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.

**Update from Version 3.4 to Version 3.6.**
ERROR CORRECTION
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 SNOM1A has not been applied to your update deck, please add it:

*IDENT SNOM1A
*/ CORRECTION TO ERROR IN SNOM1
*DELETE MACRO.773
  BX5   X5-X5
*DELETE SNOM1.33
  +   SA4   X5+B2

RECEIVED
AUG 16 1971
SOFTWARE DISTRIBUTION
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.


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 RO-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 LGO.
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(COMPILE)
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:

<table>
<thead>
<tr>
<th>Descriptor (general format)</th>
</tr>
</thead>
<tbody>
<tr>
<td>59</td>
</tr>
<tr>
<td>Address</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Descriptor (integer value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>59</td>
</tr>
<tr>
<td>Integer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Descriptor (real value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>59</td>
</tr>
<tr>
<td>Real value</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>String specifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>59</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>
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:

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

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:

<table>
<thead>
<tr>
<th>Job card</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SNOBOL(,,SYNTAX)</td>
<td></td>
</tr>
<tr>
<td>REWIND(SYNTAX)</td>
<td></td>
</tr>
<tr>
<td>COMPASS(I=SYNTAX,B=PUNCHB,S=0)</td>
<td></td>
</tr>
<tr>
<td>7/8/9</td>
<td></td>
</tr>
<tr>
<td>SYNGEN program and data</td>
<td></td>
</tr>
<tr>
<td>6/7/8/9</td>
<td></td>
</tr>
</tbody>
</table>

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 QENTRY 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.

**SNOSYS - 5**
The system, when loaded with all the system routines, has a field length (including 5000 words of blank common) of a little less than 56000 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. XLOAD 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 SNOBOL 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)).

(12/70) RO SNOSYSX - 1
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

Charachter 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):

<table>
<thead>
<tr>
<th>Character</th>
<th>CDC 6000 Equivalent</th>
<th>Multipunch Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>colon</td>
<td>:</td>
<td>2-8 punch combination</td>
</tr>
<tr>
<td>semicolon</td>
<td>;</td>
<td>12-7-8</td>
</tr>
<tr>
<td>less</td>
<td>&lt;</td>
<td>12-2-8 or 12-0</td>
</tr>
<tr>
<td>greater</td>
<td>&gt;</td>
<td>11-7-8</td>
</tr>
<tr>
<td>not sign</td>
<td>~</td>
<td>12-6-8</td>
</tr>
</tbody>
</table>

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.
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)

R0 SNOBOL4 - 5
The compiler reads the source program from whichever file is number 1. Thus to compile a program from the disk file COMPILE, the card becomes:

SNOBOL(COMPILE)

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

SNOBOL(COMPILE,,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-or-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:

\[ \text{LC}=\text{count} \]

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

\[ \text{SNOBOL(LC=100000)} \]

An example with a file list is:

\[ \text{SNOBOL(LC=100000,COMPILE,,,INPUT)} \]

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):

\[ 1 - 26 \quad \text{letters A - Z} \]
\[ 27 - 36 \quad \text{digits 0 - 9} \]
\[ 37 - 63 \quad \text{punctuation} \]
\[ 64 \quad \text{"binary 00"} \]

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 sub-programs 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. \( \text{string} \) 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 XTABLE. 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, XTABLE 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:

<table>
<thead>
<tr>
<th>59</th>
<th>18</th>
<th>17</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>address</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>column</th>
<th>11</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>VFD</td>
<td>42/0Lname,18/=Xname</td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>column 2</th>
<th>11</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDENT</td>
<td>XFTABLE</td>
<td></td>
</tr>
<tr>
<td>ENTRY</td>
<td>XFTABLE</td>
<td></td>
</tr>
<tr>
<td>XFTABLE</td>
<td>VFD</td>
<td>42/0LSIN,18/=XSIN</td>
</tr>
<tr>
<td>VFD</td>
<td>42/0LSQRT,18/=XSQRT</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>42/0LABCXYZ,18/=XABCXYZ</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

SNOBOLX - 6
B. Compile a FORTRAN IV external function and use it in a SNOBOL4 program:

Job card with CM65000 or more
RUN(5)
COMMON(SNOBOLX)
NOREDUCE.
LOAD(LGO)
LOAD(SNOBOLX)
EXECUTE.
7/8/9

FUNCTION ADD3(A,B,C)
...
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:

