name of module which first required it in this column. Undefns (so far) in this column.
-SEARCH LIBRY	
-DISPLAY UNDEFS	Used to see if any undefineds exist.
NO UNDEFS	Printed if no undefineds exist.
-END	No more relocation is to be done.
\$END SXL	SXL terminates.
*ON, FMGR	
:RP, TEST	Commands to the File Manager.
:EX	
*ON, TEST	Turn on the program.

If current page linking was desired, the statements "LINKS IN CURRENT, DIRECT" could have been entered at any time. Subsequent relocation would then be done in current page linking mode, with direct* linking (i.e., EXT/ENT matches use a link only if a page is crossed). However, code relocated prior to a LINKS statement is completely unaffected.

If your wish to call the program by some name other than the name of the main program module, use the OUTPUT statement in Phase I:

OUTPUT CI ON *file name*

where

file name is the name you wish to call the program.

*Most code can be successfully relocated in "direct" mode. However, reentrant code and code on the same page as RTE system entry points (e.g., EXEC, \$LIBR, \$LIBX) and the core-resident library area will not run in RTE if the "direct" option is used. Drivers which use the load-time configuration feature of RTE SXL must use the "direct" linking mode.

ORDER OF COMMAND INPUT

Any of the commands that set flags can be entered at any time, and most can be changed at any time. The following command words are in this category:

MAP, ECHO, ABORT (may not be changed once entered), LINKS.

The BOUNDS statement must be given before any relocation takes place (i.e., before any RELOCATE or SEARCH commands are issued). The reason for this requirement is that the first BOUNDS statement initializes LOCC and BPLOCC (if they do not already have a value) to FWAM and FWABP, respectively.

The LINKS Creation Statement (LINKS START AT ...) must be entered before any RELOCATE or SEARCH commands. If the start address is to be a keyword, then that value must have previously been defined by a BOUNDS or SET statement.

The "immediate action" statements may be entered at any time. These are RELOCATE, SEARCH, SNAP, DISPLAY, SET, and SEGMENT.

The OUTPUT statement must be entered before any RELOCATE or SEARCH statements. It specifies the format of the program unit and gives it a name.

DISTRIBUTED SYSTEM USE

RTE SXL can be used to relocate code destined for execution in a machine other than the one in which SXL runs. When this is the case, the user must use commands to override the defaults.

CAUTION

It should be remembered that the system library for an EAU (Extented Arithmetic Unit) RTE machine cannot necessarily be used as the system library for your target machine in the relocatable binary file. For this discussion, we will assume this file is named BCSLIB. In addition, entry points must be defined for the remote monitor system.

OPERATING PROCEDURE

Example 2.

Relocate the same code used in Example 1 for use in a remote BCS terminal. Assume that the program TEST

makes only simple I/O calls to EXEC, and that an adapter routine named EXEC is available in the library file, so that it is meaningful to relocate this program for execution in a BCS system.

<u>Command</u>	<u>Purpose</u>
ON,SXL,1	Turn on SXL.
OUTPUT ABS ON TESTA	Programs to be communicated to a remote must be in absolute binary, BBL-compatible format. This file will be known as TESTA, to avoid duplicating the name of the program relocated in Example 1.
BOUNDS FWAM=2000, & LWAM=27677, LWABP=1777, & FWABP=30, FWAC=37700, & LWAC = 37677	Define core bounds for target machine.
SET .EQT. TO 33721 SET HALT TO 33701 SET XEQT TO 34156 SET .IOC. TO 33942	Define entry points in the resident BCS system of the remote terminal.
MAP ALL	Set memory map flags.
RELOCATE TESTR, POWRR, FOURRR, RFTR	Now relocate the code.
SEARCH BCSLIB	Satisfy externals.
DISPLAY UNDEFS	Check that all externals (EXTs) are satisfied.
NO UNDEFS	Printed if no undefineds (UNDEFs) exist.
END	End of relocation.

At this point, a file called TESTA exists on the disc. It can be punched if desired, or if the communication package is available, the remote terminal software may request transfer of the program for execution.

In this latter case, the core occupied by the program will be of some interest if the monitor is to keep itself from being

overlaid. These and other parameters are carried in 9 records at the end of the program, each carrying two data words, destined for core addresses 2, 3. The data exists in the following order, assuming the program exists in N (a finite number of) records:

RTE SXL

<u>Record</u>	<u>Address</u>	<u>Data</u>	<u>Default</u>
N+1	2	Word 1 of name	(First NAM encountered/
	3	Word 2 of name	type=3)
N+2	2	Word 3 of name/program type	(99)
	3	Program Priority	(0)
N+3	2	Time Res. Exec. Mult.	
	3	Spare Word	
N+4	2	Hours	(0)
	3	Minutes	(0)
N+5	2	Seconds	(0)
	3	Tens of Milliseconds	(0)
N+6	2	Low Main	
	3	High Main	
N+7	2	Low Base Page	
	3	High Base Page	
N+8	2	First Address of Common	(0)
	3	Size of Common	(0)
N+9	2	JMP 3,I	
	3	Transfer Address	

SECTION IV

ERROR MESSAGES

This section describes anticipated error conditions, resulting messages, and correction procedures for the Real-Time Executive System Cross Loader (RTE SXL). Table 4-1 shows a summary of all RTE SXL error messages. Error messages are output on the system teletype unless the output file is designated as the console. If an error abort flag is set (the command statement "ABORT IF ERROR" was input), SXL aborts after printing the error message. The ERROR ABORT statement preempts any possible keyboard dialogue, since its use assumes no possible operator intervention.

If an execution error occurs, execution is terminated at the point in the statement where the error is encountered. The statement should be re-entered, skipping files which have already been successfully processed. A "TR" is entered to return to the command file at the next line.

