

PROGRAM ABSTRACT COVER SHEET

① User Group: FOCUS VIM (INCOSL)

Please complete this form according to the instructions on the reverse side

② Contributing Organization PURDUE UNIVERSITY, IDA, BEL LABS <hr/> Installation Name LAFAYETTE, INDIANA <hr/> City and State	③ Author Identification R. S. GAINES <hr/> Programmer/Submitter (up to 19 characters) MICHAEL D. SHAPIRO <hr/> Revisor																												
④ Catalog Identification <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; text-align: center;">R</td> <td style="border: 1px solid black; text-align: center;">O</td> <td style="border: 1px solid black; text-align: center;">I</td> <td style="border: 1px solid black; text-align: center;">D</td> <td style="border: 1px solid black; text-align: center;">A</td> <td style="border: 1px solid black; text-align: center;">S</td> <td style="border: 1px solid black; text-align: center;">N</td> <td style="border: 1px solid black; text-align: center;">O</td> <td style="border: 1px solid black; text-align: center;">B</td> <td style="border: 1px solid black; text-align: center;">O</td> <td style="border: 1px solid black; text-align: center;">L</td> <td style="border: 1px solid black; text-align: center;">4</td> <td style="border: 1px solid black; text-align: center;">D</td> </tr> <tr> <td style="font-size: small;">Cl. Code</td> <td style="font-size: small;">Org. Code</td> <td colspan="7" style="font-size: small;">Program Name</td> <td style="font-size: small;">Rev.</td> </tr> </table>	R	O	I	D	A	S	N	O	B	O	L	4	D	Cl. Code	Org. Code	Program Name							Rev.	⑤ Operating System and Version SCOPE 3 MACE					
R	O	I	D	A	S	N	O	B	O	L	4	D																	
Cl. Code	Org. Code	Program Name							Rev.																				
⑥ Languages and Dialects (up to 21 characters) COMPASS	⑦ Configuration <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; text-align: center;">6</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">0</td> </tr> <tr> <td colspan="3" style="font-size: small;">Computer</td> <td style="font-size: small;">Other Information</td> </tr> </table>	6	0	0	0	Computer			Other Information																				
6	0	0	0																										
Computer			Other Information																										
⑧ Descriptive Title (up to 56 Characters Including Blanks) PURDUE/IDA SNOBOL4 (VERSION 3.6)																													
⑨ Program Materials Submitted <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">Write-Up</td> <td style="width: 10%; border: 1px solid black; text-align: center;">5</td> <td style="width: 10%; border: 1px solid black; text-align: center;">7</td> <td style="width: 10%;">Source Record</td> <td style="width: 10%; border: 1px solid black; text-align: center;"> </td> <td style="width: 10%;">Source Medium</td> <td style="width: 10%; border: 1px solid black; text-align: center;"> </td> <td style="width: 10%;"> MT <input checked="" type="checkbox"/> CD <input type="checkbox"/> PT <input type="checkbox"/> </td> <td style="width: 10%;"> If MT <input type="checkbox"/> or PT <input type="checkbox"/> </td> <td style="width: 10%; border: 1px solid black; text-align: center;">1</td> <td style="width: 10%; border: 1px solid black; text-align: center;">7</td> <td style="width: 10%; border: 1px solid black; text-align: center;">6</td> <td style="width: 10%; border: 1px solid black; text-align: center;">0</td> <td style="width: 10%; border: 1px solid black; text-align: center;">0</td> </tr> <tr> <td>Page Ct</td> <td colspan="2"></td> <td>Count</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td style="font-size: small;">No.</td> <td style="font-size: small;">Tr/L</td> <td colspan="3" style="font-size: small;">Length</td> </tr> </table> Other (up to 44 characters) THE SNOBOL4 PROGRAMMING LANGUAGE (2ND EDITION)		Write-Up	5	7	Source Record		Source Medium		MT <input checked="" type="checkbox"/> CD <input type="checkbox"/> PT <input type="checkbox"/>	If MT <input type="checkbox"/> or PT <input type="checkbox"/>	1	7	6	0	0	Page Ct			Count						No.	Tr/L	Length		
Write-Up	5	7	Source Record		Source Medium		MT <input checked="" type="checkbox"/> CD <input type="checkbox"/> PT <input type="checkbox"/>	If MT <input type="checkbox"/> or PT <input type="checkbox"/>	1	7	6	0	0																
Page Ct			Count						No.	Tr/L	Length																		
⑩ Date Written <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; text-align: center;">M</td> <td style="border: 1px solid black; text-align: center;">A</td> <td style="border: 1px solid black; text-align: center;">R</td> <td style="border: 1px solid black; text-align: center;">1</td> <td style="border: 1px solid black; text-align: center;">9</td> <td style="border: 1px solid black; text-align: center;">6</td> <td style="border: 1px solid black; text-align: center;">9</td> <td style="border: 1px solid black; text-align: center;">A</td> <td style="border: 1px solid black; text-align: center;">U</td> <td style="border: 1px solid black; text-align: center;">G</td> <td style="border: 1px solid black; text-align: center;">1</td> <td style="border: 1px solid black; text-align: center;">9</td> <td style="border: 1px solid black; text-align: center;">7</td> <td style="border: 1px solid black; text-align: center;">1</td> </tr> <tr> <td colspan="6" style="font-size: small;">Original</td> <td colspan="7" style="font-size: small;">Revised</td> </tr> </table>	M	A	R	1	9	6	9	A	U	G	1	9	7	1	Original						Revised							⑪ Restricted: No <input checked="" type="checkbox"/> Yes <input type="checkbox"/> (Requires ordering information) Reason: Classified <input type="checkbox"/> Geographic <input type="checkbox"/> Other <input type="checkbox"/>	
M	A	R	1	9	6	9	A	U	G	1	9	7	1																
Original						Revised																							
⑫ Required Library Routines <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; text-align: center;"> </td> <td style="border: 1px solid black; text-align: center;"> </td> <td style="border: 1px solid black; text-align: center;"> </td> <td style="border: 1px solid black; text-align: center;"> </td> <td style="border: 1px solid black; text-align: center;"> </td> <td style="border: 1px solid black; text-align: center;"> </td> <td style="border: 1px solid black; text-align: center;"> </td> <td style="border: 1px solid black; text-align: center;"> </td> <td style="border: 1px solid black; text-align: center;"> </td> <td style="border: 1px solid black; text-align: center;"> </td> <td style="border: 1px solid black; text-align: center;"> </td> <td style="border: 1px solid black; text-align: center;"> </td> <td style="border: 1px solid black; text-align: center;"> </td> <td style="border: 1px solid black; text-align: center;"> </td> </tr> <tr> <td style="font-size: small;">Cl. Code</td> <td style="font-size: small;">Org. Code</td> <td colspan="7" style="font-size: small;">Program Name</td> <td style="font-size: small;">Rev.</td> </tr> </table>																Cl. Code	Org. Code	Program Name							Rev.				
Cl. Code	Org. Code	Program Name							Rev.																				
⑬ Entry Point Names																													
⑭ Original/Revised Program Abstract <p> SNOBOL4 IS A COMPUTER PROGRAMMING LANGUAGE DESIGNED FOR HANDLING A VARIETY OF DATA TYPES, MOST NOTABLY STRINGS OF CHARACTERS. IT IS AN OUTGROWTH OF EARLIER LANGUAGES DEVELOPED BY BELL LABORATORIES, AND HAS PROVEN USEFUL IN MANY TYPES OF PROGRAMMING RESEARCH AND DEVELOPMENT, AS WELL AS IN SOME PRODUCTION WORK. THE SYSTEM CONTAINS A COMPILER AND INTERPRETER. IT IS WRITTEN IN MACHINE-INDEPENDENT MACRO CODE, ALLOWING A VARIETY OF MACHINES TO HAVE VIRTUALLY IDENTICAL SYSTEMS. THE RELEASE CORRESPONDS TO BELL LABS VERSION 3.6. OPTIONAL MATERIAL WHICH EXTENDS SNOBOL4 TO USE EXTERNAL (FORTRAN, COMPASS, ETC.) FUNCTIONS IS INCLUDED. </p>																													
⑮ Nature of Revision <input checked="" type="checkbox"/> Proprietary Ordering Information <input type="checkbox"/> Additional Information <input type="checkbox"/> UPDATE FROM VERSION 3.4 TO VERSION 3.6.																													

E R R O R C O R R E C T I O N

for Purdue/IDA SNOBOL4 (v. 3.6)

Friday, August 13, 1971

From: Michael D. Shapiro
Purdue University Computing Center
Lafayette, Indiana 47907

An error has been noted in the LINKOR macro in the recently distributed version 3.6 of SNOBOL4, causing an internal loop in the routine which constructs alternatives in patterns. If the correction identifier SNOMIA has not been applied to your update deck, please add it:

```
*IDENT SNOMIA
*/ CORRECTION TO ERROR IN SNOMI
*DELETE MACRO.773
      BX5      X5-X5
*DELETE SNOM1.33
+      SA4      X5+B2
```

RECEIVED

AUG 16 1971

SOFTWARE
DISTRIBUTION

Purdue/IDA SNOBOL4 (Version 3.6)
For the CDC 6000 Series Computers

System Programmer Notes

Author M. D. Shapiro
Purdue University Computing Center
Mathematical Sciences Building
Lafayette, Indiana 47907

Date July 1, 1971

These notes provide system programmer information on the Purdue/IDA SNOBOL4 (Version 3.6) system for the CDC 6000 series computers. This version is an upgrade from the previously released version 3.4, with all known bugs removed and several macros and subroutines improved. User documents are unchanged.

The first section describes the distribution tape format. This is followed by instructions for preparing and testing the system. A section on internal data structures is next, followed by information on field length. Finally there is a short section on future work.

The SNOBOL4 system is furnished as a courtesy of Purdue University, I. D. A., and Bell Telephone Laboratories, Inc.

No warranty is made for the performance or quality of the system, nor can any agreement be made to maintain it. Every attempt will be made, however, to provide as much assistance as possible, commensurate with other responsibilities.