<table>
<thead>
<tr>
<th>Register</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>STRING</td>
</tr>
<tr>
<td>2</td>
<td>BLOCK (internal)</td>
</tr>
<tr>
<td>3</td>
<td>PATTERN</td>
</tr>
<tr>
<td>4</td>
<td>ARRAY</td>
</tr>
<tr>
<td>5</td>
<td>TABLE</td>
</tr>
<tr>
<td>6</td>
<td>INTEGER</td>
</tr>
<tr>
<td>7</td>
<td>REAL</td>
</tr>
<tr>
<td>8</td>
<td>CODE</td>
</tr>
<tr>
<td>9</td>
<td>NAME</td>
</tr>
<tr>
<td>10</td>
<td>KEYWORD</td>
</tr>
<tr>
<td>11</td>
<td>EXPRESSION</td>
</tr>
<tr>
<td>12</td>
<td>LINK (internal)</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>59</th>
<th>29</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>length</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>59</th>
<th>29</th>
<th>17</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>flag</td>
<td>value</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>59</th>
<th>23</th>
<th>17</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>available for data</td>
<td>must be zero</td>
<td>data type code $\geq 1000$</td>
<td></td>
</tr>
</tbody>
</table>

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


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

* New or revised entry


*Emerson, Philip, "PDP-9 SNOBOL Version A," DECUS No. 9-43a, The Cleveland State University, Cleveland, Ohio, February 1970.


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.


* New or revised entry


Guard, James R., "SNOBOL-6" (for the PDP-6 Computer), Applied Logic Corporation, Princeton, N.J.


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.


* New or revised entry


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

* New or revised entry


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., "SNOBOL3 to SNOBOL4 Conversion Guide," Purdue University Computing Center, October 1970.

*Shapiro, Michael D., "SNOBAL-T-SNOBOL4 for PROCESY 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.

* New or revised entry


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


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


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

* New or revised entry


* New or revised entry
This page left blank intentionally.
Part 2: Uses of SNOBOL


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


* New or revised entry


Guzman, Adolfo, and McIntosh, Harold V., "Comments on 'All Paths Through a Maze'," *Proceedings of the IEEE* 55(8) August 1967 (1525-1526).