Table 4-1. Summary of RTE SXL Error Messages

Messages	Page
BP MEM OVF	4-1
CHSM ERR.	4-1
COMMON LNTH ERR	4-1
CREAT ERR	4-2
DUPL EP:	4-2
EVAL. ERR	4-2
FILE READ OR WRITE ERR	4-2
ILL. BP LNTH	4-2
ILL. CHAR	4-2
ILL. EXT	4-2
ILL. LINKS STRT ADDR	4-3
ILL. RCRD TYPE	4-3
MEM OVF	4-3
NAM OUT OF SEQ	4-3
NO CMND ID	4-3
NO LINK ROOM	4-3
NO MAIN PRGM	4-3
NO NAM RCRD	4-4
OPEN ERR	4-4
SXL WAITING FOR DISC TRACKS	4-4
UNRECOGNIZED STMT	4-4
WHAT NAME TO USE FOR ABS. FILE?	4-4

BP MEM OVF BPLOCC = *octal number* LWABP = *octal number*

- Meaning: The present module requires too much Base Page data (ORB). This message is output when the sum of the Base Page Links required and the ORB data size for the current module exceed LWABP.
- Action: Increase LWABP and decrease FWABP, and restart.
- CHSM ERR. BACKUP TO LAST EOT?# FRAMES = *decimal number*
- Meaning: A checksum occurred in the current relocatable file. The "BACKUP TO LAST EOT" part of this message is included for the paper tape user. The number of frames are printed for the user who wishes to back up the tape manually. When this error occurs, it means the whole record (including the first word, from which the number of frames is obtained) may be in error. If the option to backup to the last EOT sensed is taken, SXL skips the number of good records it has read without verifying anything, using a counter which is zeroed at the beginning of each file (BOT).
- Action: The operator manually backs up the tape to either the record just read and answers "NO," or backs up to the beginning of the tape and answers "YES," or enters "BS" to backspace the device (legal for magnetic tape only). The system may abort SXL if this operation is illegal for the device. There is no need to "TR."
- COMMON LNTH ERR, LNTH = *octal number*, NOW = *octal number*
- Meaning: The length declared in the present module is larger than the block size declared in the first COMMON-using module (or via a BOUNDS or SET statement).
- Action: BOUNDS or SET statements may be overridden on-line to declare new block sizes.

RTE SXL

CREAT *file name* ERR *decimal number*

Meaning: The reported FMP error code was returned when an attempt was made to create a file. The error may have the following causes: disc down, duplicate name, file too long or REC size error, cartridge or file not found, or no room, not enough parameters, cartridge locked, directory full, illegal name, illegal type or size 0.

Action: See the FMP manual for corrective action. Re-enter the statement with a unique file name.

DUPL EP: *entry point name*

DEF'D IN *module name*

REDEFINE?

Meaning: A duplicate entry point name appears in an ENT record. The module name which defined it is output.

Action: To ignore the new definition and keep the previous one, type "NO" on keyboard. Otherwise, the previous definition is lost and the new one is substituted. All future references to the symbol use the new value. The previous link(s) allocated for the symbol are not redefined. There is no need to enter "TR" to continue.

EVAL.ERR,EXPR = *octal number* (8)

Meaning: A statement of the form "SET \$ (A + C + . . .) TO . . ." has been entered, and the address expression (i.e., A + C + . . .) evaluates to less than 4.

Action: Define all terms and re-enter.

FILE READ OR WRITE ERR = *decimal number* FILE *file name*

Meaning: This error can result from any of the following: disc down, attempt to read or position to a record not

(Continued)

Meaning: (Cont) written, or write an illegal length on update, attempt to write on a cartridge or file not found, or no room to write, attempt to write with an invalid security code, not enough parameters, DCB not open, SOF or EOF read or sensed, attempt to write with directory full, attempt to read or write on type 0, which does not support the operation.

Action: See the FMP manual for corrective action. Retry.

ILL. BP LNTH

Meaning: A negative number is given in the base page length word of the NAM record (word 8).

Action: It is impossible for SXL to allocate ORB data space if it is unknown in advance how much is needed. This is because the base page area for links and for ORB data is allocated contiguously, starting from BPLOCC (ORB data first). At present, the SIO ALGOL Compiler is the only HP-supported translator which produces such a NAM record, owing to its one-pass nature. Relocation of such a module is impossible.

ILL CHAR *character*

Meaning: A character which is not a letter, digit, blank, or legal special character is entered. This error also occurs when a non-octal digit is entered in an octal number (decimal numbers are suffixed with a "D").

Action: Re-enter the line correctly.

ILL. EXT

Meaning: A DBL record contains a reference to an external not previously entered with an EXT record. This error may also indicate a READ error, including skipping (see CHSM ERR.).

Action: Relocate the module (rather than attempt to locate the correct EXT record), and rerun RTE SXL.

ILL. LINKS STRT ADDR. = *octal number* BPLOCC = *octal number*

Meaning: A statement such as "LINKS START AT 1500, .IOC., .IOC. + 3,..." was entered when the link start address was less than the current value of BPLOCC. Constants in the statement do not cause the error.

Action: Re-enter with a legal start address, or reset the value of BPLOCC and re-enter.

ILL. RCRD TYPE

Meaning: The record type is not a NAM, ENT, EXT, DBL, or END as defined in the relocatable code.

Action: Retry; reassemble or recompile module, and start over.

MEM OVF LOCC = *octal number* LWAM = *octal number*,

Meaning: Present module is too large.

Action: Increase value for LWAM, and rerun RTE SXL.

NAM OUT OF SEQ

Meaning: A second NAM record was encountered before an END.

Action: Correct tape and rerun RTE SXL.

NO CMND ID

Meaning: A command identifier character (-) is required in column 1 whenever the current command file is not the keyboard, except in the case of comments which must contain an asterisk (*) in column 1. When input is from the keyboard, the identifier is optional.

Action: Re-enter the line correctly on teletype and retry the execution.

NO LINK ROOM

Meaning: SXL is unable to allocate a link in the current linking mode.

Action: Any or all of the following corrective actions should be taken:

1. Change to current page linking mode.

2. The "SET LOCC TO NEXT PAGE" command should be invoked (as often as necessary) to set the location counter to the next page boundary to avoid a page crossing.

3. Rearrange modules so that small modules fall on page crossings instead of large ones.

4. Relocate the modules with a larger "guess" area (200 should do).

NO MAIN PRGM

Meaning: An END or SEGMENT statement is entered, but no main program has yet been relocated.

Action: Continue relocation of the main program module (and other subroutines, if required). If the primary entry point (address of the first instruction to be executed when control is passed to the program unit) is known, set the keyword "?XFER" to that address using the SET statement, and re-enter either END or SEGMENT.