REFERENCES

1. CDC 6000 SNOBOL4 (Version 3) User's Guide, M. D. Shapiro, Purdue University Computer Science Center, October 1, 1970.
2. The SNOBOL4 Programming Language, (second edition), R. E. Griswold, J. F. Poage, and I. P. Polonsky, Prentice Hall, 1971.
3. A Guide to the Macro Implementation of SNOBOL4, R. E. Griswold, Bell Laboratories, Document S408c, February 27, 1970.

Distribution Tape

The SNOBOL4 symbolics are distributed on a 556 bpi tape containing one file. It is an UPDATE (Version 1.2) program library containing the decks:

1. MACRO - Macros required for assembly of the machine-independent code. Must be used whenever SNOBOL is used.
2. SNOBOL - Machine-independent macro-code distributed by Bell laboratories, maintaining their original numbering for reference.
3. SUBR - Machine-dependent subroutines called by SNOBOL.
4. SYNTAX - Syntax tables.
5. SYSIO - Cut-down versions of SCOPE 3.1 routines SYSTEM, INPUTC, OUTPTC (removing unused code). Modifications may be required for individual installations.
6. XLINK - External linkage routine (see R0-SNOSYSX)
7. SYNGEN - SNOBOL4 program and data to generate syntax tables.
8. TESTS - SNOBOL4 program test decks.

The following correction modifiers have been applied:

1. VER30 - Corrections to original source for version 3.0.
2. VER34 - Corrections to update original source to version 3.4 (YANKed by VER36)
3. VER34A - Corrections to version 3.4 (YANKED by VER36)
4. VER36 - Corrections to original source for version 3.6 (YANKs VER34 and VER34A)
5. CLOCK - Add CLOCK() function.
6. TEST1,...,TEST10 - Test programs
7. SMODO - Initial corrections.
8. SMOD1 - Version 3.6 corrections to macros.
9. SMOD2 - Version 3.6 corrections to subroutines.
10. SNOX1 - Corrections to XLINK.
11. LCCLGO - Change loader control card to write overlay file on LG0.
12. TEST9A - Correction to test program 9.

Preparing the System

The following deck setup may be used to assemble the system routines and prepare an overlay deck on the file LGO. The deck is then copied onto the file SNOBOLX, which is made a common file for testing.

```
Job Card
REQUEST(OLDPL,...) SNOBOL4 Update tape.
REWIND(OLDPL)
UPDATE(Q)
REWIND(COMPIL)
COMPASS(I=COMPILE,B=BIN)
MAP(ON)
LOAD(BIN)
NOGO.
REWIND(LGO)
COPYBF(LGO,SNOBOLX)
COMMON(SNOBOLX)
7/8/9
*IDENT MODS
  Correction cards, if any.
*COMPILE MACRO
*COMPILE SNOBOL
*COMPILE SUBR
*COMPILE SYNTAX
*COMPILE SYSIO (If desired)
6/7/8/9
```

With the common file created, tests can be run as follows:

```
Job card
COMMON(SNOBOLX)
SNOBOLX.
7/8/9
Test deck
6/7/8/9
```

The test decks can be obtained by using the same UPDATE structure as above, except punching the COMPILE file instead of assembling it. The UPDATE directive card is:

*COMPILE TESTS

The output on COMPILE is ten decks, with record marks between them.

Data Structures

General information about the implementation structure can be found in reference 3. The basic data item is a descriptor, which in this implementation for the CDC 6000 machines, has the structure:

Descriptor (general format)

59	29	23	17	0
Address	0	Flag	Value	

Descriptor (integer value)

59	29	23	17	0
Integer	0	0	6	

Descriptor (real value)

59	23	17	0
Real value	0	7	

A string is designated by a descriptor pair, called a specifier, which has the format:

String specifier

59	53	47	29	23	17	0
0	B.0.	Word offset	0	0	String length	
0	0	First word addr.	0	Flag	Value	

The bit offset (B.O.) is the number of bits from the start (bit 59) of a word where the string text begins. The word offset (W.O.) is the number of words after the first word address in which the desired text starts. These can be calculated from the character offset by:

$$\begin{aligned} \text{W.O.} &= \text{C.O.}/10 \text{ (Integer division with truncation)} \\ \text{B.O.} &= 6 \times [\text{C.O.} - (10 \times \text{W.O.})] \end{aligned}$$

The character string is packed ten 6-bit display code characters per word, using the entire CDC character set as specified for SNOBOL4.

The "binary 00" character is considered the 64th character for ^ALPHABET and LGT() purposes, and as a blank for compilation.

Syntax Tables

Each table is a 64-word entry, containing executable code which is plugged into the subroutine STREAM. The deck SYNGEN contains a SNOBOL4 program and data for preparing the tables. It can be used for modifying the syntax tables if desired. To run this program and assemble new syntax tables, use the control cards:

```
Job card
SNOBOL(,SYNTAX)
REWIND(SYNTAX)
COMPASS(I=SYNTAX,B=PUNCHB,S=0)
7/8/9
  SYNGEN program and data
6/7/8/9
```

Field Length

Version 3 of SNOBOL4 appears in nearly every way to the operating system to be a FORTRAN program. The only exception is with a few system communication macros (e.g. TIME, ABORT) in the subroutines (SUBR).

If automatic field length reduction is provided at an installation for FORTRAN programs, it must be suppressed. Check the beginning of SUBR for proper linkage to the routine Q8NTRY for any installation.

SNOBOL4 uses as free storage space the region from the start of blank common through the end of the field length. The more space a user provides, the more SNOBOL4 has to use, and the fewer storage regenerations are required. One program tested required 118 regenerations at 77700₈ words, 184 at 77000₈, and 534 at 76000₈. At 75000₈ it was still going after 1000 regenerations when dropped.

The system, when loaded with all the system routines, has a field length (including 5000₁₀ words of blank common) of a little less than 56000₈ words.

This is sufficient for short programs. These figures assume the use of cut-down versions of SYSTEM, INPUTC and OUTPTC. Symbolics are included as deck SYSIO, but will probably require modification for each installation.

Looking Forward

The following items are being worked on:

1. Compatible extensions of the input/output procedures, allowing dynamic allocation of I/O buffers.
2. Speed increases by improvement of storage allocation mechanism.
3. Improvement of macro-generated code by post-processing and inter-macro communications.

SNOBOL4 (Version 3) External Functions
System Programmer Notes

Author: Michael D. Shapiro
Purdue University Computing Center

Date: December 11, 1970

SNOBOLX is a relocatable version of SNOBOL, existing as a system common file. SNOBOLX contains the following modifications from SNOBOL:

1. The routines XLOADER, XNLOADR, and XLINKER have been removed from the program SUBR. These are dummy routines in the standard version, making LOAD and UNLOAD undefined functions.
2. Program XLINK has been added, containing the entry points deleted from SUBR. XLOADER is a routine which scans the external function table, XFTABLE, to find the entry point address of the desired routine. XLINKER is a routine to process the arguments, link to the external routine, and process the result. XNLOADR performs no operation. Two additional routines, SRETURN and FRETURN are used for special returns, as described in the programming notes.
3. XLINK contains two externals, XFTABLE and UNDEFIN which must be an external function table and an undefined value, respectively. The external function table structure is described in the programming notes. The undefined value is the value provided by the loader for undefined addresses. If XFTABLE and UNDEFIN are the same, no external table is believed to exist and any external function referenced is considered undefined. Any entry in the table which matches UNDEFIN is considered undefined. If an external function is undefined, the LOAD function call for it fails.

The system, as distributed, is in the most general form possible. Individual installations may wish to modify the LOAD function so the XFTABLE is no longer required, or to add dynamic loading, if available in the system. Unloading can be added, using the routine UNLOAD, if the system provides for it. In any case, the routine XLINKER will probably require little modification. If dynamic loading can be provided in an overlay program, it may not be necessary to provide two versions of SNOBOL on the system, although having a version of SNOBOL4 with full I/O format capabilities may be desirable.

Programming documentation should be changed to match whatever modifications are made for an individual installation. In particular, the MACE NOREDUCE card will be different at other installations, reflecting the convention used (e.g., REDUCE (OFF)).

This page left blank intentionally.

CDC 6000 SNOBOL4 (Version 3)

User's Guide

Michael D. Shapiro

Purdue University Computing Center

This document describes version 3 of the Purdue/IDA SNOBOL4 system for the CDC 6000 series computers. It acts as a supplement to The SNOBOL4 Programming Language (2nd edition), by Griswold, Poage, and Polonsky (Prentice-Hall, 1970). This description replaces all previous editions, which are obsolete and should be discarded.

October 1, 1970

This page contains no other essential information.

Version 3 of SNOBOL4 is the computer programming language described in The SNOBOL4 Programming Language (Second edition) by R. E. Griswold, J. F. Poage, and I. P. Polonsky (Prentice-Hall, 1970). It is an extension and improvement over the previously distributed version 2 release. This guide provides user information for Purdue/IDA SNOBOL4 (version 3) for the CDC 6000 series computers, and is supplemental to the referenced manual.

DECK SETUP

The following deck setup is used for a "standard" run:

Job card with CM65000 or larger.
SNOBOL.
7/8/9 end-of-record.
Program, terminated by END statement.
Data cards, if any.
6/7/8/9 end-of-information.

The section on Input/Output and Files provides additional information on the use of files for input or output.

IMPLEMENTATION RESTRICTIONS

The BACKSPACE function has been eliminated from the language.

The largest magnitude integer which may be used is $2^{29}-1$ and any attempt to use a larger value results in a failure, causing either statement failure or fatal termination.

In the termination dump of natural variables, the names are not completely alphabetized.

CHARACTER SET

Character set differences between the IBM System/360 computers used for initial preparation of the language and manual and the CDC 6000 series computers require changes in about half the punctuation characters when switching from one machine to the other. Upper case letters and digits are the same. Lower case letters and the underscore () character used on the System/360 have no equivalents on the 6000 machines.

