SAMSO-TR-67-3

COMPILER REQUIREMENTS FOR SPACE PROGRAMMING LANGUAGE (SPL)

Prepared By:

L. J. Carey
A. E. Kroger

System Development Corporation
Santa Monica, California 90406

August 1967

Prepared For:

SPACE AND MISSILE SYSTEMS ORGANIZATION
AIR FORCE SYSTEMS COMMAND
AIR FORCE UNIT POST OFFICE
LOS ANGELES, CALIFORNIA 90045

SAMSO-TR-67-3

COMPILER REQUIREMENTS FOR SPACE PROGRAMMING LANGUAGE (SPL)

Prepared By:

L. J. Carey
A. E. Kroger

System Development Corporation
Santa Monica, California 90406

August 1967

Prepared For:

SPACE AND MISSILE SYSTEMS ORGANIZATION
AIR FORCE SYSTEMS COMMAND
AIR FORCE UNIT POST OFFICE
LOS ANGELES, CALIFORNIA 90045

FOREWORD

This is a technical report defining the Space Programming Language (SPL) compiler requirements. It was produced by the System Development Corporation during the contract period from February 1967 through August 1967. This work was performed under Contract Number FO 4695-67-C-0096. Also produced under this contract was a Specification of SPL, SAMSO TR-67-23.

Based on a study of spaceborne software, SDC previously produced a four volume report, SSD TR-67-11. Volume III of that series contains an initial description of SPL in English prose form.

This report is intended to provide a description of the basic SPL compiler characteristics, capacities, inputs, and outputs.

The personnel involved were:

a. Air Force Project Officers

Roger B. Engelbach
Lieutenant, USAF

Michael A. Ikezawa
Major, USAF

b. Air Force Project Consultant

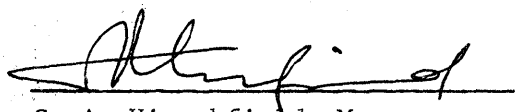
Dr. Walter A. Sturm
Aerospace Corporation

c. SDC Project Manager

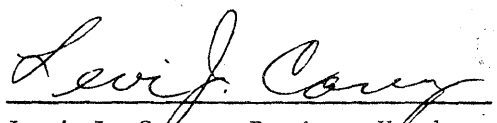
Gerard A. Hirschfield

d. SDC Technical Staff

L. J. Carey, Project Head
A. E. Kroger

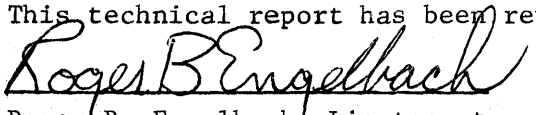


G. A. Hirschfield, Manager
Advanced Space and Range Projects



Levi J. Carey, Project Head
Space Programming Language

This technical report has been reviewed and is approved.



Roger B. Engelbach, Lieutenant
USAF Project Officer

ABSTRACT

This document contains basic functional requirements for SPL compilers. Descriptions of the general SPL compiler characteristics, such as the compiler capacities for program statements, names, signs, etc. are given. Input and output formats are described, and compool and library interfaces are specified. Error-detection, program-documentation, and quality-assurance provisions are specified. Other support programs for the SPL compilers are identified.

(page iv blank)

TABLE OF CONTENTS

CONTENTS

1. SCOPE

This document establishes the requirements for the design, test, performance, and qualification of computer programs identified as an SPL compiler and associated support programs. The standards for compatibility among SPL compilers are also stated.

2. APPLICABLE DOCUMENTS

The SPL specification (SPL Language Specification, SAMSO-TR-67-23) forms part of the compiler requirements. The reference document is obtainable from Defense Documentation Center (DDC).

3. REQUIREMENTS

3.1 GENERAL REQUIREMENTS

SPL compilers will process the language forms defined in the SPL Specification, SAMSO-TR-67-23.

The SPL compiler system shall be constructed so that the following capabilities are facilitated:

a. producing code for a new object computer.

b. transferring the SPL compiler to a new host computer for operation.

3.1.1 Programs

The SPL compiler system programs specified herein provide for the following:

a. Compiler System Control Program
b. Compiler
c. Compiler Compool Maintenance Program
d. Compiler Library Maintenance Program
e. Environment Documentation Program
f. Program Formatting Documentation Program

3.1.2 Documentation

a. SPL Programming Guide

b. SPL Operator's Manual

4. SPL COMPILER SYSTEM PROGRAMS

4.1 COMPILER SYSTEM CONTROL PROGRAM

The system control program shall execute each program in the SPL compiler system and provide the primary interface with any executive, monitor or operating system control program. The compiler system control program will read and interpret only compiler system control inputs.