NOTE

At least one module must be relocated between turning on SXL and entering the first END or SEGMENT statement. Also, at least one module must be relocated between SEGMENT statements, and between the last SEGMENT and END statements.

RTE SXL

NO NAM RCRD

Meaning: The first record in a module is not a NAM record. The checksum is verified before record and record-legality tests are made. This error indicates a problem with the attempt to read relocatable code and includes mistaking ASCII as relocatable binary.

Action: Restart with correct tape (or file).

OPEN *file name* ERR *decimal number*

Meaning: The reported error code was returned when an attempt to OPEN the file was made. The error may have the following causes: disc down, cartridge or file not found, no room, invalid security code, file currently open to eight programs, file open exclusively, or LOCK rejected, attempt to OPEN a type 0 as a type 1, not enough parameters, cartridge locked.

Action: See the EMP manual for corrective action and re-enter the statement.

SXL WAITING FOR DISC TRACKS

Meaning: SXL was unable to get any disc tracks.

Action: Terminate, and arrange for more disc space. Then start again.

UNRECOGNIZED STMT

Meaning: A parsing error was detected in this statement.

Action: Correct the syntax and re-enter the statement.

WHAT NAME TO USE FOR ABS. FILE?

Meaning: When the output file format is core-image, SXL attempts to create a scratch file for the absolute binary output. SXL attempts to call this file “!!!!!!” initially, and attempts to find a combination of the first two characters that will work, until it gets to “\$\$!!!!”, at which point this error message is output.

Action: Enter a unique file name. There is no need to enter “TR” to continue.

SECTION V

NON-RTE TARGET SYSTEMS

The Real-Time Executive System Cross Loader (RTE SXL) can be used in an RTE environment to generate programs for use not only in another RTE system, but also for use in either a Basic Control System (BCS) or a Terminal Communication Executive (TCE/3) target system. A software interface is provided to ease the generation of either of these non-RTE oriented programs. For a BCS application, the interface is the Basic Control System Generator (BCSGN), HP Part Number 29103-60119. For a TCE/3 application (as used with the HP 91701A/B Software Package), the interface is the Terminal Communication Executive Generator (TC3GN), HP Part Number 29100-60032. The host system is the Real-Time Executive (RTE) with the File Manager (FMP) option.

The user is not required to have detailed knowledge of RTE SXL to utilize these generators. Both BCSGN and TC3GN provide an operator interface (like the Prepare Control System (PCS)) so that users who are familiar with BCS generation procedures can perform these activities on an RTE without extensive training in the use of RTE SXL. Thus, an RTE may be used as a BCS or TCE/3 program preparation facility.

THEORY OF OPERATION

Both BCSGN and TC3GN function similarly in that they operate interactively and proceed sequentially through a series of requests. An input is expected after each request and the program will not proceed until that input is received. Thus, the operator will not overlook any required parameters needed for subsequent generation by RTE SXL. See Figure 5-1.

The host RTE must have the File Manager (option Y01) installed. RTE SXL must be available to process the output of BCSGN or TC3GN. All necessary relocatable modules and libraries must exist in disc files. The transfer file must be able to be created and opened. None of these files may have a read-restrictive security code.

Several limitations are imposed by the RTE, the RTE SXL, and the BCSGN or TC3GN schedule:

- RTE-SXL cannot accept a security code with a file name. Therefore, all relocatable modules, libraries, transfer files, etc. must not have a security code which will restrict read access.

- BCSGN or TC3GN output is to a disc file only. Prior to subsequent executions of BCSGN or TC3GN, this file must be purged so that BCSGN's or TC3GN's creation attempt does not result in an FMP error return.
- BCS, BCS with applications, TCE/3, and TCE/3 with applications are the only supported system configurations. Addition of the BCS relocating loader to a system is not supported.
- No checking is possible by BCSGN or TC3GN to determine that the operator has input the correct number of interrupt links for the required drivers, that there are no undefined externals, nor that a "Main" has been included in the set of modules loaded. These items are discovered by RTE SXL during its processing and result in error messages. (See Section IV, Error Messages, for the meaning of each message and the recommended operator action.)
- The snap contains all module entry points and BCSGN or TC3GN makes no attempt to determine whether any entries should be undefined.

SOFTWARE REQUIREMENTS

BCSGN and TC3GN are designed to execute in the background area of an RTE with option Y01 and RTE SXL. No unique or unusual support software is required.

The generator (either BCSGN or TC3GN) does not require greater than 6080 words of background disc resident memory when it is configured to run in that area. The program will require one ID segment.

HARDWARE REQUIREMENTS

Since the generator (BCSGN or TC3GN) is intended to execute in conjunction with the operating system (RTE, RTE File Manager, and RTE SXL), the operating system dictates the host environment. BCSGN or TC3GN themselves require no additional hardware.

The nominal RTE systems which contain at least 6080 words of background disc resident area are required for this program.