These punctuation characters appear on both machines and have identical meaning in SNOBOL4:

() + - * / . , : ; = \$ ~ < >

While the graphics for these characters are the same on both machines, the punch combinations to produce them are not the same in all cases. All of the above graphics can be punched on an IBM 026 keypunch with these exceptions (which must be multipunched):

colon	:	2-8 punch combination
semicolon	;	12-7-8
less	<	12-2-8 or 12-0
greater	>	11-7-8
not sign	~	12-6-8

Other punctuation characters used in SNOBOL4 have different graphics on the CDC machines than described in the manual. The following table summarizes these equivalences and the multipunch combinations which produce the characters for the CDC equipment.

<u>Character in Manual</u>	<u>CDC 6000 Equivalent</u>	<u>Multipunch Combination</u>
"	#	4-8
'	†	11-5-8
<	[7-8
>]	0-2-8
&	^	0-7-8
	v	11-2-8 or 11-0
?	→	0-5-8
@	↓	11-6-8
#	≤	5-8
%	≥	12-5-8
!	≡	0-6-8

The 12 punch is a plus sign (+) and the 11 punch is a minus sign (-). The 4-8 punch (#) appears on most 026 keypunches as either an apostrophe (') or an at-sign (@). There is no distinction as to which of the two string delimiters (# or †) corresponds to the apostrophe and which to the quote marks ("), so long as the programmer remembers to maintain one convention for keypunching purposes. The square brackets are alternatives for the less and greater symbols used in the book and may be interchanged as desired.

INPUT/OUTPUT AND FILES

A few minor changes in input/output are required to SNOBOL4 programs, as described in the manual, to run them under the CDC 6000 operating systems. As in the OS version, standard FORTRAN I/O routines are used, with input or output being handled strictly as arrays of characters. Numeric conversions are handled internal to the SNOBOL4 system. Primary changes to information in the manual are:

1. The "data set reference numbers" become "file numbers". They have the following values:
 - 1 = normal input stream (file INPUT)
 - 2 = normal printed output (file OUTPUT)
 - 3 = normal Hollerith punched output (file PUNCH)
2. The variables INPUT, OUTPUT, and PUNCH have predefined I/O associations with their corresponding files.
3. One additional file number, 4, may be used. It is predefined to be a file named SCRATCH, but has no predefined I/O association.
4. The default format for printed output is (1X,135A1), reflecting the 136-column width of CDC printers.
5. The maximum length I/O record is 136 characters. Longer strings must be broken into several records for printing. This can be handled by format, as the default formats do.
6. In output format strings, the CDC convention of using asterisks (*) to delimit literal strings applies.

Any of the file names indicated above is alterable by designating a different name in the appropriate parameter position on the SNOBOL control card. The file number indicates the parameter position. For example, to place the output generated for the internal output association with PUNCH onto the file named COMPILE, the control card becomes:

```
SNOBOL (, , COMPILE)
```

The compiler reads the source program from whichever file is number 1. Thus to compile a program from the disk file COMPILER, the card becomes:

```
SNOBOL(COMPILER)
```

If, in addition, data cards are to be read from the system input file INPUT, the control card might become:

```
SNOBOL(COMPILER,,INPUT)
```

In order to read the data from INPUT, the following statement (perhaps with a different variable name) must be executed in the SNOBOL4 program:

```
INPUT(=INPUT#,4)
```

The I/O association functions have default values for file numbers of 1 for the INPUT function and 2 for the OUTPUT function. The following function calls are equivalent:

```
INPUT(=ABC#)    or    INPUT(=ABC#,1)
OUTPUT(=DEF#)   or    OUTPUT(=DEF#,2)
```

COMPILER INPUT

As indicated previously, the compiler reads the source program from file 1. Only the first 72 columns of the input cards are compiled. During compilation, single end-of-record cards are ignored. A source program can therefore be located in several consecutive records on a file. Compilation ends when:

- an END statement is compiled, or
- two consecutive end-of-record cards are read, or
- an end-of-information card is encountered.

In the last two cases, an END statement is generated, with an appropriate warning.

EXECUTION INPUT

At execution time, reading an end-of-record or end-of-information card causes the read to fail. If the read was of an end-of-record, the next read on the same file inputs the next card image. Subsequent reads of an end-of-information fail.

OUTPUT LINE LIMIT

The OUTPUT file line limit count is preset at 10000 (decimal) records. This count includes all items on the file produced by the SNOBOL4 system, as well as program output. The line count may be modified by a parameter on the SNOBOL control card. This parameter, which must precede the file name list, if any, is given in the form:

LC=count

where "count" is the new limit in OCTAL. For example, to change the limit to 100000 (octal) lines, the card is:

SNOBOL(LC=100000)

An example with a file list is:

SNOBOL(LC=100000,COMPILE,,,INPUT)

KEYWORDS

All keywords described in the manual are included in this version of the language. Two are machine dependent and require additional comments:

ALPHABET contains the 64-character alphabet of the CDC 6000 computers, in internal display code order (see CDC system reference manual for additional detail):

1 - 26	letters A - Z
27 - 36	digits 0 - 9
37 - 63	punctuation
64	"binary 00"

The "binary 00" character is special in that it cannot be input and on output it is converted to a blank. A convenient usage of this character is as an internal marker in strings.

If the keyword ABEND is nonzero at program termination, a MODE 0 rather than a normal exit is taken for returning to the system. The value of the keyword ABEND is loaded into register A0 and the system macro ABORT is used. Information remaining in the output buffers (perhaps several lines of printing) may be lost.

FUNCTIONS

All functions described in the manual except BACKSPACE are included in the system.

The value of TIME() is an integer which is the elapsed time in milliseconds from the beginning of program execution. Compilation time is not included. Resolution varies from installation to installation.

The function LGT uses the same lexical ordering as the keyword ALPHABET as a reference.

An additional function CLOCK is added to the system. The value is an eight-character string with the current system clock reading (HH:MM:SS). The argument list is ignored as in DATE and TIME functions.

External Functions for CDC 6000 SNOBOL4

Michael D. Shapiro

**Purdue University Computing Center
Lafayette, Indiana**

December 11, 1970

This page left blank intentionally.

I. Introduction

The SNOBOL4 language includes features which allow the user to define his own functions and data structures as additions to the primitive functions and basic data structures built into the language. External functions are a logical extension of primitive functions and programmer-defined functions. They are separately compiled subprograms written in languages other than SNOBOL4 which may be called by a SNOBOL4 program.

With this facility the SNOBOL4 programmer is free to choose other languages better suited to certain operations. The external function features are available in a special version of SNOBOL4 called SNOBOLX on the CDC 6000 machines. Among the capabilities the user has are:

- a) Access to FORTRAN library functions.
- b) Increased speed in performing operations which cannot be done as efficiently in SNOBOL4. Involved numerical computations and complex input/output are two such operations.
- c) Avoiding unnecessary variable generation and storage regeneration when, for example, scanning a data file for specific data records.
- d) Performing operations not otherwise available in SNOBOL4.

This paper describes usage of FORTRAN and CDC 6000 assembly language functions. The next section concerns definition and usage of functions within the program. The following section describes deck setups for running SNOBOL4 with FORTRAN external functions, including several examples. These should be sufficient for most users.

A final section is included for programmers wishing to write external functions in assembly language. Information is provided about such items as internal data structures, external data types, and register conventions.

This document is based, in part, on the paper "External Functions for SNOBOL4," by H. J. Strauss, Bell Telephone Laboratories, Holmdel, NJ (Document S4D10c, July 31, 1970), which describes the implementation of external functions for the IBM System/360 version of SNOBOL4 running under OS/360.

II. Defining External Functions

As with programmer-defined functions, two steps must be taken before external functions may be used in a SNOBOL4 program. The program code must be defined and the system must be told the function properties: the name, location, arguments and returned datatype.

The program written in FORTRAN or assembly language must be processed before the SNOBOL system is invoked. This can be done either in the same computer run or in an earlier run (using the "binary deck" produced on that run). Samples of deck setups for both these are shown in Section III. In addition, a function table, listing the program names and memory addresses must be provided. This is discussed later.

Before the function can be used, the function LOAD must be called to tell the system about the function. LOAD corresponds to DEFINE for programmer-defined functions.

The remainder of this section provides additional information about the linkage routines and other information the programmer must provide.

The Primitive Function LOAD

The primitive function LOAD provides information to the system about external functions. LOAD passes the following information to the SNOBOL4 system:

1. The function name.
2. The number of arguments expected by the external function.
3. The data type of each argument.
4. The data type of the value which the function will return.

The memory address of the function is found in the function table. The format of the LOAD call is:

```
LOAD('fname(arglist)result')
```

fname is the name of the function. It must be a valid SNOBOL4 name and must be a name in the function table (described later). Although the length of the actual function name must be seven characters or less, fname may be any length and is truncated to seven characters if longer. Thus MULTIPLIER, MULTIPLICATION and MULTIPLICITY all refer to the function MULTIPL.

arglist is a series of data type identifiers or null strings, separated by commas. For example:

```
INTEGER,,REAL,NODE,BOOLE
```

Each element defines the existence and data type of an argument to an external function. The elements have an ordered correspondence to the arguments of the external functions. An element of arglist may be any SNOBOL4 data type identifier, including data types defined by using the primitive function DATA. The null string or the data type of a structure defined in an external function may also be used. Data types defined in external functions are called external data types. Null elements indicate that an argument of any data type may be used when the function is invoked.

When the data type of an argument is explicitly stated, automatic conversion among INTEGER, REAL and STRING is performed when the function is invoked. Explicitly specifying the datatypes of arguments to an external function results in less ambiguous programs. If the number of arguments specified differs from the number expected by the external function, the results depend on the function. A maximum of six arguments may be used in the current implementation.

result indicates the datatype of the value returned by an external function. It may have any value that an element of arglist may have. If result is not INTEGER, REAL or STRING the function is expected to return a DESCRIPTOR defining the returned structure. (See the section on data structures.)

LOAD succeeds returning the null string, if the requested function is available and failing if it is not available.