* New or revised entry


* New or revised entry


* New or revised entry


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


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


* New or revised entry


* New or revised entry
Catalog of Bell Laboratories SNOBOL Publications

Prepared by Michael D. Shapiro

Bell Laboratories, Incorporated
Holmdel, New Jersey

July 31, 1970

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

Other outside publications about SNOBOL are listed in "A Bibliography of SNOBOL Publications" (Document S4D12).
This page left blank intentionally.
<table>
<thead>
<tr>
<th>Document Number</th>
<th>Date</th>
<th>Title</th>
<th>Applicable Version(s)</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>S4D1</td>
<td>Undated</td>
<td>Preliminary description of the SNOBOL4 programming language</td>
<td></td>
<td>Preliminary</td>
</tr>
<tr>
<td>S4D1a</td>
<td>7/6/67</td>
<td>Preliminary description of the SNOBOL4 programming language</td>
<td></td>
<td>Superseded by S4D1a</td>
</tr>
<tr>
<td>S4D2</td>
<td>8/10/67</td>
<td>SNOBOL4 language features: input and output</td>
<td></td>
<td>Preliminary</td>
</tr>
<tr>
<td>S4D2a</td>
<td>8/10/67</td>
<td>SNOBOL4 language features: input and output</td>
<td></td>
<td>Superseded by S4D4</td>
</tr>
<tr>
<td>S4D3</td>
<td>Undated</td>
<td>Keywords</td>
<td></td>
<td>Preliminary</td>
</tr>
<tr>
<td>S4D4</td>
<td>10/20/67</td>
<td>Preliminary report on the SNOBOL4 programming language - II</td>
<td></td>
<td>Superseded by S4D4a</td>
</tr>
<tr>
<td>S4D4a</td>
<td>11/22/67</td>
<td>Preliminary report on the SNOBOL4 programming language - II</td>
<td></td>
<td>Superseded by S4D4b</td>
</tr>
<tr>
<td>S4D4b</td>
<td>3/10/68</td>
<td>Preliminary report on the SNOBOL4 programming language - II</td>
<td></td>
<td>Preliminary</td>
</tr>
<tr>
<td>S4D5</td>
<td>Undated</td>
<td>JCL for SNOBOL4</td>
<td></td>
<td>Superseded by S4D5a</td>
</tr>
<tr>
<td>S4D5a</td>
<td>Undated</td>
<td>JCL for SNOBOL4</td>
<td></td>
<td>Superseded by S4D5b</td>
</tr>
<tr>
<td>S4D5b</td>
<td>Undated</td>
<td>Version 1 of SNOBOL4</td>
<td>1</td>
<td>Superseded by S4D5c</td>
</tr>
<tr>
<td>S4D5c</td>
<td>Undated</td>
<td>Version 1 of SNOBOL4</td>
<td>1</td>
<td>Superseded by S4D5d</td>
</tr>
<tr>
<td>S4D5d</td>
<td>Undated</td>
<td>Version 1 of SNOBOL4</td>
<td>1</td>
<td>Superseded by S4D5e</td>
</tr>
<tr>
<td>S4D5e</td>
<td>Undated</td>
<td>Version 2.0 of SNOBOL4</td>
<td>2.0</td>
<td>Superseded by S4D5t</td>
</tr>
<tr>
<td>S4D5f</td>
<td>Undated</td>
<td>Version 3 of SNOBOL4</td>
<td>3</td>
<td>Current</td>
</tr>
<tr>
<td>S4D6</td>
<td>Undated</td>
<td>SNOBOL4 functions and operations</td>
<td></td>
<td>Superseded by S4D9 before completion</td>
</tr>
<tr>
<td>S4D7</td>
<td>Undated</td>
<td>Tracing facilities for SNOBOL4</td>
<td></td>
<td>Superseded by S4D7a</td>
</tr>
<tr>
<td>S4D7a</td>
<td>3/27/68</td>
<td>Tracing facilities for SNOBOL4</td>
<td></td>
<td>Superseded by S4D9</td>
</tr>
<tr>
<td>S4D8</td>
<td>12/27/68</td>
<td>A guide to the macro implementation of SNOBOL4</td>
<td>2.0 (E22)</td>
<td>Superseded by S4D8 dated 5/7/69</td>
</tr>
<tr>
<td>S4D8a</td>
<td>5/7/69</td>
<td>A guide to the macro implementation of SNOBOL4</td>
<td>2.0 (E22)</td>
<td>Superseded by S4D8a</td>
</tr>
<tr>
<td>S4D8b</td>
<td>7/28/69</td>
<td>A guide to the macro implementation of SNOBOL4</td>
<td>2.0 (E22)</td>
<td>Superseded by S4D8b</td>
</tr>
<tr>
<td>S4D8c</td>
<td>12/15/69</td>
<td>A guide to the macro implementation of SNOBOL4</td>
<td>3.0 (E32)</td>
<td>Superseded by S4D8c</td>
</tr>
<tr>
<td>S4D9</td>
<td>8/6/68</td>
<td>The SNOBOL4 programming language</td>
<td>1.3</td>
<td>Superseded by S4D9a</td>
</tr>
<tr>
<td>S4D9a</td>
<td>8/6/68</td>
<td>The SNOBOL4 programming language</td>
<td>1.3</td>
<td>Superseded by Prentice-Hall book</td>
</tr>
<tr>
<td>S4D10</td>
<td>10/10/68</td>
<td>External functions for SNOBOL4</td>
<td>1.3</td>
<td>Superseded by S4D10a</td>
</tr>
<tr>
<td>S4D10a</td>
<td>5/5/69</td>
<td>External functions for SNOBOL4</td>
<td>2.0</td>
<td>Superseded by S4D10b</td>
</tr>
<tr>
<td>S4D10b</td>
<td>12/1/69</td>
<td>External functions for SNOBOL4</td>
<td>3.0</td>
<td>Superseded by S4D10c</td>
</tr>
<tr>
<td>S4D10c</td>
<td>7/31/70</td>
<td>External functions for SNOBOL4</td>
<td>3.0</td>
<td>Current</td>
</tr>
<tr>
<td>S4D11</td>
<td>7/1/68</td>
<td>Blocks - a proposed new datatype for programming languages</td>
<td></td>
<td>Preliminary</td>
</tr>
<tr>
<td>S4D11a</td>
<td>5/1/69</td>
<td>A user manual for BLOCKS</td>
<td>BLOCKS 1.1</td>
<td>Superseded by S4D11a</td>
</tr>
<tr>
<td>S4D11b</td>
<td>4/1/70</td>
<td>A user manual for BLOCKS</td>
<td>BLOCKS 1.2</td>
<td>Superseded by S4D11b dated 6/1/70</td>
</tr>
<tr>
<td>S4D11c</td>
<td>6/1/70</td>
<td>A user manual for BLOCKS</td>
<td>BLOCKS 1.4</td>
<td>Current</td>
</tr>
<tr>
<td>S4D12</td>
<td>8/1/69</td>
<td>A bibliography of SNOBOL publications</td>
<td></td>
<td>Superseded by S4D12a</td>
</tr>
<tr>
<td>Document Number</td>
<td>Date</td>
<td>Title</td>
<td>Applicable Version(s)</td>
<td>Status</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------</td>
<td>------------------------------------------------</td>
<td>------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>S4D12a</td>
<td>8/1/70</td>
<td>A bibliography of SNOBOL publications</td>
<td></td>
<td>Current</td>