RTE SXL

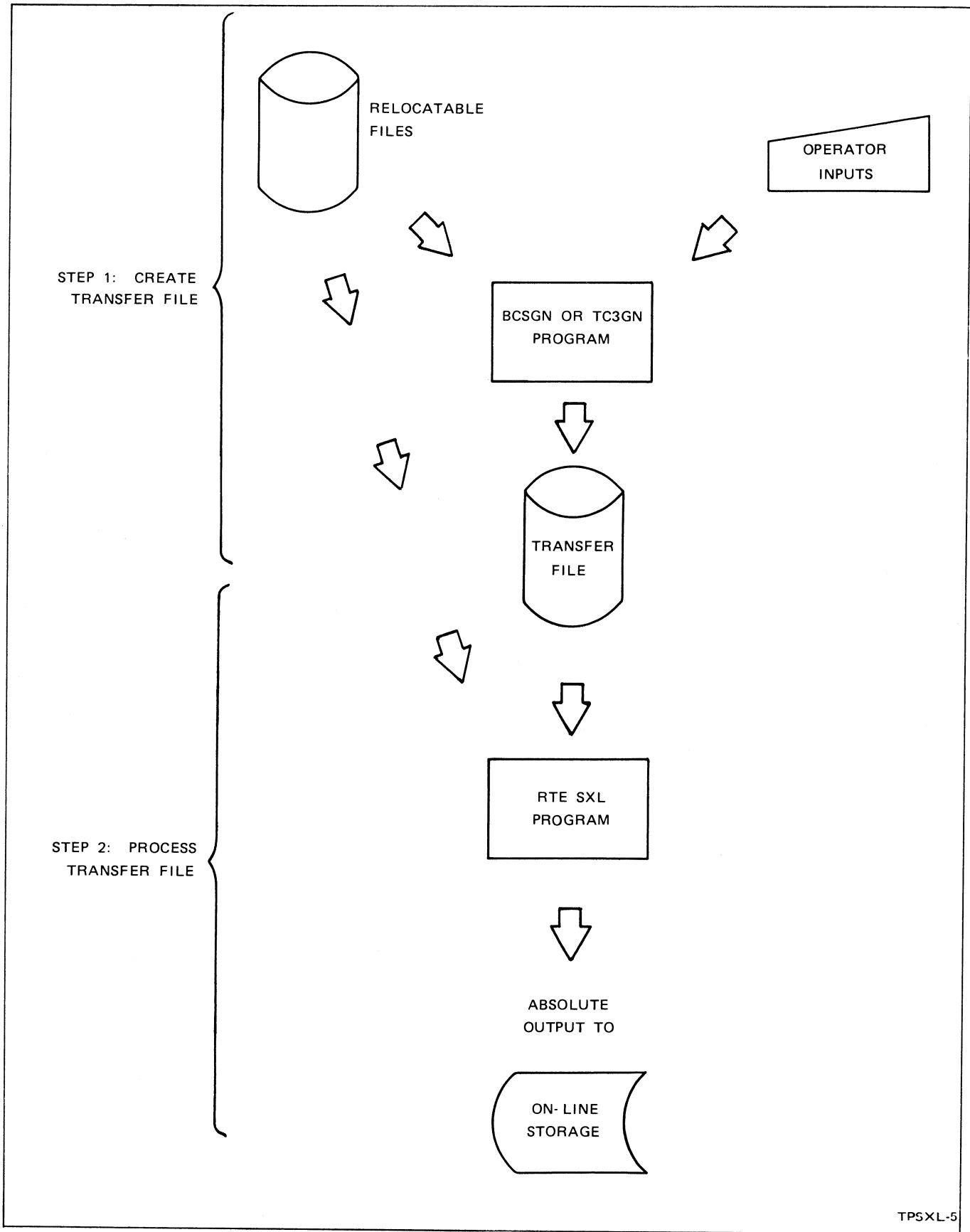


Figure 5-1. BCS or TCE/3 Generation

OPERATING INSTRUCTIONS**PROCEDURE FOR TURNING ON BCSGN**

The Basic Control System Generator (BCSGN) is turned on as follows:

ON, BCSGN**Purpose:**

To turn on the special interface for generating a program to run in a BCS target system.

Format:

***ON, BCSGN [, p1 , p2 , p3 , p4]**

Where:

*	=	The RTE system prompt character.
p1	=	Input logical unit number. (Default is the system console.)
p2	=	List and error logical unit number. (Default is the system console.)
p3	=	Error input logical unit number. (Default is the system console.)
p4	=	Error output logical unit number. (Default is the system console.)

ON, TC3GN (Continued)**Format:**

***ON, TC3GN [, p1 , p2 , p3 , p4]**

Where:

*	=	The RTE system prompt character.
p1	=	Input logical unit number. (Default is the system console.)
p2	=	List and error logical unit number. (Default is the system console.)
p3	=	Error input logical unit number. (Default is the system console.)
p4	=	Error output logical unit number. (Default is the system console.)

COMMENTS

When TC3GN executes, the message "TC3GN..." is displayed on the system console, followed by "CR/LF" and the first input request.

PROCEDURE TO ABORT BCSGN OR TC3GN

BCSGN or TC3GN normally tie up the system console. Therefore, a method is provided to abort the BCSGN or TC3GN at any time. Typing an exclamation point (!) as the first character in response to any input request will abort the program, closing all files currently created and closing any files that are currently open.

NOTE

If an existing file name begins with a "!", input the name starting in the second column, leaving a space in the first column.

OPERATOR DIALOGUE

BCSGN or TC3GN generate their respective programs through a step-by-step dialogue with the operator. Each of the following requests must be answered by the operator with the appropriate response as outlined.

COMMENTS

When BCSGN executes, the message "BCSGN..." is displayed on the system console, followed by "CR/LF" and the first input request.

PROCEDURE FOR TURNING ON TC3GN

The Terminal Communication Executive Generator (TC3GN) is turned on as follows:

ON, TC3GN**Purpose:**

To turn on the special interface for generating a program to run in a TCE/3 target system.

(Continued)

RTE SXL

TRFILE:	Name of file which will contain the commands to direct RTE SXL in generating a BCS or TCE/3 system (this file is the output of this program).	Error Message:	ERROR*FMGR-XX Where XX is a File Manager error (normally-2).
Expected Response:	Any legal disc file name, which has not already been defined.	ECHO COMMAND?	This question is used to determine if the commands that direct RTE SXL to generate a BCS or TCE/3 system are to be "echoed" onto the list device when RTE SXL begins execution.
Error Message:	ERROR*FMGR-XX Where XX is a File Manager error (normally-2)	Expected Response:	YES if commands are to be echoed, or NO if commands are not to be echoed. (The normal response is "NO".)
OUTFILE:	Name of output file where the absolute code generated by RTE SXL is to be put.	Error Message:	ERROR*ILLEGAL INPUT PARAMETER
Expected Response:	<i>name</i> , SC, CR	FWABP?	INPUT first word of available base page (this would normally be 110 (octal)).
Where:		Expected Response:	A number between 20 (octal) and 1776 (octal), inclusive.
<i>name</i> =	Any legal file name.	Error Message:	ERROR*ILLEGAL INPUT PARAMETER
SC =	A positive numeric security code or a two-character A S C I I security code. (Optional.)	LWAM:	Input last word of available memory (this would normally be x7677, where x ranged from 0 to 7)
CR =	A positive number referring to the cartridge where the output file is to be placed. (Optional.)	Expected Response:	A number greater than entry for LWABP and less than 77677 (octal).
Error Message:	ERROR*FMGR-XX Where XX is a File Manager error (normally-2)	Error Message:	ERROR*ILLEGAL INPUT PARAMETER
SNPFILE:	Name of file which will contain the commands required to direct RTE SXL to relocate a module once the initial BCS or TCE/3 system has been generated (this file is the output of the RTE SXL program).	FWAM?	Input first word of memory available for relocating. (normally 2000 (octal))
NOTE		Expected Response:	A number greater than 1777 (octal) than entry for LWAM.
This is not a required file.			
Expected Response:	Any legal disc file name which has not already been defined, or \$NUL which means to not generate a snap.	Error Message:	ERROR*ILLEGAL INPUT PARAMETER

