Subject: Re: IDL DICOM writer
Posted by Marcus O'Brien on Fri, 11 Jan 2002 15:51:06 GMT
View Forum Message <> Reply to Message

Hi Bhautik,

Thanks for all the nice comments about TIFF_TO_DICOM.c, if I'd realized anyone would appreciate it I'd have made certain the file comments actually reflected its current usage, DUH! Oh and that I had my current e-mail address on it (m.obrien@sghms.ac.uk).

Love the DICOM_WRITER for IDL, took a little bit of playing for ctn dicom software to open the created files. I believe that the minimum bits per pixel supported by the standard is 16. I've attached a slightly hacked version that sets bpp internally irrespective of the image type coming in. Had played with the idea of allowing floats and setting pixel representation according to image array type, but settled on bytscl and a conversion to uint for now. See what you think.

Trivial bit, physicians name tag is 0008,0090 (some readers will drop the file if the tags are out of order).

I'm sure all the med image people out there that use IDL love you :)

Thanks for an excellent bit of porting, I'd been putting it off for ages.

Marc

--
Tel: 0208 725 2857
Fax: 0208 725 2992

St George's Hospital Medical School
Department of Biochemistry and Immunology
Cranmer Terrace
Tooting
London SW17 0RE

```
;
; NAME:
;  DICOM_WRITER
;
; VERSION:
; 0.11
;
; PURPOSE:
; Generate a dicom file from within RSI IDL
;
```

```
; AUTHOR:
; Bhautik Joshi
;
; EMAIL:
; bjoshi@geocities.com
;
; HOMEPAGE:
; http://cow.mooh.org
;
; USE:
; DICOM_WRITER, filename, image
;
; INPUT:
; filename - string containing name of dicom file to be written to
; bpp - number of _bytes_ per pixel in the image
; image - byte formatted version of image
;
; NOTES ON USAGE:
; * At the moment the program only writes to a single slice
; * The input image must be squashed into a 1D array of bytes before
;   it can be used in dicom_writer
; * bpp specifies the number of bytes (not bits!!) per pixel
; * Extra dicom tags can be easily added (see body of program)
; * There is little to no error-checking at the moment, so be
;   careful!
; * Analyse seems to need a minimum image size of somewhere around
;   100x100
;
; EXAMPLE:
; Create a horrendously boring byte image and store it in a
; dicom file, test.dcm :
;
; > rows = 200
; > cols = 200
; > bpp = 1
; > image = BYTESCL(indgen(rows,cols))
; > dicom_writer, 'test.dcm', bpp, image
;
; HISTORY:
; Based on Marc O'Briens (marcus@sghms.ac.uk) TIFF_to_DICOM.c
; version 0.1  08-01-2002 - first working version produced
; version 0.11 09-01-2002 - fixed endian-ness issue & added get_lun
;                functionality
;
; TODO:
; * Allow for more robust dicom writing
; * Expand the number of tags written (using DICOM data
;   dictionary)
```

```
; * Part 10 compliance (!!!!!!!!!!)
;
; DISCLAIMER:
;
; Permission to use, copy, modify, and distribute this software and its
; documentation for any purpose and without fee is hereby granted,
; provided that the above copyright notice appear in all copies and that
; both that copyright notice and this permission notice appear in
; supporting documentation.
;
; This file is provided AS IS with no warranties of any kind.  The author
; shall have no liability with respect to the infringement of copyrights,
; trade secrets or any patents by this file or any part thereof.  In no
; event will the author be liable for any lost revenue or profits or
; other special, indirect and consequential damages.
;
; The author accepts no responsibility for any action arising from use of
; this package. The software is not guaranteed to write compliant DICOM
; files. If it causes damage to you or your system, you have been warned -
; this is a work in progress. If it bites your dog, its not my fault. If
; it causes you to curl up on the floor in the foetal position muttering
; about pixies and mushrooms, its not my fault. If it causes you or someone
; else to spontaneously burst into song and dance, its not my fault but
; I'd like to hear about it. You have been warned.
;

;convert a value, val, that is num bytes long, into
;a series of ordered bytes
function getbytes, val, num
 ret=BYTARR(num)
 offset=0
;work in big endian ONLY
 ;val=swap_endian(val)
 if (!version.arch eq 'x86') then begin
  byteorder,val,/SWAP_IF_BIG_ENDIAN
 endif else begin
  byteorder,val,/SWAP_IF_LITTLE_ENDIAN
 endelse