External Function Invocation

External functions may be called in any context that primitive and programmer-defined functions may be called. At the time the function must have been defined with a LOAD (which did not fail). Linkage is thru the standard FORTRAN (RUN compiler) linkage. If the function is called with excessive arguments, they are evaluated but not passed as either integer or real zero or null string, depending on the data type used in the LOAD call. A maximum of six arguments may be used.

Returning Results from FORTRAN IV External Functions

Since external functions must return a single value, function subprograms are the only FORTRAN IV subprograms usable with SNOBOL4. If the type of function is not explicitly specified, the FORTRAN naming convention is assumed by the FORTRAN IV compiler. The SNOBOL4 system does not check the external function, but assumes that the value returned by the function is as specified in the "result" field of the defining LOAD function.

The result is returned in machine register "X6". This convention is assumed by the SNOBOL4 system. Multiple entry points are permissible, using FORTRAN conventions, if all entry points used are included in the function table.

Only integer and real results can be returned by the normal FORTRAN function manner. Two special routines have been added to the system to provide additional special returns peculiar to SNOBOL4:

CALL SRETURN(n,string)

returns a string value. The first argument n is an integer value indicating the length (number of characters) of the string. string is a word (or array of words) containing the character string (in "A10" format).

CALL SRETURN(0)

returns a null string value.

CALL FRETURN

can be used to cause the function to fail as a transfer to FRETURN does for a programmer-defined function.

The Function Table

The SNOBOL4 function table, provides the SNOBOL4 system with information about external functions which have been loaded into memory and forces the loader to load routines from the system library. The system assumes the table has an entry point named XFTABLE. If this entry is undefined (found by comparing to an undefined external UNDEFIN), the external function facility is not available. If they are not the same, XFTABLE is assumed to be the first word of a table containing the names and entry points of available functions. Each word is of the form:

59	18	17	0
name		address	

The name is assumed to be left justified display code with binary zero fill. The name is a maximum of seven characters long. The address is provided by the loader. The list is terminated by a word of zeros. The name is the name SNOBOL4 uses. This is not necessarily the function name FORTRAN would use, but it usually is the same to avoid confusion.

A very short program written in assembly language and assembled using either COMPASS by itself, or COMPASS called in by the FORTRAN (RUN) compiler, can be used to define the table. For each routine to be used, one line of assembly code is required:

```
column      11      18
            VFD     42/0Lname,18/=Xname
```

where name is the name of the routine.

The following is a complete sample program for linking library routines SIN and SQRT, and the programmer-provided routine ABCXYZ:

```
column 2    11      18
            IDENT   XFTABLE
            ENTRY   XFTABLE
XFTABLE    VFD     42/0LSIN,18/=XSIN
            VFD     42/0LSQRT,18/=XSQRT
            VFD     42/0LABCXYZ,18/=XABCXYZ
            DATA   0
            END
```

The program should be reproduced as it appears, except for routine names. Note that all 0's are the digit zero in the program.

The Primitive Function UNLOAD

The primitive function UNLOAD disconnects linkage to an external function, making it undefined. In a future extension, it may be possible to release the space used by the function for other operations, although this is not now possible. The format is:

```
UNLOAD('fname')
```

where fname is the name of any previously LOAded external functions. UNLOAD undefines programmer-defined functions as well as external functions.

III. Using the External Function Facility

This section shows several deck setups for using external functions with SNOBOL4. Each setup is introduced by a short description. Finally, a complete program is shown.

- A. Run a SNOBOL4 program using the FORTRAN library functions SIN and SQRT:

```
Job card with CM65000 or more
COMPASS.
COMMON(SNOBOLX)
NOREDUCE.
LOAD(LGO)
LOAD(SNOBOLX)
EXECUTE.
7/8/9
```

```

IDENT XFTABLE
ENTRY XFTABLE
XFTABLE VFD 42/OLSIN,18/=XSIN
VFD 42/OLSQRT,18/=XSQRT
DATA 0
END

```

7/8/9

SNOBOL4 program and data

6/7/8/9

B. Compile a FORTRAN IV external function and use it in a SNOBOL4 program:

```

Job card with CM65000 or more
RUN(S)
COMMON(SNOBOLX)
NOREDUCE.
LOAD(LGO)
LOAD(SNOBOLX)
EXECUTE.
7/8/9

```

```

FUNCTION ADD3(A,B,C)

```

```

...

```

```

END

```

```

IDENT XFTABLE
ENTRY XFTABLE
XFTABLE VFD 42/OLADD3,18/=XADD3
DATA 0
END

```

7/8/9

SNOBOL4 program and data

6/7/8/9

C. Load a previously compiled external function and table:

```

Job card with CM65000 or more
COMMON(SNOBOLX)
NOREDUCE.
LOAD(INPUT)
LOAD(SNOBOLX)
EXECUTE.
7/8/9

```

Binary deck(s) produced by previous compilations.
Each deck is terminated by a 7/8/9 card.

7/8/9 (in addition to 7/8/9 for last deck)

SNOBOL4 program and data

6/7/8/9

- D. If formatted read, print, punch, or write statements appear in the external functions, the following card must be added after the COMMON(SNOBOLX) card in each of the above setups:

COPYBR(SNOBOLX,SNOBOLZ,4)

and each of the LOAD(SNOBOLX) cards must be replaced by:

LOAD(SNOBOLZ)

As an example, sample C becomes:

```
Jobcard with CM70000 or more (note larger size)
COMMON(SNOBOLX)
COPYBR(SNOBOLX,SNOBOLZ,4)
NOREDUCE.
LOAD(INPUT)
LOAD(SNOBOLZ)
EXECUTE.
7/8/9
Binary deck(s)
7/8/9
SNOBOL4 program and data
6/7/8/9
```

IV. Assembly Language External Functions

This section describes the conventions to be used in writing assembly language functions for use with SNOBOL4. A knowledge of assembly language programming is assumed. It should be noted that any violation of the conventions may cause unforeseen difficulties, the scope of which is beyond this discussion.

Arguments

The RUN compiler linkage is used on input, with a few additions. Registers B1 through B6 contain addresses which point to the first through sixth arguments respectively. Register B7 contains a number from one through six, indicating the number of arguments which are valid. If B7 contains a four, for example, B5 and B6 will point to locations whose values are undefined for this call.

Register X7 contains the data type code of the desired result type:

1	STRING	7	REAL
2	BLOCK (internal)	8	CODE
3	PATTERN	9	NAME
4	ARRAY	10	KEYWORD
5	TABLE	11	EXPRESSION
6	INTEGER	12	LINK (internal)

Argument Values

Each argument is a single CDC 6000 word. If the argument is real or integer data type, the argument value is in standard floating point or integer format. If the argument is a string, the argument is in "external string descriptor" format. This format contains the address of the first word of the string in the high-order thirty bits and the number of characters in the string in the low-order thirty bits.

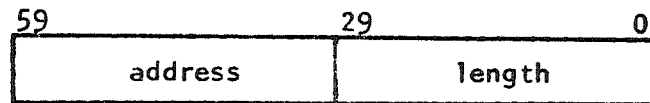


Figure 1. External String Descriptor Format

Each of the values is right-justified in its field. A null string is represented by a descriptor of all zero bits. The string, which is left justified in the first word, is actually part of a SNOBOL4 structure. This structure is the only copy of the string available to the SNOBOL4 system and should not be altered in any way. To modify the string, it should be copied.

Any other data type is passed in SNOBOL4 internal descriptor format. This format contains an address field in the high-order thirty bits, an internal code (flag code) in the next twelve bits, and a data type value in the low order eighteen bits.

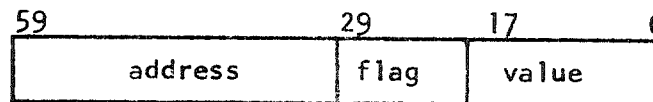


Figure 2. Internal Descriptor

For internal or programmer defined data structures, the address field points to the structure, while the data type is an integer value from 1 through 999. The flag code is used by the system, and normally should not concern the programmer.

Data type codes greater than or equal to 1000 (but still fitting as a positive number in the 18-bit field), are reserved for defining external data types. External data types can contain any value in the address field and the high order six bits of the flag field. The low order six bits of the flag field must be zeros.

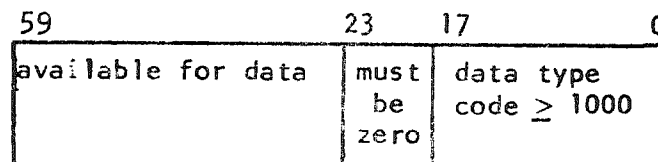


Figure 3. External Data Type Format.

The external data types can be used for such items as bit strings. Since they cannot be generated inside the SNOBOL4 system, the only way they can appear as arguments is if they have been defined previously by external routines.

Returning Values to the SNOBOL4 System

An external function is called using the "return jump" (RJ) instruction, and control is returned to the SNOBOL4 system by branching indirectly through the entry/exit word. Whatever value is in register X6 is considered the result. The result should be in the same format as an argument. If the system expects a real, integer, or string, the result is converted to the appropriate internal format (which is different). Any other data type is copied as an internal value. It is important that the rules for constructing descriptors be followed; otherwise, the results are unpredictable.

The string descriptor can be called by using the FORTRAN callable subroutine:

```
SB1    LENGTH
SB2    STRING
RJ     =XSRETURN
```

where LENGTH is the address of a word containing the length of the string and STRING is the address of the first word of the string (left justified) in A10 format.

A null string can be returned either by setting X6 to zero or by using SRETURN:

```
SB1    =0
RJ     =XSRETURN
```

Function failure can be indicated by using the routine FRETURN:

```
RJ     =XFRETURN
```

This page left blank intentionally.

A Bibliography of SNOBOL Publications
(excluding Bell Laboratories documents)

Compiled by Michael D. Shapiro

Bell Telephone Laboratories, Incorporated
Holmdel, New Jersey

Revised by Madge T. Hammer
Bell Telephone Laboratories, Incorporated
Holmdel, New Jersey

June 1, 1971

S4D12b

This page left blank intentionally.