#OF WORDS OF COMMON	If the operator wishes to state the maximum amount that the system allows, input desired number of words of Common. Otherwise, Common will be allocated as indicated by the first program that requires it.	Error Message:	ERROR*FMGR-XX Where XX is a File Manager error (normally-6)
Expected Response:	Any octal number which is less than the amount of core allotted between the FWAM and LWAM, or zero (0) indicating that common should be allotted by the program requiring it. (Normally the response would be 0).	*SEARCH	Identify point in BCSGN or TC3GN where all library file names to be searched are entered.
Error Message:	E R R O R * I N P U T PARAMETER ERROR	?	Prompt requesting file name of library modules to be searched. This would normally be the BCS library.
CURRENT PAGE LINKING?	This input is used to determine if current page linking of all modules is required.	NOTE	If any routines in the BCS library require a second pass, the BCS library file must be stated twice.
Expected Response:	YES if current page linking is required, or NO if default to base page linking is preferred.	Expected Response:	Any legal disc file name which has been defined and currently resides on the disc, or a "/E" which indicates that there are no more library modules to be searched.
Error Message:	ERROR*ILLEGAL INPUT PARAMETER.	Error Message:	ERROR*FMG-XX Where XX is a File Manager error (normally-6).
*LOAD	Identify point in BCSGN or TC3GN at which time all relocatable module file names are to be entered.	*EQT	Identify the point in BCSGN or TC3GN where the equipment table entries are to be defined. The order in which the EQT statements are submitted defines the position of the entry in the equipment table. It also establishes the unit-reference number that the programmer uses in writing input/output request to IOC.. The first statement entered describes the unit which is to be referenced as number 7 (octal); the second statement, number 10 (octal); the third statement, number 11 (octal); etc. Numbers 1-6 are reserved for Standard unit definition in the Standard Equipment Table.
?	Prompt requesting a file name of a relocatable module to be input. This includes IOC. and all associated drivers. Only one file name per ? prompt.	?	Prompt requesting EQT entries to be input.
Expected Response:	Any legal disc file name which has been defined and currently resides on the disc, or a "/E" which indicates that there are no more modules to be relocated.		

RTE SXL

Expected Response:	nn,D.ee (,D) (,Uu)	Error Message:	E R R O R * I N P U T PARAMETER ERROR
Where:		TTY?	Input unit-reference number for teleprinter output.
nn	The channel number (select code) for the device. For a device connected to two or more channels, nn is the lower numbered channel.	Expected Response:	Any octal number ranging from 7 (octal) to maximum EQT entry assigned.
D.ee	The BCS symbolic name for the related equipment driver subroutine ee is the equipment code.	Error Message:	E R R O R * I N P U T PARAMETER ERROR
D	A direct memory access channel is required to operate the device.	LIB?	Input unit-reference number for messages from the relocatable library.
Uu	The physical unit number u (0-7) for addressing the device if it is attached to a multiunit controller.	Expected Response:	Any octal number ranging from 7 (octal) to maximum EQT entry assigned.
or: /E		Error Message:	E R R O R * I N P U T PARAMETER ERROR.
Where:		PUNCH?	Input unit-reference number for tape punch.
	This alternate response indicates that there are no more EQT entries to be entered.	Expected Response:	Any octal number ranging from 7 (octal) to maximum EQT entry assigned.
Error Message:	ERROR*ILLEGAL INPUT PARAMETER	Error Message:	E R R O R * I N P U T PARAMETER ERROR
*SQT	Identify the point in the BCSGN or TC3GN where the standard equipment table entries are to be defined. In constructing the Standard Equipment Table, BCSGN types a mnemonic for the standard unit and waits for the reply. The reply consists of the unit-reference number for a device previously described in the equipment table.	INPUT?	Input unit-reference number for paper tape reader.
KYBD?	Input unit-reference number which will be assigned to the keyboard input.	Expected Response:	Any octal number ranging from 7 (octal) to maximum EQT entry assigned.
Expected Response:	Any octal number ranging from 7 (octal) to maximum EQT entry assigned.	Error Message:	E R R O R * I N P U T PARAMETER ERROR
		LIST?	Input unit-reference number for list output device.
		Expected Response:	Any octal number ranging from 7 (octal) to maximum EQT entry assigned.
		Error Message:	E R R O R * I N P U T PARAMETER ERROR

DMA?

After the equipment tables are completed, the generator (BCSGN or TC3GN) requests information about the availability of DMA channels to be controlled by the Input/Output Control (IOC) and equipment driver subroutines. BCSGN or TC3GN types the message "DMA?" and the operator responds with the available DMA channel numbers.

Expected Response: C₁ (C₂)

Where:

C₁ is 6 if one channel is available.

C₂ is 7 if the second channel is available.

If no DMA channel is available, the C₁ reply should be 0 (zero).

Error Message:

E R R O R * I N P U T
PARAMETER ERROR

*INT LNK

Prompt Requesting Interrupt Link. The generator (BCSGN or TC3GN) requires certain parameters to set the interrupt linkage for input/output processing. The information required for each device includes:

- 1) The interrupt location within the reserved area in low core.
- 2) The entry point name of the interrupt processing section in the equipment driver subroutine for the device.
- 3) The address of the word in base page which is to contain the 15-bit absolute address of this entry point name.

Expected Response: a₁,a₂,I.ee

Where:

a₁ is the address in low core of the interrupt location for the device (channel).

a₂ is the address in the base page of the word to contain the absolute address of the interrupt processor entry point.

I.ee is the entry point name of the interrupt processor section or the equipment driver subroutine. ee is the equipment type code used by BCS.

There are some devices which use more than one I/O channel (such as magnetic tape and disc). In devices of this type, the lower numbered channel transfers data; the higher numbered channel transfers commands.