</tr>
<tr>
<td>S4D13</td>
<td>8/11/69</td>
<td>A guide to the implementation of SNOBOL4</td>
<td></td>
<td>Current</td>
</tr>
<tr>
<td>S4D14a</td>
<td>11/15/69</td>
<td>Version 3 of SNOBOL4</td>
<td>All BLOCKS</td>
<td>Current</td>
</tr>
<tr>
<td>S4D14b</td>
<td>1/2/70</td>
<td>Version 3 of SNOBOL4</td>
<td>3</td>
<td>Superseded by S4D14a</td>
</tr>
<tr>
<td>S4D15</td>
<td>3/3/70</td>
<td>Version 3 of SNOBOL4</td>
<td>3</td>
<td>Current</td>
</tr>
<tr>
<td>S4D15a</td>
<td>1/1/70</td>
<td>Macro usage tables for SNOBOL4</td>
<td>3.0</td>
<td>Superseded by S4D15a</td>
</tr>
<tr>
<td>S4D15b</td>
<td>1/23/70</td>
<td>Macro usage tables for SNOBOL4</td>
<td>3.0</td>
<td>Superseded by S4D15b</td>
</tr>
<tr>
<td>S4D16</td>
<td>8/5/70</td>
<td>Macro usage tables for SNOBOL4</td>
<td>3.0</td>
<td>Current</td>
</tr>
<tr>
<td>S4D17</td>
<td>7/13/70</td>
<td>SNOBOL4 address list</td>
<td>3</td>
<td>Current</td>
</tr>
<tr>
<td>S4D18</td>
<td>7/15/70</td>
<td>SNOBOL4 macro source for version 3.0</td>
<td>3.0</td>
<td>Current</td>
</tr>
<tr>
<td>S4D19</td>
<td>7/31/70</td>
<td>Catalog of Bell Laboratories SNOBOL4 publications</td>
<td></td>
<td>Current</td>
</tr>
<tr>
<td>S4D20</td>
<td>7/17/70</td>
<td>IBM 360 subroutines for version 3.0 or SNOBOL4</td>
<td>3.0</td>
<td>Current</td>
</tr>
<tr>
<td>S4D21</td>
<td>7/17/70</td>
<td>IBM 360 macro definitions for version 3.0 of SNOBOL4</td>
<td>3.0</td>
<td>Current</td>
</tr>
<tr>
<td>S4D22</td>
<td>7/31/70</td>
<td>Catalog of Bell Laboratories SNOBOL publications</td>
<td></td>
<td>Current</td>
</tr>
<tr>
<td></td>
<td>8/11/70</td>
<td>Procedure structure cross-reference tables for SNOBOL4</td>
<td>3.0</td>
<td>Current</td>
</tr>
<tr>
<td>Document Number</td>
<td>Date</td>
<td>Title and contents</td>
<td>Applicable Version(s)</td>
<td>Status</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>-----------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>S4N1</td>
<td>6/2/69</td>
<td>Notice to installations running version 2.0 of SNOBOL4 on the IBM System/360</td>
<td>2.0</td>
<td>Superseded by S4N1a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Correction for release 17 MVT Task timing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4N1a</td>
<td>6/2/69</td>
<td>Notice to installations running version 2.0 of SNOBOL4 on the IBM System/360</td>
<td>2.0</td>
<td>Obsolete</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Correction for release 17 MVT Task timing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4N2</td>
<td>6/16/69</td>
<td>Notice to installations running version 2.0 of SNOBOL4 on the IBM System/360</td>
<td>2.0</td>
<td>Obsolete</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Correction for S4N1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4N3</td>
<td>12/15/69</td>
<td>Extended distribution tape for version 3 of SNOBOL4</td>
<td>3.0</td>
<td>Superseded by S4N3a</td>
</tr>
<tr>
<td>S4N3a</td>
<td>1/26/70</td>
<td>Extended distribution tape for version 3 of SNOBOL4</td>
<td>3.1</td>
<td>Superseded by S4N3b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update to version 3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Corrections to MAIN1 and MAIN2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4N3b</td>
<td>3/10/70</td>
<td>Extended distribution tape for version 3 of SNOBOL4</td>
<td>3.2</td>
<td>Superseded by S4N3c</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Version 3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update to version 3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Corrections to MAIN1 and MAIN2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4N3c</td>
<td>6/30/70</td>
<td>Extended distribution tape for version 3 of SNOBOL4</td>
<td>3.3</td>
<td>Superseded by S4N3d</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Version 3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update to version 3.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4N3d</td>
<td>7/17/70</td>
<td>Extended distribution tape for version 3 of SNOBOL4</td>
<td>3.4</td>
<td>Current</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Version 3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update to version 3.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4N4</td>
<td>1/23/70</td>
<td>Version 3 of SNOBOL4: Installation information for the IBM 360 operating under OS</td>
<td>3.1</td>
<td>Superseded by S4N4a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Introduction</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Installation of version 3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update to version 3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Information for programmers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4N4a</td>
<td>3/20/70</td>
<td>Version 3 of SNOBOL4: Installation information for the IBM 360 operating under OS</td>
<td>3.2</td>
<td>Superseded by S4N4b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Introduction</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Installation of version 3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update to version 3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Information for programmers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4N4b</td>
<td>6/25/70</td>
<td>Version 3 of SNOBOL4: Installation information for the IBM 360 operating under OS</td>
