APPENDIX A

RELOCATABLE OBJECT MODULE FILE FORMAT

Relocatable Object Module Storage Format

The VERSAdos operating system stores relocatable object modules in sequential files with fixed length records of 256 bytes. (Refer to the VERSAdos Data Management Services and Program Loader User's Manual for information on the file formats supported by the operating system.)

Within each 256-byte record a variable number of variable length relocatable object records are stored. Each one of these records consists of a 1-byte byte count followed by the actual data of the relocatable record. The byte count indicates the number of data bytes that follow in the record. The byte count may contain any value between 0 and 255, inclusive. A byte count of zero indicates a relocatable object record with no data bytes (a record of this type is ignored by the linkage editor). Thus, the length of one relocatable object record is limited to a total of 256 bytes; one byte for the byte count and a maximum of 255 data bytes.

However, this does not mean that only one variable length relocatable object record can be stored in one fixed length record. Each 256-byte fixed length record is totally filled before continuing to the next 256-byte record. Thus, it is possible for a variable length record to be divided between two fixed length records. For example, suppose the first three relocatable object records of a relocatable object module contained 50, 150, and 200 bytes of data, respectively. The first two records, along with their byte counts, would be stored within the first 202 bytes of the first 256-byte fixed-length record. This would leave 54 bytes remaining in that record. These 54 bytes would be filled with the byte count of the third relocatable object record, followed by the first 53 data bytes of that record. The remaining 147 data bytes of the third relocatable object record would then be stored at the beginning of the second 256-byte fixed length record.

Any space not used in the last 256-byte fixed length record of a relocatable object module file must be filled with binary zeros. This fills out the rest of the file with relocatable object records that have zero bytes of data (which are ignored by the linkage editor).

Relocatable Object Record Format

There are four basic types of relocatable object records. The record type is indicated by the first byte of data (the byte immediately after the byte count), in the record. This byte is the ASCII code of one of the digits between "1" and "4", inclusive. The byte values and the types of records are:

Value of First Data Byte	Record Type
1 ($31)	Identification Record
2 ($32)	External Symbol Definition Record
3 ($33)	Object Text Record
4 ($34)	End Record

The formats of these four record types is discussed in detail in the following paragraphs.

Identification Record (Type 1)

Each relocatable object module must contain an identification record as the first record in the module. It is this record that indicates the beginning of a relocatable object module. The identification record contains general information about the relocatable object module, such as its name, version and revision, what language processor was used to create the module, what source file was used to create the module, the time and date the module was created, and a description of the module. The format of an identification record is:

```
      +----+---+-----+-+-+-+---+----+---+-----+---+----+----+-----+
Bytes | 1  | 1 | 10  |1|1|1| 4 | 2  | 8 |  8  | 2 | 3  | 3  |0-211|
      +----+---+-----+-+-+-+---+----+---+-----+---+----+----+-----+
Field |Size| 1 |Mname|V|R|L|Vol|User|Cat|Fname|Ext|Time|Date|Descr|
      +----+---+-----+-+-+-+---+----+---+-----+---+----+----+-----+
```

Field	Size (Bytes)	Contents
1	1	Record Type (ASCII $31).
Mname	10	Module name (ASCII).
V	1	Module version number (0-255).
R	1	Module revision number (0-255).
L	1	Language processor type (ASCII): A ($41) - Assembler B ($42) - BASIC C ($43) - COBOL F ($46) - FORTRAN P ($50) - Pascal
Vol	4	Source file volume name (ASCII).
User	2	Source file user number (0-9999).
Cat	8	Source file catalog name (ASCII).
Fname	8	Source file filename (ASCII).
Ext	2	Source file extension (ASCII).
Time	3	Module creation time (hhmmss). (NOTE)
Date	3	Module creation date (mmddyy). (NOTE)

Descr	varies	Module description (ASCII). Occupies remainder of identification record as indicated by record length. May be 0-211 bytes (characters) long.

(NOTE) Time and date are stored in a BCD format with two decimal digits per byte. For example, if the time of creation was 9:27:56, it would be stored in the identification record as $092756.