Introduction

This bibliography references two categories of publications concerned with the SNOBOL programming language which have been published outside Bell Laboratories. Part 1 lists descriptions of SNOBOL or "SNOBOL-like" languages, including manuals for various implementations, discussions of features, and comparisons with other similar languages. It also includes publications which indicate that the systems described are intended to provide similar processing capabilities to SNOBOL. Part 2 contains information about uses of SNOBOL in the solution of various types of problems, covering both articles describing the applications and their programs, and announcements of work being done (e.g., brief project descriptions in the "Directory of Scholars Active" in Computers and the Humanities).

While every attempt has been made to find as many entries as possible, many sources have undoubtedly been missed. This is particularly true in the applications area (Part 2). The compiler would appreciate having additional material brought to his attention for inclusion in a revision of this document and wishes to thank those who have contributed material for this edition.

Entries marked with an asterisk have been added or revised since the previous version of this document was prepared.

This page left blank intentionally.

Part 1: Descriptions of SNOBOL or "SNOBOL-like" Languages

Abrahams, Paul W., "Symbol Manipulation Languages," Advances in Computers, volume 9, F. L. Alt and M. Rubinoff, editors, New York: Academic Press, 1968 (92-102).

Anderson, E. R., and Sturgeon, R., "Interactive SNOBOL4 System for the SDS 940," University of California (Berkeley), ARPA Contract SD-185, Document R-34, September 6, 1968.

*Applied Logic Corporation, "AL/COM SNOBOL-D (PRELIMINARY) USER'S MANUAL," Applied Logic Corporation, Princeton, N.J., July 1970.

Bailey, F. N., and Kain, R. Y., "An Exact Description of a SNOBOL Dialect," NASA Grant 24-005-063 and NSF Grant GK-40, University of Minnesota, Center for Control Sciences, Report number CS101-68 (1968).

Bolles, Robert C., "A SNOBOL4-like Pattern Matching Scheme and Its Use in a Syntax-directed Translator," Master's thesis, Moore School of Electrical Engineering, University of Pennsylvania, Report number 69-18, December 1968.

Bunker-Ramo Corporation, "SNOBOL4 Programming Manual," Reprint, with minor modifications, of BTL S4D4, October 1967.

California, University of (Berkeley), "SNOBOL4 Reference Manual," April 1969.

Chai, David T., "Some Remarks on SNOBOL4 Features," The Bunker-Ramo Corporation, Optical Character Reader Application Study, RADC Contract F30602-67-C-0169, Technical Note, November 1968.

Com-Share Incorporated, "SNOBOL Reference Manual," Reference number 9010-2, June 1968, Revised August 1968, Third edition, reference number 9010-3, May 1970.

Desautels, Edouard J., and Smith, Douglas K., "An Introduction to the String Processing Language SNOBOL," Programming Systems and Languages, Saul Rosen, editor, New York: McGraw-Hill, 1967 (419-454).

Dewar, R. B. K., and Manacher, G. K., "Some New File-handling, Input-output, and Monitoring Procedures for Use with the SNOBOL Language," University of Chicago (no date).

*Dewar, Robert B. K., "SPITBOL, Version 1.0," (no organization, no date).

- *Dewar, Robert B. K., "SPITBBOL, Version 2.0," Illinois Institute of Technology, February 12, 1971.
- D'Imperio, M. E., "Data Structures and their Representation," Annual Review in Automatic Programming, volume 5, Mark I. Halpern and Christopher J. Shaw, editors, London: Pergamon Press, 1969 (60-75).
- *Emerson, Philip, "PDP-9 SNOBOL Version A," DECUS No. 9-43a, The Cleveland State University, Cleveland, Ohio, February 1970.
- Farber, D. J., Griswold, R. E., and Polonsky, I. P., "SNOBOL, a String Manipulation Language," Journal of the ACM 11(1) January 1964 (21-30).
- Farber, D. J., Griswold, R. E., and Polonsky, I. P., "The SNOBOL3 Programming Language," Bell System Technical Journal 45(6) July-August 1966 (895-943).
- Fasana, Paul J., Shank, Russell, and Burgess, Thomas K., "Generalized Programming Languages and Systems," American Society for Information Science Tutorial Subcommittee, CFSTI report PB-182 229, 1968 (Instructors' edition), CFTSI report PB-182 230, 1968 (Students' edition).
- *Fleck, A. C., "TOWARDS A THEORY OF DATA STRUCTURES," Themis Project, University of Iowa, Technical Report No. 30, August 1970.
- Forte, Allen, "The Programming Language SNOBOL3: An Introduction," Computers and the Humanities, 1(5) May 1967 (157-163).
- Forte, Allen, SNOBOL3 Primer: An Introduction to the Computer Programming Language, Cambridge, Mass.: MIT Press, 1967.
- Gaines, R. Stockton, "Basic Elements of SNOBOL4," Institute for Defense Analyses, Princeton, N. J., Working Paper 224, March 1968.
- Gaines, R. Stockton, editor, "Preliminary Report on the SNOBOL4 Language," (Revision of BTL S4D4 to conform to the CDC 6000 series implementation), March 1, 1968.
- Goff, Adelbert G., "OS SNOBOL User's Manual for the System/360 at Brown University," Brown University Computing Review 1(2) June 1967 (36-63).
- Griswold, R. E., Poage, J. F., and Polonsky, I. P., The SNOBOL4 Programming Language, Englewood Cliffs, N.J.: Prentice-Hall, 1968.

- *Griswold, R. E., Poage, J. F., and Polonsky, I. P., The SNOBOL4 Programming Language, second edition, Englewood Cliffs, N.J.: Prentice-Hall, 1971.
- Griswold, R. E., "String Processing and the SNOBOL4 Language," International Summer School on Fundamental Aspects and Current Developments in Computer Science, Copenhagen, Denmark, August 11 to 23, 1969.
- Guard, James R., "SNOBOL-6" (for the PDP-6 Computer), Applied Logic Corporation, Princeton, N.J.
- *Guard, James R., "AL/COM," Program Bulletin No. 67-006, Applied Logic Corporation, Princeton, N.J., December 19, 1967.
- *Henninger, Ernest Henry, "A STUDY OF THE EFFICIENCIES IN THE MOBILE PROGRAMMING SYSTEM," Master's Thesis, Naval Postgraduate School, Monterey, California, June 1969.
- Hsu, R. W. and Gould, Laura, "A Linguist's Introduction to SNOBOL, Part 1," Pacific and Asian Linguistics Institute, University of Hawaii, Honolulu, 1966.
- Jeffries, Ronald E., "Choosing a Programming Language," Com-Share News 3(1) Spring 1970 (2-5).
- Kain, R. Y., and Bailey, F. N., "SNOBOL 67 User's Reference Manual," University of Minnesota, Dept. of Electrical Engineering, Edition 4, May 1, 1969.
- Kain, R. Y., "Block Structures, Indirect Addressing, and Garbage Collection," Communications of the ACM 12(7) July 1969 (395-398).
- Katzman, Harry J., Advanced Programming: Programming and Operating Systems, New York: Van Nostrand Reinhold, 1970 (36-48).
- Kimstadt, Eric P., "SNOBOL(-1) Programming Guide," September 22, 1966 (Affiliation not indicated).
- King, Gerald, editor, "The SNOBOL4 Programming Language and Its Relation to SNOBOL3," (Minor modification of BTL S4D4), University of Oregon, Statistical Laboratory and Computing Center, March 1968.
- Kostolansky, Eduard, "The Definition of the Syntax and Semantics of the Language SNOBOL I," Kybernetika 3 (1967) (in Slovak).

Kostolansky, Eduard, "Implementation of SNOBOL I," Proceedings of the Third Conference on Cybernetics, Bratislava, 1968 (in Slovak).

Lampson, Butler W., "930 SNOBOL System Reference Manual," University of California (Berkeley), ARPA Contract SD-185, Document 30.50.70, April 18, 1966.

Lavrov, Sviatoslav Sergeevich, "SNOBOL Adopted for Soviet Computers." Translation of excerpts from the monograph Snobol-A, Yazyk dlya obrabotki strok (Snobol-A, A String Manipulation Language), USSR Academy of Sciences, Computer Center, Moscow, 1968 (16 pages). Translated from the Russian by John Schneider, February 10, 1969, and appearing in Soviet Cybernetics 3(3) March 1969 (57-58).

Madnick, Stuart E., "CMS/360 SNOBOL User's Manual," IBM Cambridge Scientific Center, December 18, 1966.

Massachusetts Institute of Technology, "SNOBOL Implementation," Department of Electrical Engineering, April 28, 1966.

*Minow, Martin, "SNOBOL4," Modern Datateknik, April 1969, p. 61 (in Swedish).

*Pagan, F. G., "A Method of Linking PL/I to SNOBOL4," SIGPLAN Notices, 5(12), December 1970.

Raphael, Bertram, "The Structure of Programming Languages," Communications of the ACM 9(2) February 1966 (67-71).

Raphael, B., Bobrow, D. G., Fein, L., and Young, J. W., "A Brief Survey of Computer Languages for Symbolic and Algebraic Manipulation," Partial support: Air Force Contract AF-49(638)-1752, Presented at the IFIPS Working Conference on Symbol Manipulation Languages, Pisa, Italy, September 1966.

Raphael, Bertram, "Aspects and Applications of Symbol Manipulation," Proceedings of A.C.M. 21st National Conference (1966).

*Rinde, Joseph, "SNIP-SNOBOL INTERACTIVE PACKAGE," Carnegie-Mellon University, September 30, 1969.

*Raskin, Jeffrey F., "Programming Languages for the Humanities," Computers and the Humanities, 5(3), January 1971.

Rosin, Robert F., "Character Strings in General Purpose Procedural Languages," Yale University (no date).