4.2 COMPILER

4.2.1 General Compiler Requirements

4.2.1.1 The compiler shall produce object programs from any program written in SPL and within the processing constraints specified herein.

4.2.1.2 Compilation shall not require manual intervention or control input other than that specified in Section 4.2.5.

4.2.1.3 The compiler shall accept programs from card or magnetic tape input media.

4.2.1.4 The compiler shall assign a sequence number to each simple SPL statement, declaration, or command in the program and in procedures and functions called from the system library.

4.2.2 Compool Interface

The compiler will process compool-defined names. The compiler shall not allocate space in the object program for compool-defined names. It shall obtain locations for these names from the compool.

4.2.3 Library Interface

The compiler will process calls for library functions and procedures as defined in the SPL Specification, SAMSO-TR-67-23.

4.2.4 Executive Interface

The compiler shall operate under control of a compiler system control program independently of an executive system. The compiler shall be capable of producing both dependently and independently operating programs.

4.2.4.1 Independent Operation. The compiler will be capable of producing programs which can be loaded and operated independently of an executive system.

4.2.4.2 Dependent Operation. The compiler will also be capable of producing programs which will be operable under the control of an executive system. The object program generated by the compiler shall conform to the specifications of the executive interface and the loader interface.

4.2.4.3 Object Program Output. The binary object program shall be optionally relocatable or absolute.

4.2.5 Control Input

The syntax for control input statements for compilation shall be similar to the SPL syntax. For a dependent operation (using an executive system) control statements are implementation-dependent.

4.2.5.1 Control Options. Control input for control options shall precede each program in a batch. Control options include the following:

a. The program name
b. Debug-aid selections
c. Compool selection
d. Procedure library selection

e. Program input format, identifying the input line columns in which the program appears

f. Devices for input and output of the compilation

PROGRAM SHOULD BE FIRST

4.2.5.2 Names. All names shall be treated as defined in the compool, library, or program and they shall be searched for by the compiler in that order.

4.2.6 Program-Error Processing

The compiler shall detect, report and attempt to correct each statement, declaration, or command that contains a combination of symbols not defined in the SPL specification.

4.2.6.1 Error Detection. If the compiler detects one or more errors in a statement, declaration or command that prevents detection of further errors, it may suspend processing until the statement is terminated. The compiler shall resume error detection when it encounters the next statement. The detection of an error in a program shall not terminate error checking throughout the remainder of the program.

4.2.6.2 Error Reporting. The compiler shall report each error it detects in a manner that will give the programmer maximum assistance in locating the error and determining its cause. The compiler shall not issue an error report for reference to names whose declarations contain a grammar error. The compiler shall report each error in the following format:

**** ERROR a IN STATEMENT b c d

a = Key to the error list that describes the violated grammar rule or corollary

b = Sequence number of the statement in error

c = Symbol being processed when the compiler detected the error condition

d = Additional information, including key to corrective action taken

The error message shall be offset to the left of the program listing and shall follow the statement in error.

At the end of the object-program listing, the compiler shall list all locations at which direct-code errors were detected.

At the end of the object-program listing, the compiler shall print the total number of all program errors detected.

4.2.7 Capacities

The general guidelines for program capacities for an SPL compiler are shown below. The term "compool-defined names" refers only to compool-defined names in the program and in library procedures and library functions called in the program. The terms "library procedures" and "library functions" refer only to library procedures and functions called in the program.

4.2.7.1 Names. The SPL compiler will provide for at least 300 unique names in an SPL program. The unique names in the program include names in library procedures and library functions. Names that are unique within a procedure or function, that it, local names, are also included in the total.

4.2.7.2 Characters. The SPL compiler will provide for at least 10,000 characters in an SPL program, excluding comments and redundant spaces, that can occur in all statements, declarations, or commands. The number of characters includes the characters in library procedure headings and library function headings, and includes the characters that would be present in declarations for compool-defined names if these entities were fully declared in the program.

4.2.7.3 Statements. The SPL compiler will provide for at least 500 statements, declarations, and commands in an SPL program.

4.2.7.4 Words. The SPL compiler will provide for at least 25 symbols (words and marks) in any one simple statement in an SPL program.

4.2.7.5 Brackets. The SPL compiler will provide for at least 10 levels of nested brackets in an SPL program.

4.2.8 Compiler Outputs

The version number of the compiler shall be printed on the first page of symbolic output. Each page heading shall contain the program name, compool identification, and page number. A single pagination sequence shall be used for the compilation of each program. All error messages shall be printed, regardless of debug aids selected or suppressed. Compilation time shall be indicated on the last page of the source listing. Control cards shall be listed on the first page of compiler output.