External Symbol Definition Record (Type 2)

Each external symbol definition record contains a variable number of External Symbol Definitions (ESDs) and defines a relocatable section, a common section, an absolute section, an externally defined symbol, an externally referenced symbol, or a command line address. A 1-byte value at the beginning of the ESD indicates the type of ESD within an external symbol definition record. The high order nibble of the byte indicates the ESD type while the low order nibble of the byte indicates what section the ESD refers to. The format of an external symbol definition record is:

```
        +----+---+--------+----+
  Bytes | 1  | 1 |    1   | Var|
        +----+---+--------+----+-------+----+---/ /+-------+----+
  Field |Size| 2 |Typ/Sct |Data|Typ/Sct|Data|      |Typ/Sct|Data|
        +----+---+--------+----+-------+----+-/ /--+-------+----+
```

Field	Size (Bytes)	Constants
2	1	Record type (ASCII $32)
Typ/Sct	1	Typ (high nibble) = Type of ESD

0 - Absolute section

1 - Common section (in section Sct)

2 - Standard relocatable section (section number Sct)

3 - Short address relocatable section (section number Sct)

4 - External symbol definition (in relocatable section Sct)

5 - External symbol definition (in an absolute section)

6 - External symbol reference (to section Sct)

7 - External symbol reference (to any section)

```
                              8 - Command line address (in section Sct)

                              9 - Command line address (in an absolute section)

                              A - Command line address (in a common section in
                                  section Sct)

                          Sct (low nibble) = Relocatable section referring to
                                             (0-15)

   Data      Varies    Depends on Typ (see below)
```

Several ESD entries may be included in one ESD record. The following descriptions outline the contents of the Data field from the general ESD record format:

```
                              +---+----+-----+
                        Bytes | 1 | 4  |  4  |
                              +---+----+-----+
   Absolute section           |0/0|Size|Start|
                              +---+----+-----+

                              +-----+------+----+
                        Bytes |  1  |  10  | 4  |
                              +-----+------+----+
   Common section             |1/Sct|Common|Size|
                              +-----+------+----+

                              +-----+----+
                        Bytes |  1  | 4  |
                              +-----+----+
   Standard relocatable       |2/Sct|Size|
   section                    +-----+----+

                              +-----+----+
                        Bytes |  1  | 4  |
                              +-----+----+
   Short address              |3/Sct|Size|
   relocatable section        +-----+----+

                              +-----+----+-------+
                        Bytes |  1  | 10 |   4   |
                              +-----+----+-------+
   XDEF (in                   |4/Sct|XDEF|Address|
   section Sct)               +-----+----+-------+

                              +---+----+-------+
                        Bytes | 1 | 10 |   4   |
                              +---+----+-------+
   XDEF (in an                |5/0|XDEF|Address|
   absolute section)          +---+----+-------+
                          +-----+----+
                    Bytes |  1  | 10 |
                          +-----+----+
XREF (to                  |6/Sct|XREF|
section Sct)              +-----+----+

                          +---+----+
                    Bytes | 1 | 10 |
                          +---+----+
XREF (to any              |7/0|XREF|
section)                  +---+----+

                          +-----+------+--------+
                    Bytes |  1  |  4   |   1    |
                          +-----+------+--------+
Command line address      |8/Sct|CL Adr|CL Lngth|
(in section Sct)          +-----+------+--------+

                          +---+------+--------+
                    Bytes | 1 |  4   |   1    |
                          +---+------+--------+
Command line address      |9/0|CL Adr|CL Lngth|
(in an absolute section)  +---+------+--------+

                          +-----+------+------+--------+
                    Bytes |  1  |  10  |  4   |   1    |
                          +-----+------+------+--------+
Command line address      |A/Sct|CL Com|CL Adr|CL Lngth|
(in a common section)     +-----+------+------+--------+
```

Field	Size (Bytes)	Contents
Size	4	Length of section (in bytes).
Start	4	Starting address of absolute section.
Common	10	Name of common section (ASCII).
XDEF	10	Name of **XDEF** symbol (ASCII).
Address	4	Address of symbol within its section (in 5/0 this is an absolute address).
XREF	10	Name of **XREF** common symbol (ASCII).
Cl Adr	4	Address of command line within its section (in 9/0 this is an absolute address).
CL Lngth	1	Maximum length of command line -1. (0-255 represents 1-256.)
CL Com	10	Name of common section that contains the command line (ASCII).