- Rosin, Robert F., "Contemporary Concepts of Micro-Programming and Emulation," Computing Surveys 1(4) December 1969 (197-212).
- *Saal, Harry J., "MILTIE: PL/I and SNOBOL4 Interfaces for Terminal I/O through MILTEN," Stanford Linear Accelerator Center, CGTM 112, November 1970.
- Sammet, Jean E., "String and List Processing Languages," Programming Languages: History and Fundamentals (Chapter VI), Englewood Cliffs, N.J.: Prentice-Hall, 1969 (436-448).
- *Santos, Paul J., Jr. and Maurer, W. Douglas, "COMPILATION OF A SUBSET OF SNOBOL4," SIGPLAN Notices, 5(12), December 1970.
- Satterthwait, A. C., "Programming Languages for Computational Linguistics," Advances in Computers, volume 7, F. L. Alt and M. Rubinoff, editors, New York: Academic Press, 1966 (220-221).
- Shapiro, Michael D., "CDC 6000 SNOBOL4 (Version 2.0) User's Guide," Purdue University Computer Science Center, March 1969, Minor revision, August 1969.
- Shapiro, Michael D., "Purdue/IDA SNOBOL4 (Version 2.0) for the CDC 6000 Series Computer-System Programmer Notes," Purdue University Computer Science Center, March 1969.
- *Shapiro, Michael D., "Version 3 of SNOBOL4," (a modification of the document S4D14b, Bell Telephone Laboratories, Incorporated, for the CDC 6000 series implementation), Purdue University Computing Center, October 1970.
- *Shapiro, Michael D., "System Programmer Notes," (Purdue/IDA CDC 6000 series implementation), Purdue University Computing Center, October 1, 1970.
- *Shapiro, Michael D., "CDC 6000 SNOBOL4 (Version 3) User's Guide," Purdue University Computing Center, October 1, 1970.
- *Shapiro, Michael D., "SNOBOL3 to SNOBOL4 Conversion Guide," Purdue University Computing Center, October 1970.
- *Shapiro, Michael D., "SNOBALT-SNOBOL4 for PROCSY TERMINALS," Purdue University Computing Center, November 1970.
- *Shapiro, Michael D., "An introduction to character string operations using FORTRAN IV and the Purdue University String Handling Package (PUSHUP)," Purdue University Computing Center, December 1970.

*Shapiro, Michael D., "SNOBOL4 (Version 3) External Functions, System Programmer Notes," Purdue University Computing Center, December 11, 1970.

Shea, Dorothy J., "CTSS SNOBOL User's Manual," Project MAC, MIT, May 19, 1966.

Simon, A. H. and Walters, D. A., "Dynamic Storage Allocation in RCA SNOBOL and Other Programming Languages" RCA Laboratories, Princeton, New Jersey, April 1965.

Simon, A. H., and Walters, D. A. "RCA SNOBOL Programmers Manual," RCA Laboratories, Princeton, New Jersey, December 1964.

Special Interest Group on Programming Languages (SIGPLAN) of the Association for Computing Machinery, "SNOBOL Bulletin," Published aperiodically in the SIGPLAN Notices.

Stone, R. A., "SNOBOL4 Primitives," Western Electric Company, 1970 (unpublished).

Sturgeon, R., "Interactive SNOBOL4 System for the SDS 940," Project Genie Document R-34, University of California, Berkeley, December 1968.

*Sturgeon, Roger, "INTERACTIVE SNOBOL4 FOR THE SDS 940," University of California, Berkeley, Contract SD-185, December 1968

Swartz, F. G., "Introduction to MTS SNOBOL4," University of Michigan (March 1969).

Tabler, O. F., and others, "The SNOBOL4 Compiler-Interpreter as Currently Implemented for the IBM System/360 Operating System: A First Analysis," Prepared by Logicon for the Bunker-Ramo Corporation, Defense Systems Division, Report CS-6816-R0117, April 12, 1968, Revised May 13, 1968.

*Towell, Alfred I., "How to Use SNOBOLS for the 6600," RANDOM BITS, Indiana University Computing Center, 6(5), January 1971.

Trotter, H. F., "Introduction to SNOBOL, Part I," Princeton University Computer Center, Programming Note Number 31, March 1966.

Wade, Larry, "SNOBOL-10 -- SNOBOL4 for the PDP-10," Digital Equipment Corporation, Palo Alto, California.

* New or revised entry

*Wade, Larry, "SNOBOL4 Version 3.4," DECUS Program Library, DECUS No. 10-104, Digital Equipment Corporation, Maynard Massachusetts, November 24, 1970.

Wegner, Peter, "The Structure of SNOBOL4," Cornell University, Department of Computer Sciences, Technical Report 68-9, February 1968.

Wegner, Peter, "Concepts and Structures in Programming Languages," Cornell University, Department of Computer Sciences, Technical Report 68-14, April 1968 (26-29).

Wegner, Peter, Programming Languages, Information Structures and Machine Organization, New York: McGraw-Hill, 1968.

Whitney, Gordon E., "A Grammar for the Specification of Dynamic Declarations Applied to SNOBOL4," Western Electric Research Center, Princeton, NJ (no date).

Yale University Computer Center, "SNOBOL," Memorandum 45L, January 26, 1965.

Yale University Computer Center, "SNOBOL, A String Manipulation Language," Memorandum 45, May 1964.

This page left blank intentionally.

Part 2: Uses of SNOBOL

*Applied Logic Corporation, "AL/COM SNOBOL-D (PRELIMINARY) USER'S MANUAL," Applied Logic Corporation, Princeton, N.J., July 1970.

Bailey, F. N., Brann, J., and Kain, R. T., "Algebra I Users Reference Manual," University of Minnesota, Electrical Engineering Department, August 10, 1969.

Block, Ronald A., "Automatic Patching of an Analog Computer Under the Control of a Digital Computer," Master's thesis, University of Illinois, 1968.

Chai, David T., Peng, F. C. C., and others, "Chinese-English Machine Translation," The Bunker-Ramo Corporation, July 28, 1967, Final Report for Contract AF30(602)-3993.

Chai, David T., and DiGiuseppe, Jack, "A Study of System Design Considerations in Computers for Symbol Manipulation," University of Michigan, Dept. of Electrical Engineering, Final Report for NSF Grant GP 786, Report 05635-1-F, May 1965.

Chambers, John M., "On Methods of Asymptotic Approximation for Multivariate Distributions," Biometrika 54(3 & 4) December 1967 (367-383).

Chen, Wai-Kai, "Iterative Procedure for Generating Trees and Directed Graphs," Electronics Letters 4(23) November 15, 1968 (516-518).

Conte, Samuel D., Kehl, William B., and Kraft, Donald H., "The Practical Problems: What Does the Humanist Have to Know and Do to Use the Computer," Computers in Humanistic Research, Edmund A. Bowles, editor, Englewood Cliffs, N. J.: Prentice-Hall, 1967.

*Davis, Randy, "SNOBOL4 CROSS-REFERENCE TABLE GENERATOR," Stanford Linear Accelerator Center, CGTM 118, January 1971.

Ducretet, Pierre R., "Frequency Dictionary and Verbal Index of Montaigne's 'Essais'," Computers and the Humanities 4(2) Nov. 1969, Report L305 (130).

*Duquet, Robert T., "ELIZA in SNOBOL4," Department of Computer Science, State University of New York at Albany, November 25, 1970.

- *Durham, Stanton P. and Rogers, David Ellis, "An Application of Computer Programming to the Reconstruction of a Proto-Language," Internationaler Technischer Literaturanzeiger (ITL), 5, 1969, (70-81).
- Erickson, Raymond F., "Music Analysis and the Computer: A Report," Computers and the Humanities 3(2) November 1968 (87-104).
- *Feldman, M. B., "Toward a Programming System for Automata Studies," University of Pennsylvania, Technical Report GRANT FP-5661, May 1970.
- Fiore, Mary E., "Harmonic Structure in Variation Movements by Anton von Webern," Computers and the Humanities 1(5) May 1967, Report M15 (225), (Revision) Computers and the Humanities 2(5) May 1968, Report M15 (243).
- *Fiore, Mary E., "Webern's Use of Motive in the Piano Variations," The Computer and Music, edited by Harry B. Lincoln, Cornell University Press 1970, (115-122).
- Forte, Allen, "The Structure of Music," Computers and the Humanities 1(5) May 1967, Report M13 (224).
- Forte, Allen, "A Program for the Analytic Reading of Scores," Music Theory, 10(2), Winter 1966 (330-363).
- *Fuller, Ramon, "Toward a Theory of Webernian Harmony, via Analysis with a Digital Computer," The Computer and Music, edited by Harry B. Lincoln, Cornell University Press 1970, (123-131).
- Gammon, E. R., "A Segmentation Procedure," Information Retrieval Note Number 47, December 3, 1964.
- Grant, Charles A., "An Interactive Command Generating Facility," Communications of the ACM 13(7) July 1970 (403-406).
- Greenberg, Nathan A., "Breakdown," Computers and the Humanities 2(5) May 1968, Report L99 (228).
- Guzman, Adolfo, and McIntosh, Harold V., "Comments on 'All Paths Through a Maze'," Proceedings of the IEEE 55(8) August 1967 (1525-1526).
- Holbrow, Charles, "CONC: A SNOBOL4 Program for Generating Selective Concordance for Short Texts," Stanford Linear Accelerator Center, CGTM 91, June 1970.