<td>3.3</td>
<td>Superseded by S4N4c</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Introduction</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Installation of version 3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updating to version 3.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Information for programmers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4N4c</td>
<td>7/17/70</td>
<td>Version 3 of SNOBOL4: Installation information for the IBM 360 operating under OS</td>
<td>3.4</td>
<td>Current</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Introduction</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Installation of version 3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updating to version 3.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Information for programmers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4N5</td>
<td>1/30/70</td>
<td>Update to version 3.1 of SNOBOL4</td>
<td>3.1</td>
<td>Superseded by S4N5a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Errors in version 3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updating to version 3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Errors in version 3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4N6</td>
<td>3/20/70</td>
<td>Updating to version 3.2 of SNOBOL4</td>
<td>3.2</td>
<td>Superseded by S4N6a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updating to version 3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Errors in version 3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Errors in version 3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4N6a</td>
<td>6/25/70</td>
<td>Updating to version 3.3 of SNOBOL4</td>
<td>3.3</td>
<td>Superseded by S4N6a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updating to version 3.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Version 3.2 errors corrected by version 3.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Errors in version 3.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Number</td>
<td>Date</td>
<td>Title and contents</td>
<td>Applicable Version(s)</td>
<td>Status</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------</td>
<td>------------------------</td>
<td>--------------------</td>
</tr>
</tbody>
</table>
| 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  
<pre><code>                | 3.4                    | Current            |
</code></pre>
<table>
<thead>
<tr>
<th>Document Number</th>
<th>Date</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>S4B1</td>
<td>8/9/68</td>
<td>Questionnaire</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Version 1.3 availability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Announcement of S4D9</td>
</tr>
<tr>
<td>S4B2</td>
<td>3/1/69</td>
<td>Version 2 of SNOBOL4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Publications</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;The SNOBOL4 Programming Language&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;External Functions for SNOBOL4&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;The Implementation of SNOBOL4&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Implementations of SNOBOL4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Questionnaire results</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Miscellanea</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tapes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mailing list</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Request for information</td>
</tr>
<tr>
<td>S4B2a</td>
<td>3/1/69</td>
<td>(Same as S4B2 except &quot;Miscellanea&quot; replaced by &quot;Request for Information&quot;)</td>
</tr>
<tr>
<td>S4B2b</td>
<td>3/1/69</td>
<td>(Same as S4B2a, but adding mailing address for Canadian installations)</td>
</tr>
<tr>
<td>S4B3</td>
<td>7/15/69</td>
<td>BLOCK data type for SNOBOL4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Publications</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;The SNOBOL4 Programming Language&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;A Guide to the Macro Implementation of SNOBOL4&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;External Functions for SNOBOL4&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;A User Manual for BLOCKS&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Correction to version 2.0 for the IBM 360</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Version 3.0 of SNOBOL4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNOBOL4 subset project</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Checklist of publications</td>
</tr>
<tr>
<td>S4B4</td>
<td>12/15/69</td>
<td>Version 3 of SNOBOL4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Implementations of SNOBOL4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Publications</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;A Bibliography of SNOBOL Publications&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;The SNOBOL4 Programming Language&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Questionnaire</td>
</tr>
</tbody>
</table>
Bell Laboratories SNOBOL publications
Documentation for SNOBOL3 and earlier versions
(All items are unpublished unless otherwise indicated.)


Griswold, R. E., "Linked-list Functions for SNOBOL3," June 1, 1965.

