

APPENDIX B

STRUCTURE OF AN OBJECT GRAMMAR

B.1 OVERALL STRUCTURE

The file produced by compiling an grammar or word dictionary is called an object grammar file. This file consists of two parts. The first is a series of object grammar "records", containing the source and object code for the statements in the grammar or dictionary; its structure is described in section B.2. The second part is the symbol table for the grammar or dictionary; its structure is described in section B.4.

B.2 STRUCTURE OF AN OBJECT GRAMMAR RECORD

For each statement (and each block of comments) in the source grammar or word dictionary, the compiler generates a series of lines; these lines will be referred to together as an object grammar record. The first line of each record is a header line, with the following structure:

<u>column(s)</u>	<u>content</u>
1	'@' (indicates that this is a header)
2-21	the name of the record, left justified. The rules for assigning a name to a record are given in section 2.4.1, above. For minor records (substatements and word dictionary subparts) only the minor record name is stored. For idioms, only the first word is stored here.
22-31	blank (to allow for future expansion of record names to 30 characters)
32-35	IDIOM-WORD-COUNT: for word definition records,

- the number of words in the idiom -1 (= 0 if the word is not an idiom or the record is not a word definition record)
- 36-39 SOURCE-LENGTH: the number of lines in the source statement
- 40-43 OBJECT-LENGTH: the number of lines of object code generated
- 44-47 TYPE-OF-STATEMENT:
- | | |
|---|--------------------------------------|
| 0 | comment |
| 1 | BNF definition |
| 2 | restriction |
| 3 | substatement |
| 4 | routine definition |
| 5 | global, attribute, or type list |
| 6 | word definition with category list |
| 7 | word definition subpart |
| 8 | word definition with canonical form |
| 9 | canonical or special form definition |
- 48-49 DERIVATIVE-FLAG: = 1 for derivative word definition records (those marked with a '^'); = 0 for all other records
- 50-55 SYMBOL-TABLE-POINTER:
- for all records except word definitions, the index of the symbol table entry for the head of the record;
 - for word definitions with canonical form, the index of the symbol table entry for the canonical form;
 - for word definitions with explicit category lists, = 0

This header line is followed by:

IDIOM-WORD-COUNT lines, each with a '+' in column 1 and a word of the idiom in columns 2 through 21, left justified (beginning with the second word of the idiom, since the first word appears as part of the header line).

SOURCE-LENGTH lines, each with a '*' in column 1 and a copy of a line of the source statement in columns 2 through 81.

OBJECT-LENGTH lines, each with a '/' in column 1 and a sequence of octal digits (and blanks, which count as 0's) in columns 2 through 81. The object record is encoded as a bit string in a compressed format described in section B.3.

This bit string is then converted to a string of octal digits, which are then divided into lines, with a maximum of 80 digits per line.

B.3 REPRESENTATION OF THE OBJECT STATEMENT

EACH STATEMENT IS COMPILED INTO A LIST STRUCTURE; I.E., A SET OF LIST ELEMENTS, EACH WITH ADDRESS, SPECIAL, AND DECREMENT FIELDS. WHEN THIS LIST STRUCTURE IS WRITTEN OUT AS PART OF AN OBJECT GRAMMAR, IT IS CONVERTED INTO A -COMPACT- LIST STRUCTURE. IN A COMPACT LIST STRUCTURE, DECREMENT POINTERS DO NOT APPEAR EXPLICITLY; INSTEAD, A LIST ELEMENT IS IMMEDIATELY FOLLOWED BY ITS CDR, IF ITS CDR IS NON-NIL THIS IMPLIES THAT THE CDR MUST BE NIL OR A LIST ELEMENT, NOT A SYMBOL, AND THAT TWO ELEMENTS CANNOT HAVE DECREMENT FIELDS POINTING TO THE SAME THIRD LIST ELEMENT. THIS COMPACT STRUCTURE IS USED FOR HISTORICAL REASONS -- ORIGINALLY, COMPACT STRUCTURES WERE USED BOTH INTERNALLY AND EXTERNALLY, BUT MORE RECENTLY THE INTERNAL LIST STRUCTURES WERE CHANGED TO USE AN EXPLICIT DECREMENT POINTER.

EACH COMPACT LIST ELEMENT HAS THREE COMPONENTS: A START-OF-LIST FLAG, A SPECIAL FIELD, AND AN ADDRESS FIELD. THE START-OF-LIST FLAG INDICATES WHETHER THIS LIST ELEMENT IS THE SUCCESSOR IN THE LIST STRUCTURE (I.E., THE CDR) OF THE ELEMENT WHICH PHYSICALLY PRECEDES IT, OR IS THE START OF A NEW LIST (HAS NO PREDECESSOR). A VALUE OF 0 INDICATES THE START OF A NEW LIST, A 1 VALUE A CONTINUATION.

THE SPECIAL AND ADDRESS FIELDS MAY EACH CONTAIN A SYMBOL OR LITERAL REFERENCE, A POINTER TO ANOTHER ELEMENT IN THE LIST STRUCTURE, A NUMBER, OR NIL. EACH FIELD IS REPRESENTED IN THE OBJECT STATEMENT BY A 3-BIT INTERPRETATION FIELD POSSIBLY FOLLOWED BY A 15-BIT DATA FIELD. THE PRESENCE AND SIGNIFICANCE OF THE DATA FIELD IS DETERMINED BY THE VALUE OF THE INTERPRETATION FIELD, AS FOLLOWS:

INTERPRETATION FIELD	DATA FIELD
0	- (NIL FIELD)
1	NUMBER (NOT RELOCATED)
2	ADDRESS OF LIST ELEMENT, RELATIVE TO BEGINNING OF CURRENT RECORD
3	A POINTER TO THE SYMBOL TABLE, INDICATING A REFERENCE TO THE SYMBOL
4	*INVALID INTERPRETATION FIELD*
5	*INVALID INTERPRETATION FIELD*
6	A POINTER TO THE SYMBOL TABLE, INDICATING A LITERAL REFERENCE TO THE NAME OF THE SYMBOL
7	- (LAST INTERPRETATION FIELD IN A RECORD)

THUS A LIST ELEMENT IS REPRESENTED BY A BIT STRING CONTAINING AS FEW AS 7 BITS (A START-OF-LIST FLAG AND TWO NIL FIELDS) OR AS MANY AS 37 BITS (TWO NON-NIL FIELDS). AN ENTIRE LIST STRUCTURE IS REPRESENTED BY CONCATENATING THE REPRESENTATIONS OF ITS LIST ELEMENTS. WORD BOUNDARIES

ARE NOT SIGNIFICANT IN THE OBJECT STATEMENT -- THE STATEMENT IS PROCESSED AS ONE LONG STRING OF BITS.

B.4 THE SYMBOL TABLE IN AN OBJECT GRAMMAR

The second part of an object grammar file is the symbol table. It begins with the line 'SYMBOL TABLE' (starting in column 1). This is followed by a series of lines with the following format:

column(s)	contents
1	blank
2-21	symbol name, left justified
22-31	blank (to allow for future expansion of symbol names to 30 characters)
32-37	SYMBOL TYPE (an integer): bits 0-3 of this number indicate the type of the symbol -- <ul style="list-style-type: none"> 1 BNF definition 2 lexical category 3 generator or operator 4 restriction, substatement, or routine name 5 attribute 6 register 7 type list 10 word dictionary subpart 15 integer
38-43	REFERENCE COUNT (an integer): a count of the number of references to this symbol table entry (both symbol and literal references) in the grammar