- Holbrow, Charles, "FONC: A Faster Concordance Generator Which Reads Input Text from Tape," Stanford Linear Accelerator Center, CGTM 95, July 1970.
- Holden, A., "The Simulation of Human Methods of Solving Mathematical Problems Using Digital Computer Programs with the Capacity to Learn," University of Washington, Department of Electrical Engineering, PhD Thesis, October 1964.
- *Hultberg, W. Earle, "Transcription of a Tablature to Standard Notation," The Computer and Music, edited by Harry B. Lincoln, Cornell University Press 1970, (288-295).
- Johnson, D., "A Problem-Solving Machine with the Capacity to Learn from Its Experience," Simulation 3(2) August 1964 (71-76).
- Kagan, Alan L., "Computerized Analysis and Classification of Asian Musical Style," Computers and the Humanities 4(5) May 1970, Report M56 (347).
- *Kaye, Jonathan and Roosen-Runge, Peter, "On-Line Ojibwa Phonological System (OOPS)," Computers and the Humanities, Directory of Scholars Active, 5(2) November 1970.
- King, T. J., "A Computer Analysis of Compositional Practice," Computers and the Humanities 1(2) November 1966 (49-50).
- Kroft, David, "All Paths Through a Maze," Proceedings of the IEEE 55(1) January 1967 (88-90).
- Kroft, David, Reply to Comments on "All Paths Through a Maze," Proceedings of the IEEE 55(8) August 1967 (1526-1527).
- Kuno, Susumu, "Research in Computational Linguistics," Computers and the Humanities 3(5) May 1969, Report L291 (308).
- Kuno, Susumu, "The Augmented Predictive Analyzer for Context-free Languages -- Its Relative Efficiency," Communications of the ACM 9(11) November 1966 (810-823).
- Kuo, F. F., "Network Analysis by Digital Computer," Proceedings of the IEEE 54(6) June 1966 (820-829).
- Lehr, M. D., "Canon One," Computers and the Humanities 2(5) May 1968, Report M34 (244).
- *Lincoln, Harry B., "TEACHING COMPUTER APPLICATIONS IN MUSICOLOGY," IBM Symposium on Introducing the Computer into the Humanities, G320-1044-0 Poughkeepsie, New York, June 30-July 2 1969.

- *Lincoln, Harry B., "The Current State of Music Research and the Computer," Computers and the Humanities, 5(1) September 1970.
- *Lubin, John Francis, "THE ROLE OF THE UNIVERSITY COMPUTER CENTER (INTRODUCING THE COMPUTER INTO THE HUMANITIES)," IBM Symposium on Introducing the Computer into the Humanities, G320-1044-0, Poughkeepsie, New York, June 30-July 2 1969.
- Madnick, Stuart E., "SPL/I: A String Processing Language," IBM Cambridge Scientific Center, (Work performed as undergraduate at MIT), Report 36.006, June 1966.
- Madnick, Stuart E., "String Processing Techniques," Communications of the ACM 10(7) July 1967 (420-424).
- McGowan, Clement, and Wegner, Peter, "A LISP Interpreter in SNOBOL4," SIGPLAN Notices 4(6) June 1969 (20-34).
- Milic, Louis T., "Syntax and Mental Disorder: A Computer Study of the Influence of Personality on Literary Style," Computers and the Humanities 1(5) May 1967, Report L76 (206).
- Milic, Louis T., "Complexity in English Writing," Computers and the Humanities 3(5) May 1969, Report L268 (302).
- Milic, Louis T., and Haskel, Peggy I., "Collocations as a Measure of Stylistic Variety," Computers and the Humanities 4(2) Nov. 1969, Report L302 (129).
- Milic, Louis T., and Hiatt, Mary P., "Parallelism in Present-day American Prose," Computers and the Humanities 4(2) Nov. 1969, Report L303 (129).
- *Milic, Louis T., "A COURSE FOR COMPUTERS FOR HUMANISTIC STUDIES," IBM Symposium on Introducing the Computer into the Humanities G320-1044-0, Poughkeepsie, New York, June 30-July 2 1969.
- *Milic, Louis T. "Computer Poetry Generation," Computers and the Humanities, Directory of Scholars Active, 5(2) November 1970.
- *Milic, Louis T., "THE 'RETURNER' POETRY PROGRAM," Department of English, The Cleveland State University, Cleveland, Ohio, 1971.
- Preston, Michael J., Wright, Constance, and Schuchter, J. D., "A Concordance to the Songs of the Tudor Court," Computers and the Humanities 4(2) Nov. 1969, Report L327 (137).

Preston, Michael J. and Baker, D. C., "A Computer-Assisted Study of the British Folk Play," Computers and the Humanities 4(2) Nov. 1969, Report L328 (137).

Preston, Michael J., Viderga, G. I., Wright, Constance, and Baker, D. C., "A Concordance to the Shorter Middle English Poem," Computers and the Humanities 4(2) Nov. 1969, Report L329 (137).

*Reidel, Johannes and Boody, Charles G., "A Computer-Assisted Study of Ecuadorian Urban Music," Computers and the Humanities, Directory of Scholars Active 5(2) November 1970.

Reitman, Judith S., "Computer Simulation of an Information Processing Model of Short Term Memory," Models of Memory, D. A. Norman, editor, New York: Academic Press, 1969.

Roosen-Runge, Peter H., "A Table of Bell Polynomials: Y1 to Y16," University of Michigan, Mental Health Research Institute, Communication 212, August 1967.

Sarna, David E. Y., and others, "Structural Analysis of Hebrew Roots," Computers and the Humanities 2(5) May 1968, Report L187 (237).

*Sarna, David E. and Schiffman, Laurence H., "Computer-Aided Manuscript Study," Computers and the Humanities, Directory of Scholars Active, 5(2) November 1970.

*Shakespeare, Dennis K., "Inverse Catalan Dictionary," Computers and the Humanities, Directory of Scholars Active, 5(2) November 1970.

Smith, Raoul N., "A Test of Rule Ordering for Phonological Changes in Russian from PIE to the Present," Computers and the Humanities 3(5) May 1969, Report L294 (309).

Shapiro, Michael D., "SNOTEXT: A Program for Text Preparation," Purdue University Computer Science Department, May 20, 1968.

Tedgard, Henry F., "A Formal System for Defining the Syntax and Semantics of Computer Languages," Massachusetts Institute of Technology, Project MAC, Report MAC-TR-60, April 1969.

*Teitel, Robert F., "COMPUTING AT A LIBERAL ARTS INSTITUTION," IBM Symposium on Introducing the Computer Into the Humanities, G320-1044-0, Poughkeepsie, New York, June 30-July 2 1969.

Turner, Ronald C., "Rhythmical Ordering as an Index to Literary Style," Computers and the Humanities 4(2) Nov. 1969, Report L299 (128).

*Wachal, Robert S., "The Machine in the Garden: Computers and Literary Scholarship, 1970," Computers and the Humanities 5(1) September 1970.

Wells, John C., and Starck, Taylor, "Dictionary of the Old High German Glosses," Computers and the Humanities 3(5) May 1969, Report L19 (283).

Catalog of Bell Laboratories SNOBOL Publications

Prepared by Michael D. Shapiro

Bell Laboratories, Incorporated
Holmdel, New Jersey

July 31, 1970

S4D21

7/70 RO-S4D21

This page left blank intentionally.

Introduction

Throughout the implementation efforts on SNOBOL, numerous documents have been prepared at Bell Laboratories: descriptions of the language and special features, information for persons installing the language on the IBM System/360 or implementing it on other machines, and publications on similar related subjects. This document is a catalog for these documents. It is intended to provide sufficient information for either the collector who wants to know if he has everything or for the user who wants to know if he has the latest (or most applicable) version of a document.

All SNOBOL4 documents are assigned a reference designation beginning with S4. The S4D items are general documentation about the language. Publications on relevant topics are also included in this section. The S4N entries which follow are installation notes. Some S4N documents relate specifically to installations running SNOBOL4 on an IBM System/360 operating under OS and others refer to the machine-independent macro-implementation.

A mailing list of interested users has been established and occasional SNOBOL4 information bulletins have been distributed. These bulletins are designated S4B and are catalogued in the third section.

Documents describing versions of SNOBOL prior to SNOBOL4 are listed in the final part. These describe various features of the language. In general these are obsolete, and few copies are available.

In addition to these documents, the SNOBOL4 user is referred to the book:

"The SNOBOL4 Programming Language" (2nd edition)
R. E. Griswold, J. F. Poage, and I. P. Polonsky
Prentice-Hall, 1971.

Other outside publications about SNOBOL are listed in "A Bibliography of SNOBOL Publications" (Document S4D12).

This page left blank intentionally.

Document Number	Date	Title	Applicable Version(s)	Status
S4D1	Undated	Preliminary description of the SNOBOL4 programming language	Preliminary	Superseded by S4D1a
S4D1a	7/6/67	Preliminary description of the SNOBOL4 programming language	Preliminary	Superseded by S4D4
S4D2	8/10/67	SNOBOL4 language features: input and output	Preliminary	Superseded by S4D2a
S4D2a	8/10/67	SNOBOL4 language features: input and output	Preliminary	Superseded by S4D4
S4D3	Undated	Keywords	Preliminary	Superseded by S4D4
S4D4	10/20/67	Preliminary report on the SNOBOL4 programming language - II	Preliminary	Superseded by S4D4a
S4D4a	11/22/67	Preliminary report on the SNOBOL4 programming language - II	Preliminary	Superseded by S4D4b
S4D4b	3/10/68	Preliminary report on the SNOBOL4 programming language - II	Preliminary	Superseded by S4D9
S4D5	Undated	JCL for SNOBOL4	Preliminary	Superseded by S4D5a
S4D5a	Undated	JCL for SNOBOL4	Preliminary	Superseded by S4D5b
S4D5b	Undated	Version 1 of SNOBOL4	1	Superseded by S4D5c
S4D5c	Undated	Version 1 of SNOBOL4	1	Superseded by S4D5d
S4D5d	Undated	Version 1 of SNOBOL4	1	Superseded by S4D5e
S4D5e	Undated	Version 2.0 of SNOBOL4	2.0	Superseded by S4D5f
S4D5f	Undated	Version 3 of SNOBOL4	3	Current
S4D6		SNOBOL4 functions and operations	Preliminary	Superseded by S4D9 before completion
S4D7	Undated	Tracing facilities for SNOBOL4	Preliminary	Superseded by S4D7a
S4D7a	3/27/68	Tracing facilities for SNOBOL4	Preliminary	Superseded by S4D9
S4D8	12/27/68	A guide to the macro implementation of SNOBOL4	2.0 (E22)	Superseded by S4D8 dated 5/7/69
S4D8	5/7/69	A guide to the macro implementation of SNOBOL4	2.0 (E22)	Superseded by S4D8a
S4D8a	7/28/69	A guide to the macro implementation of SNOBOL4	2.0 (E22)	Superseded by S4D8b
S4D8b	12/15/69	A guide to the macro implementation of SNOBOL4	3.0 (E32)	Superseded by S4D8c
S4D8c	2/27/70	A guide to the macro implementation of SNOBOL4	3.0 (E32)	Current
S4D9	8/6/68	The SNOBOL4 programming language	1.3	Superseded by S4D9a
S4D9a	8/6/68	The SNOBOL4 programming language	1.3	Superseded by Prentice-Hall book
S4D10	10/10/68	External functions for SNOBOL4	1.3	Superseded by S4D10a
S4D10a	5/5/69	External functions for SNOBOL4	2.0	Superseded by S4D10b
S4D10b	12/1/69	External functions for SNOBOL4	3.0	Superseded by S4D10c
S4D10c	7/31/70	External functions for SNOBOL4	3.0	Current
S4D11	7/1/68	Blocks - a proposed new datatype for programming languages	Preliminary	Superseded by S4D11a
S4D11a	5/1/69	A user manual for BLOCKS	BLOCKS 1.1	Superseded by S4D11b
S4D11b	4/1/70	A user manual for BLOCKS	BLOCKS 1.2	Superseded by S4D11b dated 6/1/70
S4D11b	6/1/70	A user manual for BLOCKS	BLOCKS 1.4	Current
S4D12	8/1/69	A bibliography of SNOBOL publications		Superseded by S4D12a

