Subject: Re: Porting VMS "writeu" files to UNIX
Posted by thompson on Mon, 19 Oct 1992 14:34:00 GMT
View Forum Message <> Reply to Message

In article <gotwols.719353668@warper.jhuapl.edu>, gotwols@warper.jhuapl.edu (Bruce Gotwols) writes...
> I have found that it is very difficult to port IDL files made using
> unformatted writes (eg. writeu,1,buf) to other systems such as UNIX. The
> culprit seems to be that under VMS these files have a record attribute of
> implied carriage return. If I Kermit the files (having set binary) I get an
> irrelevant line feed (ironically not a carriage return) at the end of each
> record. Does anyone have a suggestion?

You might use the /NONE keyword on the OPENW call. This should solve your immediate problem.

As has already been pointed out, one could use XDR when opening the file for read and write. This should make the file completely transportable. However, the record attribute is still Carriage-Return-Carriage-Control, so you may still have problems. Also, this does put some restrictions on the kind of I/O you can and cannot do with the file. In particular, you can't write fixed-length record (direct access) files, which I like to use.

I've had good luck writing transportable files using the following technique.

1. Open the file with the /BLOCK keyword set (then /NONE not needed).
2. Use the BYTEORDER routine to convert the data to or from network format (e.g. using the HOST_TO_IEEE and IEEE_TO_HOST routines below) before writing or after reading the data.
3. Use only READU and WRITEU for I/O.
4. Use POINT_LUN to go directly to any desired point in the file.

The disadvantage of this approach is that it doesn't allow you to use any of the VMS record-oriented services. However, you can't use them on UNIX machines anyways.

Bill Thompson

```
 ================================================================
==================
 PRO HOST_TO_IEEE, DATA, DATATYPE=DATATYPE
;+
; NAME:
; HOST_TO_IEEE
; PURPOSE:
; To translate an IDL variable in the host architecture into the IEEE-754
; standard (e.g. as used in FITS files).
; CALLING SEQUENCE:
```

```
; HOST_TO_IEEE, DATA
; INPUT/OUTPUT PARAMETERS:
; DATA  = Any IDL variable, scalar or vector.  DATA will be modified
;     by HOST_TO_IEEE to convert it to IEEE format.  Byte and
;     string variables are returned by HOST_TO_IEEE unchanged.
; OPTIONAL KEYWORD PARAMETERS:
; DATATYPE = Numerical code for IDL data type to use for interpreting the
;     data instead of using the data type of the IDL variable.
; RESTRICTIONS:
; None.
; SIDE EFFECTS:
; None.
; PROCEDURE:
; BYTEORDER is used to convert from the host format to network format for
; integer arrays or XDR format for floating point arrays.  These are
; equivalent to IEEE format.
; MODIFICATION HISTORY:
; Adapted from TRANS_BYTES  W. Landsman   Hughes/STX    January, 1992
; William Thompson, Feb. 1992, modified to use BYTEORDER.
;-
;
 ON_ERROR,2
;
;  Check the number of parameters.
;
 IF N_PARAMS() EQ 0 THEN MESSAGE,'Syntax:  HOST_TO_IEEE, DATA'
;
;  If DATATYPE was not passed, then use the data type of the IDL variable.
;
 IF N_ELEMENTS(DATATYPE) EQ 1 THEN TYPE = DATATYPE ELSE BEGIN
  SZ = SIZE(DATA)
  TYPE = SZ(SZ(0)+1)
 ENDELSE
;
;  Convert the data depending on the data type.
;
 CASE TYPE OF
  0:  MESSAGE,'DATA not defined'
  1:  RETURN   ;Byte
  2:  BYTEORDER,DATA,/HTONS ;Short integer
  3:  BYTEORDER,DATA,/HTONL ;Long integer
  4:  BYTEORDER,DATA,/FTOXDR ;Floating point
  5:  BYTEORDER,DATA,/DTOXDR ;Double precision
  6:  BYTEORDER,DATA,/FTOXDR ;Complex
  7:  RETURN   ;String
  8:  BEGIN   ;Structure
   NTAG = N_TAGS(DATA)
   FOR T = 0,NTAG-1 DO BEGIN
```

```
    TEMP = DATA.(T)
    HOST_TO_IEEE, TEMP
    DATA.(T) = TEMP
   ENDFOR
   END
  ELSE:  MESSAGE,'Unrecognized data type'
 ENDCASE
;
 RETURN
 END
 ================================================================
==================
 PRO IEEE_TO_HOST, DATA, DATATYPE=DATATYPE
;+
; NAME:
; IEEE_TO_HOST
; PURPOSE:
; To translate an IDL variable into the host architecture from the
; IEEE-754 standard (e.g. as used in FITS files).
; CALLING SEQUENCE:
; IEEE_TO_HOST, DATA
; INPUT/OUTPUT PARAMETERS:
; DATA  = Any IDL variable, scalar or vector.  DATA will be modified
;     by IEEE_TO_HOST to convert it from IEEE format.  Byte and
;     string variables are returned by IEEE_TO_HOST unchanged.
; OPTIONAL KEYWORD PARAMETERS:
; DATATYPE = Numerical code for IDL data type to use for interpreting the
;     data instead of using the data type of the IDL variable.
; RESTRICTIONS:
; None.
; SIDE EFFECTS:
; None.
; PROCEDURE:
; BYTEORDER is used to convert to the host format from network format for
; integer arrays or XDR format for floating point arrays.  These are
; equivalent to IEEE format.
; MODIFICATION HISTORY:
; Adapted from TRANS_BYTES  W. Landsman   Hughes/STX    January, 1992
; William Thompson, Feb. 1992, modified to use BYTEORDER.
;-
;
 ON_ERROR,2
;
;  Check the number of parameters.
;
 IF N_PARAMS() EQ 0 THEN MESSAGE,'Syntax:  IEEE_TO_HOST, DATA'
;
;  If DATATYPE was not passed, then use the data type of the IDL variable.
```

```
;
 IF N_ELEMENTS(DATATYPE) EQ 1 THEN TYPE = DATATYPE ELSE BEGIN
  SZ = SIZE(DATA)
  TYPE = SZ(SZ(0)+1)
 ENDELSE
;
;  Convert the data depending on the data type.
;
 CASE TYPE OF
  0:  MESSAGE,'DATA not defined'
  1:  RETURN   ;Byte
  2:  BYTEORDER,DATA,/NTOHS ;Short integer
  3:  BYTEORDER,DATA,/NTOHL ;Long integer
  4:  BYTEORDER,DATA,/XDRTOF ;Floating point
  5:  BYTEORDER,DATA,/XDRTOD ;Double precision
  6:  BYTEORDER,DATA,/XDRTOF ;Complex
  7:  RETURN   ;String
  8:  BEGIN   ;Structure
   NTAG = N_TAGS(DATA)
   FOR T = 0,NTAG-1 DO BEGIN
    TEMP = DATA.(T)
    IEEE_TO_HOST, TEMP
    DATA.(T) = TEMP
   ENDFOR
   END
  ELSE:  MESSAGE,'Unrecognized data type'
 ENDCASE
;
 RETURN
 END
```