 for i=0,(num-1) do begin
  tmpres=BYTE(ISHFT(val, offset) AND 255)
  ret[i]=tmpres
  offset=offset-8
 endfor

 return, ret
end

;write any tag
```

```
function generate_anytag, group, element, data

 rs=[getbytes(group,2),getbytes(element,2)]

;correct to even length if necessary
 bs=BYTE(data)
 nl=n_elements(bs)
 if ((nl mod 2) ne 0) then begin
  bs=[bs,BYTE(0)]
  nl=nl+1
 end
;size of field
 rs=[rs,getbytes(nl,2)]
;padding
 rs=[rs,[0,0]]
;string itself
 rs=[rs,bs]

 return, rs
end

;generate string tag
function generate_stringtag, group, element, string
 return, generate_anytag(group, element, BYTE(string))
end

;generate binary element (unsigned short) tag
function generate_UStag, group, element, val
 param=getbytes(val,2)
 return, generate_anytag(group,element,param)
end

;generate pixel tag
function generate_pixeltag, group, element, val
 return, [getbytes(group,2),getbytes(element,2), getbytes(val,4)]
end

;generate unsigned long tag
function generate_ULtag, group, element, val
 param=getbytes(val,4)
 return, generate_anytag(group,element,param)
end


pro dicom_writer, filename, image
 ; bytescl and convert input image to 16bit unsigned integer
 sz=size(image)
 image = reform(UINT(BYTSCL(image)),sz[1],sz[2])
```

```
;dummy fill-in variables.
 random= '123456'
 SOPClass = '1.2.840.10008.5.1.4.1.1.20'
 SOPInstance = '1.2.840.10008.5.1.4.1.1.20.1'
 StudyID = '1.2.3.4'
 StudyInstanceUID = SOPInstance + random
 SeriesInstanceUID = StudyInstanceUID
 RelFrameOfReferenceUID = StudyInstanceUID
 SeriesInstanceUID = SeriesInstanceUID + '.1'
 RelFrameOfReferenceUID = RelFrameOfReferenceUID + '.2'
 StudyID = StudyID + 'SIGNA '

;image variables
 Seriesnum=0
 Imagenum=0
 thickness=1.0
 spacing='1.0\\1.0'
 rows=sz[1]
 cols=sz[2]
 bpp=2

 GET_LUN, U

 OPENW, U, filename

; DICOM tags - feel free to add more!

;0008 tags

;MR type
 WRITEU, U,  BYTE(generate_stringtag('0008'x,'0008'x,'ORIGINAL\\PRIMARY\\ OTHER'))
;Instance date
 WRITEU, U, BYTE(generate_stringtag('0008'x,'0012'x,'20020108'))
;Instance time
 WRITEU, U, BYTE(generate_stringtag('0008'x,'0013'x,'000000.00000'))
;SOP class
 WRITEU, U, BYTE(generate_stringtag('0008'x,'0016'x,SOPClass))
;SOP instance
 WRITEU, U, BYTE(generate_stringtag('0008'x,'0018'x,SOPInstance))
;Modality
 WRITEU, U, BYTE(generate_stringtag('0008'x,'0060'x,'MR'))
;Manufacturer
 WRITEU, U, BYTE(generate_stringtag('0008'x,'0070'x,'GE'))
;Study Physicians Name
 WRITEU, U, BYTE(generate_stringtag('0008'x,'0090'x,'fizzy'))

;0010 tags
```

```
;Patient name
 WRITEU, U, BYTE(generate_stringtag('0010'x,'0010'x,'A Patient'))
;Patient ID
 WRITEU, U, BYTE(generate_stringtag('0010'x,'0020'x,'TK247'))
;Patient birth date
 WRITEU, U, BYTE(generate_stringtag('0010'x,'0030'x,'20020111'))
;Patient sex
 WRITEU, U, BYTE(generate_stringtag('0010'x,'0040'x,'F'))


;0018 tags

;Acquisition type
 WRITEU, U, BYTE(generate_stringtag('0018'x,'0023'x,'2D'))
;Slice thickness
 WRITEU, U, BYTE(generate_stringtag('0018'x,'0050'x,STRING(thickness)))

;0020 tags

;Study instance
 WRITEU, U, BYTE(generate_stringtag('0020'x,'000D'x,StudyInstanceUID))
;Series instance UID
 WRITEU, U, BYTE(generate_stringtag('0020'x,'000E'x,SeriesInstanceUID))
;StudyID
 WRITEU, U, BYTE(generate_stringtag('0020'x,'0010'x,StudyID))
;Series number
 WRITEU, U, BYTE(generate_stringtag('0020'x,'0011'x,STRING(seriesnum)))
;Image number
 WRITEU, U, BYTE(generate_stringtag('0020'x,'0013'x,STRING(imagenum)))

;0028 tags

;samples per pixel
 WRITEU, U, BYTE(generate_UStag('0028'x,'0002'x,1))
;Photometric interpretation
 WRITEU, U, BYTE(generate_stringtag('0028'x,'0004'x,'MONOCHROME2'))
;Rows in image
 WRITEU, U, BYTE(generate_UStag('0028'x,'0010'x,rows))
;Columns in image
 WRITEU, U, BYTE(generate_UStag('0028'x,'0011'x,cols))
;pixel spacing
 WRITEU, U, BYTE(generate_stringtag('0028'x,'0030'x,spacing))
;bits allocated per sample
 WRITEU, U, BYTE(generate_UStag('0028'x,'0100'x,bpp*8))
;bits stored per sample
 WRITEU, U, BYTE(generate_UStag('0028'x,'0101'x,bpp*8))
;high bit
```

```
 WRITEU, U, BYTE(generate_UStag('0028'x,'0102'x,(bpp*8)-1))
;pixel representation
 WRITEU, U, BYTE(generate_UStag('0028'x,'0103'x,'0000'x))

;write image data
 imsize=rows*cols*bpp


 WRITEU, U, BYTE(generate_pixeltag('7FE0'x,'0010'x,imsize))
 WRITEU, U, image

 CLOSE, U
 FREE_LUN, U
end
```

File Attachments

1) dicom_writer.pro, downloaded 93 times