
Subject: Re: Reading real numbers from VMS
Posted by [hofer](#) on Sat, 26 Sep 1992 11:37:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <1992Sep25.234223.15365@news2.cis.umn.edu>, grad@sparky.drad.umn.edu (Jonathan Grad) writes:

> I have some real numbers (real *4) that were saved on a VAX using VMS, and
> have been imported to a Unix machine. Does anyone have a pwave
> program to read in real numbers (saved under VMS) to a Unix system?
>
> Thanks in advance.
>
> Jonathan Grad
> grad@sparky.drad.umn.edu

I've written the following routine to convert a VMS F float (4 bytes) to
to pv-wave floats. Hope this helps.

Remo Hofer

```
--
      RFC822: <hofer@urz.unibas.ch> or <hofer%urz.unibas.ch@CERNVAX.BITNET>
      X.400: S=hofer;OU=urz;O=unibas;P=SWITCH;A=ARCOM;C=CH
      HEPNET/SPAN: CHGATE::YOGI::HOFER or 20579::48130::HOFER
-----8<-----8<-----
;-----
; Convert a VAX F float number to WAVE double precision. RHo
; last modified      29/1/92
;-----
;
;+
;1 VAXF_WAVE
;!
; Purpose:
;
; This procedure converts a VAX single precision floating point
; number (F float) into a WAVE double precision number.
; The conversion is independent of the exact format used by the
; machine WAVE runs on.
;
; Calling Sequence:
;
; SXM_VAXF_WAVE, buffer, value
;!
;!-----
;2 Input
; Name Type Description
;
```

```

; buffer byte (4) Buffer of four bytes that are to be
;   converted into a double precision number
; !
; !-----
; 2 Output
; Name Type Description
;
; value dfloat The result of the conversion
; !
; !-----
; 2 Procedure
; The procedure tries to convert the four bytes in the buffer into a WAVE
; double precision floating point number. The conversion performs the
; interpretation of the buffer contents independent of the floating point
; architecture of the CPU/FPU on which WAVE is running. The buffer is
; interpreted according to the VAX single precision F floating point speci-
; fication.
; -
; -----

PRO sxm_vaxf_wave, buffer, value

; check dimensions, size and type of the buffer:

save = SIZE (buffer)
IF (save(0) NE 1) OR (save(1) NE 4) OR (save(save(0) + 1) NE 1) THEN $
BEGIN
  PRINT, 'VAX double precision buffer of wrong data type.'
  value = 0D
  RETURN
ENDIF

; determine sign, exponent and mantissa:

s = buffer(1)/128B
e = 2*(buffer(1) AND 127B) + buffer(0)/128B
f = 65536L*(buffer(0) AND 127B) + 256L*buffer(3) + buffer(2)

; buffer represents a valid, finite number:

IF e EQ 0 THEN value = 0D ELSE value = (2D^(e-129)) * (f/(2D^23) + 1D)
IF s EQ 1 THEN value = (-1) * value
RETURN

END

```
