
Subject: USGS SDTS - revisited

Posted by [Kelly Dean](#) on Mon, 04 Dec 2000 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Below is a crude routine to read the USGS 1:24,000 scale DLG and DEM files in the Spatial Data Transfer Standard (SDTS) format. These data files are made available at the EROS Data Center. The conversion to SDTS is still underway, so some Quads may not be available at the EROS Data Center. These routines will display the elevation data from the DEM and the lines for the road DLG.

<http://edcwww.cr.usgs.gov/doc/edchome/ndcddb/ndcddb.html>

Some research needs to be done before the routines are executed in IDL. However, all the required information are in the ASCII part of the Data Dictionary/Definition (ddf) files. I did not put in the effort to extract this information from these individual files.

```
nelem = 354L ; From Raster Definition ( *RSDF.ddf )
nline = 463L ; From Raster Definition ( *RSDF.ddf )
minhgt = 1529L ; From Data Dictionary/Domain (*DDOM.ddf )
maxhgt = 2334L ; from Data Dictionary/Domain ( *DDOM,ddf )
XRange = [ 47881597, 48942766 ] ; Spatial Domain ( *SPDM.ddf )
YRange = [ 448305100, 449694850 ] ; Spatial Domain ( *SPDM.ddf )
```

I made some comment earlier about USGS putting in extract CR/LF in the files. Well, it was actually WinZip doing this to me. So for Windows users that use WinZip to decompress and detar, you need to change the WinZip default settings.

Under "Options/Configuration", make sure the checkbox for "TAR file smart CR/LF translation" is NOT checked.

This program was developed under Windows 2000, so if you try this on a UNIX machine, the keyword /Swap_Endian will need to be removed from the Open statements. The /Swap_Endian keyword should remain for Linux and VAX users.

So if you give this a try - Good Luck. I would be interested in hearing your comments about these routines. All this is new territory for me.

Kelly Dean
CSU/CIRA
krdean@lamar.colostate.edu

```
=====TestQuad=====
FUNCTION ReadDLG, FileName, XRange, YRange
;
```

```

; Read USGS DLG files in the Spatial Data Transfer Standard (SDTS)
format
;
; Digital Line Graphs (DLG) - 1:24,000-scale
;
; http://edcwww.cr.usgs.gov/doc/edchome/ndcldb/dlg_large/states .html
;
; For Windows users that use WinZip to decompress and detar, you need to
change the default settings.
;
; Under "Options/Configuration", make sure the checkbox for "TAR file
smart CR/LF translation" is NOT checked.
;
status = 0
miny = 999999999
maxy = -999999999
minx = 999999999
maxx = -999999999
OpenR, lun, filename, /Get_LUN, /SWAP_Endian
FileInfo = FStat( lun )
bchar = 0B
manychr = ''
FileSize = FileInfo.Size
LE01 = BYTarr(FileSize)
ReadU, lun, LE01
CLOSE, lun
Free_LUN, lun
status = 1
;
;
;
;
FOR i = 0L, FileSize-1L DO BEGIN
  bchar = LE01(i:i)
  char = STRING([bchar])
  IF ( bchar(0) EQ 30 ) THEN BEGIN
    IF ( STRMID( manychr, 1,2 ) EQ 'LE' ) THEN BEGIN
;     LineNum = LONG(STRMID( manychr, 5, 6 ))
    ENDIF ELSE IF ( STRMID( manychr, 1,2 ) EQ 'NO' ) THEN BEGIN
;     PRINT, ' ManyChr >', ManyChr
    ENDIF ELSE IF ( STRMID( manychr, 1,2 ) EQ 'AC' ) THEN BEGIN
;     PRINT, ' ManyChr >', ManyChr
    ENDIF ELSE IF ( STRMID( manychr, 1,2 ) EQ 'PC' ) THEN BEGIN
;     PRINT, ' ManyChr >', ManyChr
    ENDIF ELSE IF ( STRMID( manychr, 1,4 ) EQ 'ARDF' ) THEN BEGIN
;     PRINT, ' ManyChr >', ManyChr
    ENDIF ELSE IF ( STRMID( manychr, 1,4 ) EQ 'ARDM' ) THEN BEGIN
;     PRINT, ' ManyChr >', ManyChr
    ENDIF ELSE IF ( STRMID( manychr, 33,4 ) EQ 'LINE' ) THEN BEGIN

