The basic reference manual for our implementation is the Bell Labs Preliminary Report on the SNOBOL4 Programming Language, version of November, 1967. Basic differences: replacement of Keywords by standard procedures having same purposes; elimination of the truth predicates; and a different character set.

<table>
<thead>
<tr>
<th>Bell Labs Character</th>
<th>CAL SNOBOL character</th>
</tr>
</thead>
<tbody>
<tr>
<td>(alteration)</td>
<td>v, or a composite //</td>
</tr>
<tr>
<td>' and &quot; (single, double quotes)</td>
<td>≠, ↑</td>
</tr>
<tr>
<td>:</td>
<td>: or / not followed by blank.</td>
</tr>
<tr>
<td>&lt; and &gt; (array brackets)</td>
<td>{ and }, or composite (/ and /)</td>
</tr>
</tbody>
</table>

The array function has one argument only: ARRAY (P) generates an array according to the prototype P. The prototypes used as arguments for ARRAY, DATA, and DEFINE should not contain embedded blanks.

The association functions are called as follows:

```
INPUT(V, F, N)
OUTPUT(V, F, C)
```

where V is the variable to be associated.

F is the file with which it is to be associated.
N is a unit record length. If records on the file are shorter, blanks will be added. If records are longer, the extra will be lost.
C is a carriage control character, which if not specified as null will be prefixed to every string placed on the file.

The manipulative functions REWIND and ENDGROUP exist. Each is called with one argument, the name of a file to be rewound or to have a logical end of record written on, respectively. (REWIND writes an eor before rewinding output files).
The following standard procedures are currently implemented (note that there are no tracing facilities, nor functions corresponding to the DUMP Keyword):

ALPHABET()
STLIMIT()
STCOUNT()
MAXLENGTH()
DATA
LGT
ENDGROUP
REWIND
DETACH
INPUT
OUTPUT
ARRAY
IDENT
DIFFER
DEFINE
ARBNO
ANCHOR()
TRIM
ANY
NOTANY
EQ
NE
GT
GE
LT
LE
BREAK
SPAN
RTAB
TAB
RPOS
POS
LEN
SIZE

COMPILE is implemented; this is the CODE of the Bell Labs version. Successive statements are separated by semicolon (12-7-8), syntax is the same as the source program.

CONVERT(X) is implemented. It's value is:

1. If X is a string or integer, then the corresponding real number.
2. If X is a real number, then the corresponding string.

The relational predicates (EQ, etc.) and the arithmetic operators +,-,*,/ work if both operands are real.
DATATYPE(X) is implemented, and returns

- INTEGER if X is an integer or a string conforming to integer syntax (including the null string)
- STRING if X is any other string
- PATTERN (value returned by the COMPILE fcn.)
- CODE (value of unary . operator)
- NAME (value returned by the COMPILE fcn.)
- REAL
- <data-type-name>, where <data-type-name> is the name of a programmer-defined data type.

FNCLEVEL() is implemented, and returns the function calling level, 0 at main program.

Assembling SNOBOL produces two overlays, SNOBOL and SNOJOB. (Compass will also produce a third overlay of all zeros, which should be ignored.) SNOJOB contains the run-time error processor, and the SNOBOL overlay expects to find SNOJOB in the library.

The SNOBOL control card may specify any of the following, in any order: I = input file name, L = output file name or 0, E = error overlay file, *F = max fl in octal, *B = buffer size in octal. The defaults are INPUT, OUTPUT, error overlay in the library, 30K fl, 200 buffer. L = 0 will stop the listing unless a compilation error occurs, in which case the listing will start just after the first error (sorry!). If the E parameter is specified without the = filename part, the file SNOJOB is assumed.

SNOBOL will ask for more field length, in steps of 1000$, until the limit, whose default value is 30000$, is reached. This limit can be changed with the *F parameter on the control card.

SNOBOL should load in $12000 and have enough core for a small program. After compilation, the compiler and any unused standard procedures are thrown out.