4.2.8.1 Standard Outputs. Unless suppressed by control input at compile time, the following outputs shall be provided:

a. Source program listing
b. Library source program listing
c. Object program listing

4.2.8.2 Source Program Listing. The source program listing shall contain all program input lines processed by the compiler, plus error messages for program errors detected prior to or during the production of the source program listing. In addition, where the notational definition feature has been exercised, the SPL equivalent to any newly defined form will be printed. The compiler shall print each line of the source program listing as follows:

a. First field--statement sequence number
b. Remainder of line--original input line
c. If notational definitions have been used, the next line will contain the SPL equivalent code offset to the right.

The sequence number assigned by the compiler shall appear in the source program listing at the beginning of each input line. If an input line contains more characters than can be printed on a line, the compiler shall format the line in as meaningful a manner as possible, as specified by the procuring agency.

Characters inserted by the compiler shall not appear in the source program listing as part of the original input line.

4.2.8.3 Library Source Program Listing. The library source program listing shall be identified in the source program listing.

4.2.8.4 Object Program Listing. The compiler shall produce an object program listing containing all instructions, data, and constants generated by the compiler for the program. The statement number shall be included as a comment on each line. One output line shall be used for each machine instruction.

The first field, the address field, shall contain the address (or relative address) of the memory cell.

The second field, the machine code field, shall contain the machine language representation of the contents of the memory cell. The contents of the second field shall determine the spacing of the instruction subfield, modifier subfield(s), index register subfield(s), etc. All relocatable addresses in this field shall be flagged.

The third field, the symbolic field, shall contain the symbolic representation of the object code contained in the second field. The format of this field shall be that established for direct code. This symbolic representation shall contain internal sequence tags generated by the compiler, a symbolic representation of each instruction, the symbolic representation of addresses, etc. The address subfield or variable subfield(s) of this output shall follow direct code conventions. If an instruction refers to the location of a name, the compiler shall include the name in either the third or the fourth field.

The fourth field, the comment field, shall contain identification of error-monitor procedure calls and loop control object code. This field shall contain the name referred to by an instruction and may contain additional information.

4.3 COMPOOL REQUIREMENTS

The compool shall be a file containing computer representations of descriptions of items, arrays, groups, files, external programs, procedures, and functions. These descriptions shall include the following information:

a.	item	name, core-memory address and item declaration information
b.	array	name, core-memory address and array declaration information
c.	group	name, core-memory address, group description
d.	file	name, file-declaration information
e.	external program	name, core-memory address, and length
f.	procedure	name, core-memory address, formal parameter list with a description of each name therein, and length

The structure of the compool shall be expressible by data declarations. Program-system descriptions should be analyzed to determine specific compool requirements. Locations for names may be represented in several ways depending upon the method used to allocate storage.

4.4 LIBRARY REQUIREMENTS

The procedure library shall be designed to permit library procedures and functions to call other library procedures and functions. Library maintenance capabilities shall be provided to delete, add, change or duplicate a library procedure or function.

4.5 ENVIRONMENT DOCUMENTATION PROGRAM

The SPL environment documentation program will provide information on all data referencing (set/used), all statement name referencing, and maps of storage utilization for an SPL program. The environment documentation program will provide options to obtain any one of or all of the environment documentation outputs.

4.6 PROGRAM FORMATTING DOCUMENTATION PROGRAM

The SPL program formatting documentation program will provide a reorganized program listing in a standard form. This form will organize data declarations such that each data declaration will occupy one line and be left justified on the page. All statement names shall be left justified. Each SPL statement shall be on a single line. Statements will be indented at least four character positions. Nested compound statements will be indented for each separate nesting at least four character positions. This form will also include a reduction of any define commands to SPL equivalents.

5. DOCUMENTATION

5.1 SPL PROGRAMMING MANUAL (APPENDIX)

The SPL programming manual shall contain all information necessary for preparing an SPL program for compilation. This document will be common for all SPL users and installations. The SPL programming manual appendix will contain information specific to a single implementation, including special language forms implemented on that compiler, methods of selecting debug aids and other control input options,

and a list of errors detectable by the compiler. Further, it shall contain example usage of any extended SPL language form implemented.

5.1.1 Language Summary

The SPL programming language appendix shall include machine-specific or application-specific terms, and shall describe the limits imposed by the compiler on numbers of variables which may be declared, number of nested brackets, number of loop variables which may be defined at one time, etc. If limits are variable, that is, if they depend on a number of factors, a formula shall be given to determine the total limit.