```

```

posSADR = STRPOS( ManyChr, 'SADR' )
lenManyChr = STRLEN( ManyChr )
dif = LenManyChr - posSADR - 1
SADR = LONG(STRMID( ManyChr, posSADR+4, dif ))
lenSADR = ( SADR / 100L )
locSADR = ( SADR MOD ( lenSADR * 100L ) ) + 1
bSADR = LE01(i+locSADR:i+locSADR+lenSADR-1)
numSADR = lenSADR / 4 / 2
xSADR = LONarr(numSADR)
ySADR = LONarr(numSADR)
k = 0
FOR j = 0, lenSADR-8, 8 DO BEGIN
  xs = j
  xe = j + 3
  ys = j + 4
  ye = j + 7
  xSADR(k) = LONG( TOTAL( ISHFT( LONG(bSADR(xs:xe))), [24,16,8,0] )
))
  ySADR(k) = LONG( TOTAL( ISHFT( LONG(bSADR(ys:ye))), [24,16,8,0] )
))
  IF ( xsadr(k) LT 0L ) THEN PRINT, k, xSADR(k)
  IF ( ysadr(k) LT 0L ) THEN PRINT, k, ySADR(k)
;   IF ( k EQ 2 ) THEN xsadr(k) = xSADR(1)
;   IF ( k EQ 2 ) THEN ysadr(k) = ySADR(1)
  k = k + 1
ENDFOR
miny = [ miny, ySADR ]
miny = MIN(miny)
maxy = [ maxy, ySADR ]
maxy = MAX(maxy)
minx = [ minx, xSADR ]
minx = MIN(minx)
maxx = [ maxx, xSADR ]
maxx = MAX(maxx)
PLOT, xSADR, ySADR, $
  /NoErase, $
  XRange = XRange, $
  YRange = YRange, $
  XMargin= [0 , 0], YMargin=[ 0, 0 ], $
  XStyle=5, YStyle=5
xSADR = 0L
ySADR = 0L
ENDIF ELSE BEGIN
;   PRINT, ' ManyChr >', ManyChr
ENDELSE
  manychr = ''
ENDIF ELSE BEGIN
  manychr = manychr + char

```

```

    ENDELSE
ENDFOR
;
;
;
PRINT, ' XRange = [ ', minx, ',', maxx, ' ]'
PRINT, ' YRange = [ ', miny, ',', maxy, ' ]'
;
;
RETURN, status
END

FUNCTION ReadDDF, FileName
;
; Read USGS SDTS file (.ddf )
;
status = 0
;
; Open SDTS file
;
OpenR, lun, filename, /Get_LUN
bchar = 0B
manychr = ''
;
; Read one byte at a time until EOF
;
WHILE NOT EOF(lun) DO BEGIN
    ReadU, lun, bchar
    ;
    ; If located a Graphic Character, print and reset string.
    ;
    IF ( bchar EQ 30B OR bchar EQ 31B ) THEN BEGIN
        PRINT, '>', manychr, '<'
        manychr = ''
    ENDIF ELSE BEGIN
        manychr = manychr + STRING([bchar])
    ENDELSE
ENDWHILE

CLOSE, lun
Free_LUN, lun
status = 1
;
;
RETURN, status
END
;

```

```

;
;
;
FUNCTION ReadDEM, FileName, img, xsize, ysize
;
; Read USGS DLF files in the Spatial Data Transfer Standard (SDTS)
format
;
; USGS 7.5 Minute Digital Elevation Models (DEM) - 1:24,000-scale
;
; http://edcwww.cr.usgs.gov/doc/edchome/ndcddb/7_min_dem/states.html
; Determine the array length
;
; For Windows users that use WinZip to decompress and detar, you need to
change the default settings.
;
; Under "Options/Configuration", make sure the checkbox for "TAR file
smart CR/LF translation" is NOT checked.
;
;
; Read USGS DEM/STDS
;
RecModule = { RecHead, $
    GChar0:0B, $
    RecID:' ', $
    B:' ', $
    C:' ', $
    D:' ', $
    Easting:STRING(' ',FORMAT='(A11)'), $
    CellID:' ', $
    CellNum:' ', $
    CVLSID:' ', $
    CVLSNum:' ', $
    GChar1:0B, $
    BNum:' ', $
    GChar2:0B, $
    Cel0:' ', $
    GChar3:0B, $
    LNum0:' ', $
    GChar4:0B, $
    LNum:' ', $
    GChar5:0B, $
    z:' ', $
    GChar6:0B }

;data = INTarr(354)

status = 0
OpenR, lun, filename, /Get_LUN, /SWAP_Endian