Document Number	Date	Title	Applicable Version(s)	Status
S4D12a	8/1/70	A bibliography of SNOBOL publications		Current
S4D13	8/11/69	A guide to the implementation of SNOBOL4B	All BLOCKS	Current
S4D14	11/15/69	Version 3 of SNOBOL4	3	Superseded by S4D14a
S4D14a	1/2/70	Version 3 of SNOBOL4	3	Superseded by S4D14b
S4D14b	3/3/70	Version 3 of SNOBOL4	3	Current
S4D15	1/1/70	Macro usage tables for SNOBOL4	3.0	Superseded by S4D15a
S4D15a	1/23/70	Macro usage tables for SNOBOL4	3.0	Superseded by S4D15b
S4D15b	8/5/70	Macro usage tables for SNOBOL4	3.0	Current
S4D16	7/13/70	SNOBOL4 address list	3	Current
S4D17	7/15/70	SNOBOL4 macro source for version 3.0	3.0	Current
S4D18	7/31/70	Catalog of Bell Laboratories SNOBOL4 publications		Current
S4D19	7/16/70	IBM 360 subroutines for version 3.0 of SNOBOL4	3.0	Current
S4D20	7/17/70	IBM 360 macro definitions for version 3.0 of SNOBOL4	3.0	Current
S4D21	7/31/70	Catalog of Bell Laboratories SNOBOL publications		Current
S4D22	8/11/70	Procedure structure cross-reference tables for SNOBOL4	3.0	Current

Document Number	Date	Title and contents	Applicable Version(s)	Status
S4N1	6/2/69	Notice to installations running version 2.0 of SNOBOL4 on the IBM System/360 Correction for release 17 MVT Task timing	2.0	Superseded by S4N1a
S4N1a	6/2/69	Notice to installations running version 2.0 of SNOBOL4 on the IBM System/360 Correction for release 17 MVT Task timing	2.0	Obsolete
S4N2	6/16/69	Notice to installations running version 2.0 of SNOBOL4 on the IBM System/360 Correction for S4N1	2.0	Obsolete
S4N3	12/15/69	Extended distribution tape for version 3 of SNOBOL4	3.0	Superseded by S4N3a
S4N3a	1/26/70	Extended distribution tape for version 3 of SNOBOL4 Version 3.0 Update to version 3.1 Corrections to MAIN1 and MAIN2	3.1	Superseded by S4N3b
S4N3b	3/10/70	Extended distribution tape for version 3 of SNOBOL4 Version 3.0 Update to version 3.2 Corrections to MAIN1 and MAIN2	3.2	Superseded by S4N3c
S4N3c	6/30/70	Extended distribution tape for version 3 of SNOBOL4 Version 3.0 Update to version 3.3	3.3	Superseded by S4N3d
S4N3d	7/17/70	Extended distribution tape for version 3 of SNOBOL4 Version 3.0 Update to version 3.4	3.4	Current
S4N4	1/23/70	Version 3 of SNOBOL4: Installation information for the IBM 360 operating under OS Introduction Installation of version 3.0 Update to version 3.1 Information for programmers	3.1	Superseded by S4N4a
S4N4a	3/20/70	Version 3 of SNOBOL4: Installation information for the IBM 360 operating under OS Introduction Installation of version 3.0 Update to version 3.2 Information for programmers	3.2	Superseded by S4N4b
S4N4b	6/25/70	Version 3 of SNOBOL4: Installation information for the IBM 360 operating under OS Introduction Installation of version 3.0 Updating to version 3.3 Information for programmers	3.3	Superseded by S4N4c
S4N4c	7/17/70	Version 3 of SNOBOL4: Installation information for the IBM 360 operating under OS Introduction Installation of version 3.0 Updating to version 3.4 Information for programmers	3.4	Current
S4N5	1/30/70	Update to version 3.1 of SNOBOL4 Errors in version 3.0 Updating to version 3.1 Errors in version 3.1	3.1	Superseded by S4N6
S4N6	3/20/70	Updating to version 3.2 of SNOBOL4 Updating to version 3.2 Errors in version 3.1 Errors in version 3.2	3.2	Superseded by S4N6a
S4N6a	6/25/70	Updating to version 3.3 of SNOBOL4 Updating to version 3.3 Version 3.2 errors corrected by version 3.3 Errors in version 3.3	3.3	Superseded by S4N6b

Document Number	Date	Title and contents	Applicable Version(s)	Status
S4N6b	7/13/70	Updating to version 3.4 of SNOBOL4 Updating to version 3.4 Version 3.3 errors corrected by version 3.4 Errors in version 3.4	3.4	Current
S4N7	4/15/70	Source updates to version 3 of SNOBOL4 Introduction Errors corrected in version 3.3	3.3	Superseded by S4N7a
S4N7a	6/15/70	Source updates to version 3 of SNOBOL4 Introduction Errors corrected in version 3.3	3.3	Superseded by S4N7b
S4N7b	7/13/70	SNOBOL4 MAIN1 updates to convert version 3.3 to 3.4 Updating to version 3.4 Errors corrected in version 3.4	3.4	Current
S4N8	7/9/70	Updates to SNOBOL4 source files Description of alter deck format	3	Current
S4N9	7/14/70	SNOBOL4 MAIN1 updates to convert version 3.0 to 3.4 Updating to version 3.4 Errors corrected in version 3.4	3.4	Current
S4N10	7/14/70	SNOBOL4 MAIN2 updates to convert version 3.0 to 3.4 Updating to version 3.4 Errors corrected in version 3.4	3.4	Current
S4N11	7/22/70	Correction to MAIN1 update from version 3.3 to 3.4	3.4	Current

Document Number	Date	Contents
S4B1	8/9/68	Questionnaire Version 1.3 availability Announcement of S4D9
S4B2	3/1/69	Version 2 of SNOBOL4 Publications "The SNOBOL4 Programming Language" "External Functions for SNOBOL4" "The Implementation of SNOBOL4" Implementations of SNOBOL4 Questionnaire results Miscellanea Tapes Mailing list Request for information
S4B2a	3/1/69	(Same as S4B2 except "Miscellanea" replaced by "Request for Information")
S4B2b	3/1/69	(Same as S4B2a, but adding mailing address for Canadian installations)
S4B3	7/15/69	BLOCK data type for SNOBOL4 Publications "The SNOBOL4 Programming Language" "A Guide to the Macro Implementation of SNOBOL4" "External Functions for SNOBOL4" "A User Manual for BLOCKS" Correction to version 2.0 for the IBM 360 Version 3.0 of SNOBOL4 SNOBOL4 subset project Checklist of publications
S4B4	12/15/69	Version 3 of SNOBOL4 Implementations of SNOBOL4 Publications "A Bibliography of SNOBOL Publications" "The SNOBOL4 Programming Language" Questionnaire

- Clark, D. L., "Floating Point Arithmetic in SNOBOL," October 21, 1965.
- Farber, D. J., Griswold, R. E., and Polonsky, I. P., "A Preliminary Report on the String Manipulation Language SNOBOL," May 16, 1963.
- Farber, D. J., Griswold, R. E., and Polonsky, I. P., "SNOBOL, A String Manipulation Language," Journal of the ACM 11(1) January 1964 (21-30).
- Farber, D. J., Griswold, R. E., and Polonsky, I. P., "SNOBOL3," October 13, 1964.
- Farber, D. J., Griswold, R. E., and Polonsky, I. P., "The SNOBOL3 Programming Language," Bell System Technical Journal 45(6) July-August 1966 (895-943).
- Farber, D. J., Griswold, R. E., Manacher, G. K., Polonsky, I. P., and Varian, L. C., "Programming Machine-language Functions for SNOBOL3," May 13, 1965.
- Griswold, R. E., and Polonsky, I. P., "String Pattern Matching in the Programming Language SNOBOL," July 1, 1963.
- Griswold, R. E., and Varian, L. C., "Special Purpose SNOBOL3 Functions," November 24, 1964.
- Griswold, R. E., and Polonsky, I. P., "Tree Functions for SNOBOL3," February 1, 1965.
- Griswold, R. E., "Linked-list Functions for SNOBOL3," June 1, 1965.
- Griswold, R. E., "Special Purpose SNOBOL3 Functions - II," April 18, 1966.
- Manacher, G. K., "A Package of Subroutines for the SNOBOL Language," July 1, 1964.
- Manacher, G. K., and Varian, L. C., "A Dimension Statement and Random Number Facility for the SNOBOL Language," October 23, 1964.