5.1.2 Error List

A complete list of all error keys which may be reported by the compiler shall be ordered by error key number. A description shall be given of each error, listing the violated grammar rule or its corollary. As much information as possible shall be included in this description to assist the programmer in discovering his error. The action taken by the compiler to correct the error shall be given. The action shall indicate where in the statement compiling resumes and what type of debug code is produced. A list of equipment errors detected by the compiler shall include a condensed version of the information in the operator's manual for that error. A list of the operator and system software errors detected shall be included with equipment errors.

5.1.3 Debug Aids

All debug aids available to the programmer shall be described. All listings shall be described in detail. Special features available in each option shall be identified and described.

5.1.4 Equipment Configuration

The equipment configuration shall list the required peripheral equipment. It shall refer to the control options used to assign optional equipment.

5.1.5 Control Options

Control options shall be described. The structure of control cards or other control input required to compile a program shall be described. Illustrations shall show the correct inter-relationships between each control card or control deck for each combination of control options.

5.2 SPL OPERATOR'S MANUAL

The SPL operator's manual shall contain all information necessary to the computer operator to aid him in operating each program. Included shall be information on how to run the compiler, how to efficiently include compilation in the overall job flow, and how to assist the programmer in determining equipment or monitor errors.

5.2.1 Equipment Configuration

The equipment configuration shall list required and optional equipment. A description of how the use of optional equipment affects operation shall be included; this shall include the impact on compiler efficiency and speed.

5.2.2 Communication

Methods for communicating with each program shall be described. Operator and programmer control options shall be listed spearately.

5.2.3 Calling Procedure

The procedure to start the compiler and compool programs operating or to call them through the monitor shall use terms that are standard for the object computer. Procedures for reinitiating the program after equipment malfunction shall be fully described. Procedures provided in the program for restarting a job after error stops have occurred shall be described.

5.2.4 Error Stops

Each error stop, which may be either a program halt or a monitor pause following a machine or monitor error, shall be listed. Included shall be a reference to the particular procedure used for recovering or aborting jobs. In addition, guidelines shall be provided to assist the operator in determining which alternative recovery procedures he should use.

5.2.5 Operating Description

A description of the operation of the compiler shall be included to help the operator maintain an efficient job flow. File usage for each phase of the compiler shall be described.

6. QUALITY-ASSURANCE PROVIVIONS

To insure that the programs meet the requirements specified throughout Section 4, the following quality-assurance provisions shall be provided.

6.1 TEST REQUIREMENTS

The test requirements document shall outline the test for the compiler programs and associated support programs, including criteria for test abort, test restart, and retest.

6.1.1 Compiler Test Requirements

The compiler test requirements shall provide for the following classes of tests:

a. Class 1

   Class 1 shall measure the accuracy of processing the SPL language forms defined in the SPL specification, SAMSO-TR-67-23.

b. Class 2

   Class 2 tests shall measure the accuracy of program-error processing.

c. Class 3

The Class 3 tests shall verify that the compiler operates within specified limits and capacities.

d. Class 4

Class 4 tests shall include tests for compool utilization, library processing, and control option processing.

e. Class 5

Class 5 tests shall include tests and analysis of the efficiency of the object code produced by the compiler.

The compiler will be deemed validated when it has processed all of the tests correctly, and when the compiler has met specified performance requirements (Section 4).

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
SYSTEM DEVELOPMENT CORPORATION
2500 COLORADO AVENUE
SANTA MONICA, CALIFORNIA 90406

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE
COMPILER REQUIREMENTS FOR SPACE PROGRAMMING LANGUAGE (SPL)

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)
L. J. CAREY AND A. E. KROGER

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
AUGUST 1967	19	

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
FO 4695-67-C-0096 SPACE PROGRAMMING LANGUAGE STUDY b. PROJECT NO.	
c. d.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) SAMSO TR-67-23

10. DISTRIBUTION STATEMENT
This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the Space & Missile Systems Organization, Space Guidance Branch (SMTGS).

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	SPACE AND MISSILE SYSTEMS ORGANIZATION AIR FORCE SYSTEMS COMMAND LOS ANGELES, CALIFORNIA 90045

13. ABSTRACT

This document contains basic functional compiler requirements for SPL compilers. Descriptions of the general SPL compiler characteristics, such as the compiler capacities for program statements, names, signs, etc., are given. Input and output formats are described, and compool and library interfaces are specified. Error-detection, program-documentation, and quality-assurance provisions are specified. Other support programs for the SPL compiler are identified.

DD FORM 1 NOV 65 1473

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Programming Language						
Compiler Design						
Compiler Requirement						
Information Processing Function						