APPENDIX D

Procedure Names

This appendix describes how a procedure must be defined in order for MacsBug to recognize it.

Procedure Definition

Whenever possible, MacsBug accepts and returns addresses as procedure names and offsets. MacsBug finds names by scanning relocatable heap blocks for valid procedure definitions. A procedure definition, in the simplest case, consists of a return instruction followed by the procedure's name.

A procedure is defined as follows:

- LINK A6—This instruction is optional; if it is missing, the start of the procedure is assumed to be immediately after the preceding procedure, or at the start of the heap block.
- Procedure code
- RTS or JMP(A0) or RTD
- Procedure name
- Procedure constants

The procedure name can be a fixed length of 8 or 16 bytes, or of variable length. Valid characters for procedure names are a–z, A–Z, 0–9, underscore (_), percent (%), period (.), and space. The space character is allowed only to pad fixed-length names to the maximum length.

With fixed-length format, the first byte is in the range $20 through $7F. The high-order bit may or may not be set. The high-order bit of the second byte is set for 16-character names, clear for 8-character names. Fixed-length 16-character names are used in object Pascal to show class.method names instead of procedure names. The method name is contained in the first 8 bytes and the class name is in the second 8 bytes. MacsBug swaps the order and inserts the period before displaying the name.

With variable-length format, the first byte is in the range $80 to $9F. Stripping the high-order bit produces a length in the range $00 through $1F. If the length is 0, the next byte contains the actual length, in the range $01 through $FF. Data after the name starts on a word boundary. Compilers can place a procedure's constant data immediately after the procedure in memory. The first word after the name specifies how many bytes of constant data are present. If there are no constants, a length of 0 must be given.

Examples of Procedure Definitions

Here are some examples of valid assembly-language procedure definitions:

```
; Variable-length name with no constant data.

Proc1    PROC
         LINK  A6, #0
         UNLK  A6
         RTS
         DC.B  $8C, 'VariableName'
         DC.W  $0000
         ENDP

; Fixed 8-character name.

Proc2    PROC
         LINK  A6, #0
         UNLK  A6
         RTS
         DC.B  $80 + 'F', 'ixed   '
         ENDP

; Fixed 16-character name.

Proc3    PROC
         LINK  A6, #0
         UNLK  A6
         RTS
         DC.B  $80 + 'M',  $80 + 'e', 'thod  Class   '
         ENDP
```