

CONSOLIDATION PROGRAM (CNSL)

The current version of CNSL is operating with paper tape input/output only.

Since GPL is not complete, the copy of GPL is omitted as described in 4. 6.

F. M. Longstaff  
November 15, 1963

Distribution

F. M. Longstaff  
M. Lucas  
D. Wray  
I. Sharp  
R. Moore  
V. Taylor  
J. McSherry  
T. McDorman  
J. Chapman  
A. Sharp  
D. Smith

## Consolidation Program (CNSL)

### 1. General Description

1.1 This program accepts the output from FORTRAN or AUTOCODER (or ALGOL?) and prepares it for input by GPL.

1.2 The output of the above compilers is assumed to be divided into segments with a leader associated with each. If only one output device is used by the compiler, each leader (in spite of its name) follows the segment which it describes. If two output devices are used, the segments and the leaders may be separated.

Only the leaders are used as data by CNSL although object program segments may be copied.

CNSL prepares a request slip for use by EXECUTIVE and a consolidated leader for use by GPL.

### 2. Structure of Object Program Segments

2.1 All information handled by CNSL is divided into blocks each consisting of up to 80 characters.

The first character of each block defines the type of block.

2.2 CNSL recognizes 8 types of blocks on input. These are:-

First character (octal)	Type
0 (00)	object program binary block
3 (03)	object program title block
4 (04)	object program terminating block

(	(30)	priority and name block
+	(33)	leader title block
,	(34)	leader terminating block
/	(37)	leader cue block
;	(13)	pause block

2.3 An object program segment consists of:-

- 1) An object program title block
- 2) A number of object program binary blocks
- 3) An object program terminating block

2.4 A leader consists of:-

- 1) A leader title block
- 2) A priority and name block (optional)
- 3) A number of leader cue blocks
- 4) A leader terminating block

2.5 Data, after the first character, consists of fields which may be either octal or alphanumeric fields. Other information, following the fields listed below, is not examined by CNSL although it will be copied to output if the whole block is output.

2.6 An object program title block consists of:-

- 1) An octal field giving type of segment
- 2) An alphanumeric field giving the name of the segment

Only one bit of the octal field is used. This is the ninth bit counting from the least significant end and is used to indicate "force in". The segment name is matched against a cue list held by CNSL to determine whether to accept or reject the segment. Acceptance or rejection of a segment may also be controlled by an overall segment copy switch. If the overall copy switch is on, the segment is accepted if:-

- 1) There are no cues in the list
- 2) The name matches a blank cue
- 3) The name does not match any cue and the force in bit is a one

If the title block is accepted, it and the following binary blocks and the terminating block are copied to output without further processing. Otherwise they are ignored.

2.7 Object program binary blocks and terminating blocks are not processed by CDSL. See the description of CPL for details of their content. The terminating block is identical to the loader terminating block except for the first character.

2.8 A loader title block is identical, except for the first character, to an object program title block and is accepted or rejected in the same way except that an overall copy switch is not examined. If the title block is accepted it and the following loader blocks are copied to output and also used as data. Otherwise they are ignored.

2.9 A priority and name block consists of:

- 1) A 4 digit octal number giving priority
- 2) An alphanumeric field giving program name

Either field may be omitted (or both for that matter) and replaced by spaces.

If a priority, other than zero, is given and if no priority has been given in any previous accepted block, the priority is used to establish the priority for EXECUTIVE. The octal number is considered as 2 characters. If no priority is given by any block, CDSL sets priority to 50 (octal 500).

If a name is given and if no name has been given in any previously accepted block, the first 4 characters of the name are used as the name for EXECUTIVE. If no name is given in any block, CDSL sets the name to XCOB (unknown compiler object program).

The priority and name block is optional and need not appear in a leader. Furthermore it may appear anywhere among the cue blocks.

2.10 A leader cue block consists of:-

- 1) An octal field giving type and value
- 2) An alphanumeric field giving the name of the cue

The octal field is treated as a 24 bit word interpreted as follows:

Bit 0 = 1 if the cue represents a storage area into which data must be loaded

= 0 otherwise

Bit 1 = 1 if the cue represents a storage area which must lie below 4096.

= 0 otherwise

Bit 2 = spare

Bits 3, 4 and 5 give the type of cue as follows:-

- 0 - blank cue (the 16 octal field must be zero)
- 1 - program cue
- 2 - entry point cue
- 3 - area cue (for common areas)
- 4 - peripheral cue
- 5-7 - spare

Bits 6, 7 and 8 - Spare

Bits 9 to 23 - Value

Cues in accepted leaders are consolidated with the stored cue list as described in section 3.

2.11 A leader terminating block consists of ten octal fields giving the private space requirements of the segment in each of ten categories. These are added to ten totals held by CNSL.

After the leader terminating block is processed, the cue list is examined to determine if any blank cues remain. If none remain, the "END" procedure is entered, otherwise reading continues.

2.12 A pause block contains no data, used by CNSL, other than the pause character.

The cue list is scanned and the name of one (the first) blank cue is selected. CNSL then types CNSL NEEDS 'name' on the console typewriter and is suspended.

The operator may then type GO if it is desired to continue reading, or GO at 10 if it is desired to enter the "END" procedure with an incomplete program.