For these devices, and these devices only, each I.ee entry must have a corresponding entry of the form:

b₁, b₂, C.ee

Where:

b₁ is the corresponding a₁ + 1

b₂ is the base page address of the word containing the absolute address of the continuation entry point (typically a₂ + 1).

ee is the same as the corresponding I.ee entry.

The response to the "*INT LNK" message may have the following form if a constant (for example, a halt) is to be set in the interrupt location:

a, c
Where:
a = The address in low core
of the interrupt
location for the device
(channel).
c = The constant in octal
form that is to be
stored at location *a*.

NOTE

Examples of this format appear in
Figure 5-2.

or: /E

Where:

This response indicates that
there are no more interrupt
linkage entries.

Error Message:

**E R R O R * I N P U T
PARAMETER ERROR**

OK?

After input from each section
has been completed, and the
input device is the keyboard,
it is possible, if an error has
been made, to reenter that
section. The "OK?" question
will not occur if the input
device is not the keyboard
input device, nor is a response
required for this question if
the input device is not the
keyboard input device.

The section breaks occur at
the following places:

- After the response to
“CURRENT PAGE
LINKING?”
- After the last response to
“*LOAD”
- After the last response to
“*SEARCH”
- After the last response to
“*EQT”
- After the last response to
“*SQT”
- After the last response to
“*INT LINK”

Expected Response: YES if everything to this
point is all right (OK),
or

NO if an error has been
made and the operator
wants to re-enter the
previous section.

Error Message:

**ERROR*ILLEGAL INPUT
PARAMETER**

ERRORS AND DIAGNOSTICS

All error messages are printed on the error device specified
when the program was turned on. The prompt will be
reprinted and the input parameter will be reaccepted. It is
not possible to continue to the next question if a mistake
has been made. Actual error messages for any one input
appear under the heading "Error Message" accompanying
each request listed in the foregoing Operator Dialogue
section. All error messages start with "ERROR*".

The diagnostics methods are similar in nature to those
found in PCS, which are text editing and checking for
illogical or out-of-range entries in parameters.

Whenever FMGR-XX is appended to ERROR* the XX
error is an error code returned to BCSGN or TC3GN from
the RTE File Manager and the error code meanings are
described in the RTE File Manager manual.

INTERFACE DETAILS**INPUT DATA FORMAT**

All data will be input from the input device (which is
determined at execution time . . . defaulted to CRT).

All data will be treated by a common input subroutine
which will divide the input string into a number of input
fields using a "," as the field delimiter.

All numeric data *defaults to octal* however, provision is
made to handle decimal numbers by appending the letter
"D" as a suffix.

All numeric data is assumed to be positive integer and an
error will occur if negative or non-integer numbers are input.

All ASCII input will be packed two bytes per word, and
will be addressed as bytes rather than words.

No input line can contain more than five fields.

All input will resemble PCS input as closely as possible.

OUTPUT DATA FORMAT

All error messages will be printed on the error device.

All commands generated are output to the disc via the File Manager in a type 3 file which has no security code. The output structure is of a type which will be acceptable as input to RTE SXL as a transfer file.

PROMPT MESSAGES

All messages typed to the operator which require an operator response are so designed that if the message is not useful or more input is required, it can be added with little effort and minor program changes.

EXAMPLES

An example of batch BCS generation is shown in Figure 5-3, and an example of interactive BCS generation is shown in Figure 5-4.

Message	Reply	Meaning
*INT LNK	27,102027	A halt executed when interrupt occurs on channel 27.
*INT LNK	26,0	A NOP is executed when interrupt occurs on channel 26; the program resumes normal execution.

Figure 5-2. Examples of Setting Constants in the Interrupt Location

RTE SXL

BCSGN	7
TRFILE? ←	TTY? ←
TEST	7
QUITFILE? ←	LIB? ←
ABSFIL,LA,1	10
SNPFILE? ←	PUNCH? ←
SNPOUT	11
ECHO COMMANDS? ←	INPUT? ←
YES	10
FWARP? ←	LIST? ←
111	12
LWAM? ←	DMA? ←
77677	6,7
FWAM? ←	*INT LNK
2000	? ←
# WORDS OF COMMON? ←	15,25,1.00
0	? ←
CURRENT PAGE LINKING? ←	16,26,1.01
YES	? ←
*LOAD	17,27,1.02
? ←	? ←
INCF	20,30,1.12
? ←	? ←
D.00F	/E
? ←	END BCSGN
D.01F	
? ←	
D.02F	
? ←	
D.12BF	
? ←	
TEST1R	
? ←	
/E	
*SEARCH	
? ←	
FAMTR	
? ←	
BCSLIB	
? ←	
/E	
*EQT	
? ←	
15,D.00	
? ←	
16,D.01	
? ←	
17,D.02	
? ←	
20,D.12	
? ←	
/E	
*SQT	
KYBD? ←	

Figure 5-3. Batch BCS Generation

```
*ON,BCSGN
BCSGN
TRFILE? TEST
OUTFILE? ABSFIL.LA,1
SNPFILE? SNPOUT
ECHO COMMANDS? YES
FWABP? 111
LWAM? 77677
FWAM? 2000
# WORDS OF COMMON? 0
CURRENT PAGE LINKING? YES
OK? YES
*LOAD
? IOCF
? D.00F
? D.01F
? D.02F
? MOVER
? /E
OK? YES
*SEARCH
? BCSLIB
? /E
OK? YES
*EQT
? 15,D.00
? 16,D.01
? 17,D.02
? /E
OK? YES
*SQT
KYBD? 7
TTY? 7
LIB? 10
PUNCH? 11
INPUT? 7
LIST? 7
OK? YES
DMA? 6,7
OK? YES
*INT LNK
? 15,25,I.00
? 16,26,I.01
? 17,27,I.02
? /E
OK? YES
END BCSGN
```

Figure 5-4. Interactive BCS Generation

SECTION VI

RELOCATING PROGRAMS FOR RTE-C (HP 9600C)

The Real-Time Executive Core-Based Software System (RTE-C) is a general purpose non-disc operating system for HP-2100A computer series. The RTE-C Executive provides some functions similar to the disc-based Real-Time Executive Software System (RTE). These include:

- Scheduling of program execution
- Interrupt processing
- Input/output processing

RTE-C is Hewlett-Packard's core-based multiprogramming system.

