Tech Info Library

Pascal II: Library units (2 of 2)

```
Revised:       11/21/84
Security:      Everyone
```

Pascal II: Library units (2 of 2)

```
======================================================================

   1:  (*$S+*)
   2:  UNIT < name >; INTRINSIC CODE xx DATA yy;
         (* DATA yy is used only if a data segment is
            required. *)

   3:  INTERFACE

   4:    USES < name of unit to be used >;
         CONST < definitions >;
         TYPE < definitions >;
   5:    VAR < definitions >;

   6:    PROCEDURE ONE (I:Integer);
         PROCEDURE TWO (I:Integer);
           (* External procedure *)
         FUNCTION THREE (I:Integer) : Integer;
         FUNCTION FOUR (I:Integer) : Integer;
          (* External function *)

   7:  IMPLEMENTATION

   8:    CONST < definitions >;
         TYPE < definitions >;

   9:    VAR < definitions >;

  10:    PROCEDURE ONE;
           BEGIN
           END;

         PROCEDURE TWO; EXTERNAL;

         FUNCTION THREE;
           BEGIN
           END;
         FUNCTION FOUR; EXTERNAL;

  11:    BEGIN
         (* initialization section *)
         END.
```

DISCUSSION

1: The swapping option is required when compiling ANY unit, regardless of its size. Omitting this option is the most common cause of compiler failures when working with units. The option should be the first line of text, preceding the UNIT header and any other compiler option.

2: The word INTRINSIC is required if the unit is to be intrinsic. DATA is optional, used only if a data segment is necessary (see #5 below). Regular units use only the UNIT < name > portion of this line.

3: INTERFACE is required. It defines the start of this section, which must contain some entries. (A totally empty INTERFACE section is not allowed.) This section contains the information which is public to both the host program and the unit (and visible from the LIBMAP program).

4: These entries are optional. If used, they are public (see #3). If a nested unit is used, it MUST be declared at this point, and the host program must also declare it in its USES statement before declaring this unit. (See pages 75-81 in the Pascal Language Reference Manual.)

5: VARs are also optional. If VARs are used in an intrinsic unit, a DATA segment MUST be declared, since these are global variables.

6: These are the public statements of the unit's procedures and functions. All parameters MUST be declared in this section, and must not appear in the IMPLEMENTATION. EXTERNAL may not be specified at this point (see #10).

7: IMPLEMENTATION is required. This section contains the code-generating statements and any private types or variables to be declared. If an item is declared in this section only, it is local to the unit and cannot be accessed from the host program directly; it is also not visible to LIBMAP.

8: CONST and TYPE definitions are optional. If included at this point, they are private and will not be visible to the calling program.

9: VARs are optional, and are private if defined at this point. A DATA segment is required if global VARs are specified in the IMPLEMENTATION section of an intrinsic unit, even though they are private.

10: These are the actual code-generating procedures. Procedures and functions already declared in the INTERFACE section need not have parameters listed here, since that would be redundant. EXTERNAL references are specified at this time.

11: This is the initialization section. Code in this portion is optional, but the BEGIN and END statements must always be present. The final END statement must be followed by a "." to indicate the end of the text. Code placed in the initialization section will be executed immediately upon access to the unit (through the USES statement in the host program), and ignored thereafter. An example of this is Turtlegraphics, where the initialization code is responsible for allocating the high-res page so that variables will not be lost.

Apple Tech Notes

Tech Info Library Article Number:693