### 3. Consolidation of cues

3.1 Types of cues, as defined by the first 2 octal digits of type and value are as follows:

- 00 - blank cue - entire octal field must be zero
- 41 - program cue - value = length of program segment in words.
- 02 - entry cue - value = distance, in words, of the entry point from the start of the segment
- 03 - area cue for upper, non preset area - value = size of area in words
- 23 - area cue for lower non preset area
- 43 - area cue for upper preset area
- 63 - area cue for lower preset area
- 04 - peripheral cue - last two octal digits of value give peripheral type and unit number respectively.

3.2 Blank cues are used when a segment refers to another segment, entry point or peripheral device without defining such segment, entry or device.

3.3 Program cues are used to define the length of a program segment. Normally one and only one program cue appears in each leader.

3.4 Entry cues are used to define entry points other than the start of the segment.

3.5 Area cues are used to establish areas of storage which may be referred to by more than one segment.

3.6 Peripheral cues are used to associate names with peripheral devices.

3.7 Consolidation of cues is the process of consolidating one read cue into a list of cues held by CNSL. It proceeds as follows:-

- 1) if no cue of a similar name appears in the list, the read cue is added to the list
- otherwise 2) if the read cue is blank it is ignored
- otherwise 3) if the list cue is blank it is replaced by the read cue

- otherwise 4) if the read cue is an area cue a new cue is formed in which the top 9 bits are an 'or' of the two cues and the value is the maximum of the two values
- otherwise 5) the read cue is ignored

#### 4. The 'END' procedure

4.1 The END procedure is entered when the cue list contains no blank cues and therefore describes a complete program. Alternatively END may be entered by operator action with blank cues in the list. If any blank cues remain they are given value = octal 70077 which is likely to give illegal results if any instruction, altered by such a cue, is obeyed.

4.2 CNSL now allocates storage to the complete program in 15 groups, starting at location 30 in the following order:

- 1) Space in category 1 *LOWER WORK*
- 2) Space in category 2 *LOWER VARIABLES*
- 3) Space in category 3
- 4) All lower non preset areas
- 5) Space in category 4 *CONSTANTS*
- 6) Space in category 5 *LITERALS*
- 7) Space in category 6
- 8) All lower preset areas
- 9) All program segments *CATEGORY 11*
- 10) Space in category 7 *UPPER PRESET*
- 11) Space in category 8
- 12) All upper preset areas
- 13) Space in category 9 *UPPER WORK*
- 14) Space in category 10
- 15) All upper non preset areas

4.3 A value of "offset" is calculated.

$$\text{offset} = \text{length of GPL} + \text{length of cue list} - \text{length of groups 1 to 4}$$
$$\text{offset} = 0 \text{ if the above is negative}$$

Offset is used by GPL to load all preset areas above their final location and to move them into position after loading is complete.

4.4 A value of waste space is calculated.

$$\text{waste} = \text{offset} - \text{length of groups 13 to 15}$$

if waste is positive it is added to group 15.

4.5 A release instruction is calculated.

release instruction = 151 order if no peripheral input devices  
of the type required by the loader are required otherwise release  
instruction = 000 0 0

4.6 CNSL now outputs:

- 1) Blank tape in the case of paper tape
- 2) A pause block
- 3) More blank tape
- 4) A request slip for EXECUTIVE
- 5) A copy of GPL (to be read by EXECUTIVE)
- 6) A consolidated leader

4.7 CNSL is now suspended (157 order) ready to deal with another  
program. From this point (or following the initial loading of CNSL)

GO will cause a further program to be consolidated with the overall  
copy switch unchanged (on after initial load)

GO at 11 will turn on the copy switch before consolidating

GO at 12 will turn off the copy switch before consolidating

4.8 The consolidated leader consists of:-

- 1) A relativizer block - first character = = (octal 15)
- 2) A parameter block - first character = > (octal 16)
- 3) A number of consolidated cue blocks - first character  
= ? (octal 17)
- 4) A terminating block - first character = < (octal 14)

The relativizer block contains 10 octal fields giving the initial  
settings of the relativizers for GPL.

The parameter block contains 2 octal fields giving the offset  
and release instruction.

The consolidated cue blocks contain an octal field giving type and  
value and an alphanumeric field giving name. The value of cues, except peripheral  
cues, at this stage represent starting addresses rather than lengths.

The terminating block consists only of the first character.

5. Detailed description of fields

5.1 An octal field

- 1) May have any number of leading spaces
- 2) May then have any number of leading zeros
- 3) May then have from none to eight significant octal digits
- 4) is terminated by any character other than an octal digit - the terminating character is considered to be part of the following field. Space may be used as a terminator only if there has been at least one leading zero or octal digit. End of block will also serve as a terminator.

5.2 An alphanumeric field

- 1) May have any number of leading spaces
- 2) May have from one to forty characters other than space
- 3) Must be terminated by space or end of block

5.3 Octal fields produced by CNSL will have:-

- 1) No leading spaces or zeros
- 2) From one to eight significant digits - single zero if number is zero
- 3) One space as a terminator

5.4 Alphanumeric fields produced by CNSL will have:-

- 1) No leading spaces
- 2) Characters of name
- 3) One space as a terminator