ESDs have a restriction that all ESDs defining externally defined and externally referenced symbols (ESD types 4-7) must appear before any other types of ESDs in the module.

Each section (relocatable, common, and absolute) and each external reference in a relocatable object module is assigned an index so that they may be easily referenced later in the relocatable object module. This index is an External Symbol Definition Index (ESDID). Since a module may contain only one of each type of relocatable section (sections 0-15), the ESDID for a relocatable section is simply the section number plus 1. Thus, the index for section 12 is 13. However, a relocatable object module may contain multiple common sections, absolute sections, and external references. Therefore, indices for these types of ESDs are assigned in increasing numerical order, starting at 17, in the order the ESDs are encountered in the module. Thus, the index of the first ESD of the type 0, 1, 6, or 7 is 17; the index of the second ESD of those types is 18; and so on. A Pascal-like algorithm for assigning ESDIDs when reading a relocatable object module is:

```
    i  := 17;
    WHILE reading ESDs DO
         BEGIN
              read an ESD of Typ/Sct;
              CASE Typ OF
                 0,1,6,7:      BEGIN
                                 ESDID  := i;
                                 i      := i + 1
                               END;
                 2,3:          ESDID   := Sct + 1;
                 4,5,8,9,10:   {no ESDID assigned};
              END; {CASE}
              process the ESD
         END  {WHILE}
```

where:

ESDID is the ESD index for the ESD that is currently being processed.

NOTE: ESDs that describe externally defined symbols and command line addresses are not assigned indices. This is because these types of ESDs do not need to be referred to later in the relocatable object module.

New ESDIDs are assigned for each relocatable object module processed. Thus, the ESDIDs in one module have no relation to the ESDIDs in another module. Each module is limited to a total of 255 ESDIDs (numbered 1 through 255). Since ESDIDs 1 through 16 always refer to the relocatable sections 0 through 15, a relocatable object module may contain at most a total of 239 absolute sections, common sections, and external references.

Object Text Record (Type 3)

Object text records define the actual code and data to be put in the resulting load module (or relocatable object module). Each object text record contains absolute code along with relocation data for computing relocated code. A bit map is employed to indicate the data that is absolute code and the data that is relocation data. The format of an object text record is:

```
       +----+---+---+-----+
Bytes  | 1  | 1 | 4 |  1  |
       +----+---+---+-----+--------+
       |Size| 3 |Map|ESDID|Data    |
       +----+---+---+-----+--------+
```

Field	Size (Bytes)	Contents
3	1	Record type (ASCII $33).
Map	4	Bit map - each bit corresponds to one 16-bit word of absolute code or one set of relocatable data: 0 - Absolute code (16 bits each - word) 1 - Relocation data (1 to 12 bits)
ESDID	1	ESD index indicating the ESD for the section in which data from this record is to be placed.
Data	varies	Absolute code alternating with relocation data as per the bit map.

ESDID is a 1-byte ESD index that indicates in what section the code generated from this object text record is to be located. The way it works is:

> A "program counter" is maintained for each section (relocatable, common, and absolute) in the relocatable object module being processed. Each program counter is initialized to the starting address of the section it represents. When an object text record is processed, the code generated from the record is placed at the address indicated by the program counter for the section whose ESD has the index indicated by ESDID. As code is generated, the program counter for the section into which the code is being placed is updated to indicate where the next code for that section should go.

Data is a variable length field that can contain up to 32 16-bit words of absolute code (code that does not need to be relocated), or up to 32 sets of relocation data, or any combination thereof. The data field is interpreted as:

> The highest order (leftmost) bit in the bit map corresponds to the first (leftmost) element in the data field. If the highest order bit in the bit map is a zero then the first 16-bit word in the data field is absolute code. However, if the highest order bit in the bit map is a 1, then the

first data item in the data field is relocation data. The second highest order bit in the bit map corresponds to the next data item in the data record in the same way, and so on. This processing of data proceeds until data corresponding to all 32 bits in the bit map has been processed or the end of the object text record (as indicated by the record length) is encountered, whichever comes first.