RTE SXL can relocate programs for RTE-C. This again allows a user to use the background of RTE to develop and relocate programs which later are to be executed on the RTE-C. The following description and diagram discuss this procedure.

SNAPSHOT PHASE

During the configuration of RTE-C, a snapshot phase is entered. The RTSGN Snapshot is on ASCII tape which is compatible with both the RTE-C Relocating Loader and the RTE SXL. This tape may be listed off-line. The Snapshot is used by either loader to relocate user programs that will be loaded by APLDR under control of RTE-C on-line. The process of relocating user programs off-line is to first the Snapshot by the loader followed by the relocatable user program(s) and subroutines. For more information on the RTE-C Relocating Loader refer to the loader section of the RTE-C manual.

RTE-C SNAPSHOT FORMAT

The RTE-C Snapshot contains information which directs the RTE-C Relocating Loader or RTE SXL in setting up the proper memory bounds and symbol table for program preparation.

CORE BOUNDS

RTSGN outputs the bounds keyword values (FWABP, LWABP, FWAM, LWAM, FWAC, and LWAC) which define the bounds of core which may be used for user programs. The six keywords specify the available memory in base page, main memory, and COMMON.

RESIDENT ENTRY POINTS

The RTE-C Snapshot also contains information for some of the executive entry points and all of the entry points of routines in the Resident Library area.

This snapshot file is accessed by RTE SXL either via an input device or from the Real-Time Executive File Manager. If it is accessed from the File Manager, it must have been previously created as an ASCII command file.

Once the user has accessed this snapshot file which defines the configuration of the corresponding RTE-C, the loading procedure continues as previously described. The absolute output file must ultimately be on paper tape for loading into RTE-C using the APLDR.

SECTION VII

PRECAUTIONS

The Real-Time Executive System Cross Loader (RTE SXL) is a generalized relocating loader, which allows the user to describe the operating system of the target machine. Because of this capability, some errors cannot be detected by RTE SXL. Below is a partial list of possible inconsistencies which should be avoided when preparing a program for use in RTE:

- Make sure the address expansion in any SET \$ (*address expression*) TO... is contained within the relocated program. DO NOT use SET \$() to place patches into the RTE system.
 - DO NOT use values for BOUNDS keywords which are inconsistent with the program type. Specifically, if the program is Type 2, the core bounds for the relocated program must fit within the foreground disc-resident area, including base page. Use of the OUTPUT statement is required if program is to be in the foreground (real-time) disc-resident area.
 - DO NOT enter an OUTPUT statement in which the program type parameter is inconsistent with the core bounds being used for it.
 - DO NOT re-enter an OUTPUT CI statement to correct an error from an OUTPUT FGR statement. Always re-enter OUTPUT FGR.
 - Be sure that LOCC is not on the same page as any core-resident system, privileged or reentrant entry point, including EXEC, \$LIBR, and \$LIBX, before using the DIRECT linking option. Interrupts are not held off by the hardware if they occur while a JSB instruction is being executed, if it is not an indirect instruction. As soon as a sufficient number of modules have been relocated to set LOCC on a higher page, the DIRECT option can be used.
 - Relocation of assembly language code in current page linking mode can be hazardous. In current page linking mode, any program (such as the HP 2313A RTE Driver, DVR62) which dynamically modifies (or moves) its own memory reference instruction may find the established link unavailable to the instruction if it has been moved to a different page. When the link is on the base page, it is accessible on any page, and no problem results. However, in current page linking mode, the necessary link stored on one page could occasionally be stored on a different page when the instruction is moved by the driver.
 - User file names must consist of no more than six characters. The allowable characters include all 26 letters of the alphabet (A through Z), all 10 numerical digits (0 through 9), and the following special characters:
 ? @ " # % . /
- (Note that the ampersand (&) and dollar sign (\$) are not among the allowable characters.) In addition to the above limitations, a name must not have a numerical digit as its first character.
- Where symbolic entry addresses and interrupt linkage addresses are required, be sure to include all relocatable driver modules. If driver entry point symbols are not encountered during relocation, RTE SXL will insert zeroes in the undefined locations. For example, if the statement

-SET \$(25) TO I.00

is issued and the symbolic entry address I.00 is not defined, RTE SXL will put a zero into location 25.

INDEX

A

Abbreviations, Keyword	2-2
ABORT	2-3
Aborting BCSGN	5-3
Aborting TC3GN	5-3

B

Base Page	2-5, 2-8
Base Page Linking Mode	2-5
Basic Control System Generator	5-1
Batch BCS Generation (Figure 5-3)	5-10
BCS or TCE/3 Generation (Figure 5-1)	5-2
BCSGN	5-1
BOUNDS	2-3
BPLOCC	2-2, 2-16
BP MEM OVF	4-1

C

CHSM ERR.	4-1
Command Input, Order of	3-4
Command Language	2-3
ABORT	2-3
BOUNDS	2-3
DISPLAY	2-4
ECHO	2-5
END	2-5
LINKS IN	2-5
LINKS START AT	2-6
MAP	2-8
OUTPUT	2-9
RELOCATE	2-12
SEARCH	2-13
SEGMENT	2-14
SET	2-15
SNAP	2-17
TRANSFER	2-18
Commands	2-1
COMMON LNTH ERR	4-1
Configuration, Target Machine	3-1, 5-1
Continuation Symbol	2-2
Core Bounds	2-3, 6-1
Core Map	2-8, 2-10, 2-11
Core Map without Current Page Linking (Figure 2-3)	2-11
CREAT ERR	4-2

CR/LF	5-3
Current-Page Linking Mode	2-5
Current Page Mode Link Areas (Figure 2-1)	2-7

D

DEBUG	1-1
Dialogue, Operator	5-3
DISPLAY	2-4
Distributed System Use	3-4
DUPL EP:	4-2
Duplicate Symbol Definition	1-1