```

```
POINT_LUN, lun, 186 ; Jump File header
```

```
FOR i = 0, ysize-1 DO BEGIN
  ReadU, lun, RecModule
  ;
  ; Initialize some variables
  ;
  IF ( i EQ 0 ) THEN BEGIN
    RecLen = FIX(RecModule.RecID)
    RecLen0 = FIX(RecModule.RecID)
    imgRec = INTarr(xsize) ; Length of image record
    img = INTarr(xsize, ysize)
  ENDIF
  RecLen = FIX(RecModule.RecID)
  IF ( RecLen GT RecLen0 ) THEN BEGIN
    POINT_LUN, -lun, position
    RecLen = FIX(RecModule.RecID)
    NewPos = position + ( RecLen - RecLen0 )
    POINT_LUN, lun, NewPos
    ReadU, lun, imgRec
  ENDIF ELSE BEGIN
    ReadU, lun, imgRec
    RecLen0 = RecLen
  ENDELSE
  img(*,i) = imgRec
;ENDWHILE
ENDFOR
```

```
CLOSE, lun
Free_LUN, lun
status = 1
```

```
RETURN, status
END
```

```
PRO TestQuad, FileName, AllDDF=AllDDF
;
;
;
; Keyword:
; AllDDF : Print results from all Data Dictionary/Definition (ddf)
files.
;
; Settings for Horsetooth, Colorado Quad
;
nelem = 354L ; From Raster Definition ( *RSDF.ddf )
nline = 463L ; From Raster Definition ( *RSDF.ddf )
```

```

minhgt = 1529L ; From Data Dictionary/Domain (*DDOM.ddf )
maxhgt = 2334L ; from Data Dictionary/Domain ( *DDOM,ddf )
; XRange = [ 47881597, 48942766 ] ; Spatial Domain ( *SPDM.ddf ) -
find to be inaccurate - KRD
; YRange = [ 448305100, 449694850 ] ; Spatial Domain ( *SPDM.ddf ) -
find to be inaccurate - KRD
XRange = [ 47881680, 48942776 ] ; Numbers from first run thru
ReadDLG
YRange = [ 448326208, 449716000 ] ; Numbers from first run thru
ReadDLG
DEMfilter = 'd:\data\usgs\dem\Horsetooth\*.ddf'
DLGfilter = 'd:\data\usgs\dlg\Horsetooth\*.ddf'
;
;
; Read a USGS SDTS file
;
files = FindFile(DEMfilter, Count=nfiles)

FOR i = 0, nfiles-1 DO BEGIN
  Filename = files(i)
  ;
  ;
  ;
  ;
  IF ( STRPOS( FileName, 'CEL0' ) EQ -1 ) THEN BEGIN
    PRINT, ' Working with >', Filename
    IF ( KEYWORD_SET(AIIDDF) ) THEN BEGIN
      IF ( ReadDDF( FileName ) ) THEN BEGIN
        PRINT, ' Success!'
      ENDIF ELSE BEGIN
        PRINT, ' Unable to read >', Filename
      ENDELSE
    ENDIF
  ENDIF ELSE BEGIN
    PRINT, ' Working with cle0>', FileName
  ;
  ;
  ;
  ;
  IF ( ReadDEM( FileName, img, nelemt, nline ) ) THEN BEGIN
    PRINT, ' Success!'
    PRINT, MAX(img), MIN(img)
    img8 = BYTscl(img, MIN=minhgt, MAX=maxhgt )
    Title = '7.5 Minute'
    WINDOW, 0, XSize = nelemt, YSize = nline, Title=Title
    TV, img8, /order
  ENDIF ELSE BEGIN
    PRINT, ' Unable to read >', Filename
  ENDELSE
ENDELSE

```

```

ENDFOR
;
;
;
;
files = FindFile(DLGfilter, Count=nfiles)

FOR i = 0, nfiles-1 DO BEGIN
  Filename = files(i)
  PRINT, ' Working with >', filename
;
;
;
;
  IF ( STRPOS( FileName, 'LE01' ) EQ -1 ) THEN BEGIN
    IF ( KEYWORD_SET(AIIDDF) ) THEN BEGIN
      IF ( ReadDDF( FileName ) ) THEN BEGIN
        PRINT, ' Success!'
      ENDIF ELSE BEGIN
        PRINT, ' Unable to read >', Filename
      ENDELSE
    ENDIF
  ENDIF ELSE BEGIN
    IF ( ReadDLG( FileName, XRange, YRange ) ) THEN BEGIN
      PRINT, ' Success!'
    ENDIF ELSE BEGIN
      PRINT, ' Unable to read >', Filename
    ENDELSE
  ENDELSE
ENDFOR
;
;
;
;
END

```