As previously mentioned, a zero bit in the bit map indicates one 16-bit word of absolute code that does not need to be relocated. This word is stored in the data field in two bytes in the exact form it is to appear in the resulting load module (or relocatable object module).

On the other hand, a 1 bit in the bit map indicates a set of relocation data in the data field. A set of relocation data is the data required to relocate a single 16-bit or a single 32-bit quantity. A set of relocation data is of variable length and can occupy from 1 to 12 bytes of data in the data field. The format of relocation data is:

```
      +----+--------+--------+
Bytes | 1  | 0 to 7 | 0 to 4 |
      +----+--------+--------+
      |Flag| ESDIDs | Offset |
      +----+--------+--------+
```

Field	Size (Bytes)	Contents
Flag	1	Indicates type of relocation data: bits 7-5 - Number of ESDIDs (0-7) bit 4 - Reserved - must be 0 bit 3 - Size of relocated data 0 - 1 word (16 bits) 1 - 2 words (32 bits) bits 2-0 - Offset field length in bytes (0-4)
ESDIDs	0 to 7	ESD indices involved in relocation (1 byte each).
Offset	0 to 4	Constant offset involved in relocation.

The purpose of relocation data is to instruct the linkage editor how to calculate a relocated value. For each set of relocation data, this relocated value is initialized to zero. The relocation data is interpreted as:

Relocation data contains up to seven ESDIDs. Each ESDID in relocation data represents a numerical value. Typically, an ESDID will be the index of an ESD representing an external symbol reference. In that case, the numerical value associated with the ESDID is the address in the result module of the referenced symbol. However, an ESDID in relocation data may also be the index of an ESD that represents a section (relocatable, common, or absolute). Here, the numerical value associated with the ESDID is the starting address of the particular section in the result module. The numerical values associated with the first, third, fifth, and seventh ESDIDs in relocation data are added to the relocated value while the values associated with the second, fourth, and sixth ESDIDs are subtracted from the relocated value. A zero ESDID in relocation data indicates that there is no ESDID for that position and therefore, no corresponding numerical value to be added or subtracted.

Relocation data may also contain a constant offset that is added to the relocated value. This offset may be from 0 to 4 bytes long and is always interpreted as a 2's complement value (thus the value -1 may be represented in one byte as $FF). An offset that is 0 bytes long indicates that there is no constant offset.

Once a relocated value has been calculated by adding and subtracting the proper ESDID values and adding the constant offset, it is placed in the resulting module in the size indicated by bit 3 of the flag byte. If this bit is off (0) then the relocated value will be put into the result module as a one-word 16-bit) value. If the bit is on (1), the relocated value will be placed into the result module as a two-word (32-bit) value.

Finally, if bits 7-5 of the flag byte indicate that there are no ESDIDs in the relocation data, then there must be an offset. Here, the offset (from 1 to 4 bytes long) is taken as a 2's complement value to be added to the current program counter for the section in which code from this object text record is being placed. Later code for the same section will be placed starting at the new location. This allows the relocating of program counters backward and forward for overwriting code that has already been generated ("fix-ups").

End Record (Type 4)

This record indicates the end of a relocatable object module and must be the last record in every module. It also contains information about the starting execution address of the module. The format of the end record is:

```
      +----+---+---+-------+
Bytes | 1  | 1 | 1 |   4   |
      +----+---+---+-------+
      |Size| 4 |Sct|Address|
      +----+---+---+-------+
```

Field	Size (Bytes)	Contents
4	1	Record type (ASCII $34).
Sct	1	Section in which execution starts: 0-15 - Relocatable section 16 - Absolute section 17 - No starting address
Address	4	Starting execution address.

If the value of Sct is between 0 and 15, inclusive, then Address is interpreted as being relative to the start of the section. If the value of Sct is 16, then Address is an absolute address. Finally, if the value of Sct is 17, then no starting execution address has been specified for this module and address does not appear in the end record.