E

ECHO	2-5
END	2-5
Entry Points, Resident	6-1
Error Messages	4-1, 5-8, 5-9
BP MEM OVF	4-1
CHSM ERR.	4-1
COMMON LNTH ERR	4-1
CREAT ERR	4-2
DUPL EP:	4-2
EVAL. ERR	4-2
FILE READ OR WRITE ERR	4-2
ILL. BP LNTH	4-2
ILL. CHAR	4-2
ILL. EXT	4-2
ILL. LINKS STRT ADDR.	4-3
ILL. RCRD TYPE	4-3
MEM OVF	4-3
NAM OUT OF SEQ	4-3
NO CMND ID	4-3
NO LINK ROOM	4-3
NO MAIN PRGM	4-3
NO NAM RCRD	4-4
OPEN ERR	4-4
SXL WAITING FOR DISC TRACKS	4-4
UNRECOGNIZED STMT	4-4
WHAT NAME TO USE FOR ABS. FILE?	4-4
Errors and Diagnostics	5-8
EVAL. ERR	4-2
Examples of Setting Constants in the Interrupt Location (Figure 5-2)	5-9
EXEC	2-9, 7-1
Execution from Dormant Status	3-1

F

FILE READ OR WRITE ERR	4-2
FIRST ADDRS	2-8
Format, Input Data	5-8
Format, Output Data	5-9
Format, RTE-C Snapshot	6-1
Formats, Input	2-2, 5-8
FWABP	2-2, 2-4
FWAC	2-2, 2-4
FWAM	2-2, 2-4

G

General Description	1-1
Global Variables Information	2-9
GLOBAL VARS	2-8
GLOBALS	2-8
Glossary	vii
Guess Area	2-6

H

Hardware Requirements	1-1, 5-1
---------------------------------	----------

I

ILL. BP LNTH	4-2
ILL CHAR	4-2
ILL. EXT	4-2
ILL. LINKS STRT ADDR	4-3
ILL. RCRD TYPE	4-3
Immediate Action Statements	3-4
Indirect Operator	2-2, 2-15
Information Contained in the Core Map	2-8
Input Data Format	5-8
Input Formats	2-2
Input, Order of Command	3-4
Installation	1-1
Interactive BCS Generation (Figure 5-4)	5-11
Interface Details	5-8

K

Keyword Abbreviations	2-2
BPLOCC	2-2
FWABP	2-2
FWAC	2-2
FWAM	2-2
L.G	2-3
LIBRY	2-3
LOCC	2-3
LWABP	2-3
LWAC	2-3
LWAM	2-3

Keyword Abbreviations (Continued)

NEXTPAGE	2-3
?XFER	2-3
Keywords	2-2

L

LAST ADDRS	2-8
L.G	2-3, 2-5
LIBRY	2-3, 2-5, 2-9
Link Table	2-9
LINKS IN	2-5
LINKS START AT	2-6
Load-and-Go	2-3, 2-5
LOCC	2-3, 2-16
LWABP	2-3, 2-4
LWAC	2-3, 2-4
LWAM	2-3, 2-4

M

MAP	2-8
Map, Core	2-8, 2-10, 2-11
MEM OVF	4-3
Messages, Error	4-1, 5-8, 5-9
Messages, Prompt	5-9
Module Core Bounds Information	2-8
Module 1 Declares Common (Figure 3-1)	3-3
Module 2 Declares Common (Figure 3-2)	3-3

N

NAM OUT OF SEQ	4-3
NEXTPAGE	2-3, 2-16
NO CMND ID	4-3
NO LINK ROOM	4-3
NO MAIN PRGM	4-3
NO NAM RCRD	4-4
Non-RTE Target Systems	5-1

O

ON, BCSGN	5-3
ON, TC3GN	5-3
ON, SXL	3-1
OPEN ERR	4-4
Operating Procedure	3-1
Operator Commands	2-1
Operator Dialogue	5-3
Order of Command Input	3-4
OUTPUT	2-9
Output Data Format	5-9

P

Phase I. Target Machine Configuration	3-1
Phase II. Relocation Commands	3-2
Precautions	7-1
Primary Linking Area	2-6
Procedure for Turning On RTE SXL	3-1
Procedure for Turning On BCSGN	5-3
Procedure for Turning On TC3GN	5-3
Procedure, Operating	3-1
Procedure to Abort BCSGN or TC3GN	5-3
Program Generator for BCS	5-1
Program Generator for TCE/3	5-1
Program Scheduling	3-1
Prompt Messages	5-9
Punctuation Symbols, Syntax	2-2

R

REF.BY	2-8
REL	2-12
RELOCATE	2-12
Relocating Programs for RTE-C (HP 9600C)	6-1
Relocation Commands	3-2
Requirements, Hardware	1-1, 5-1
Requirements, Software	1-1, 5-1
Reserved Words	2-3, 2-5
Resident Entry Points	6-1
RTE SXL Commands (Table 2-1)	2-1
RTE SXL SCHEDULE	3-1
RTE System Cross Loader Operation (Figure 1-1) . .	1-2
RTE-C	6-1
RTE-C Snapshot Format	6-1

S

Scheduling RTE SXL	3-1
SEARCH	2-13
Secondary Linking Area	2-6
SEGMENT	2-14
SET	2-15
SNAP	2-17

Snapshot	viii, 2-14, 2-17, 6-1
Snapshot Phase	6-1
Software Requirements	1-1, 5-1
Summary of RTE SXL Error Messages (Table 4-1) . .	4-1
SXL WAITING FOR DISC TRACKS	4-4
Symbols, Syntax Punctuation	2-2
Syntax Punctuation Symbols	2-2

T

Target Machine Configuration	3-1, 5-1
Target System, RTE-C	6-1
Target Systems, Non-RTE	5-1
TC3GN	5-1
Terminal Communication Executive Generator	5-1
Theory of Operation, BCSGN and TC3GN	5-1
TR	2-18
TRANSFER	2-18
Turning On BCSGN	5-3
Turning On RTE SXL	3-1
Turning On TC3GN	5-3
Typical Core Map (Figure 2-2)	2-10

U

UNRECOGNIZED STMT	4-4
Use of DEBUG	1-1

W

WHAT NAME TO USE FOR ABS. FILE?	4-4
Words, Reserved	2-3, 2-5

\$	2-2, 2-15, 7-1
\$END SXL	2-9
\$LIBR	2-9, 7-1
\$LIBX	2-9, 7-1
?	5-5, 7-1
?XFER	2-3, 2-16
*	2-2, 5-3
!	5-3
&	2-2, 7-1

