Subject: GDLffDICOM
Posted by Robert Barnett on Wed, 24 Aug 2005 06:01:04 GMT
View Forum Message <> Reply to Message

Hi,

I just thought I might let you know about my recent implementation of GDLffDICOM

It is currently available from
http://www.zipworld.com.au/~retsil/idl/gdlffdicom0.1.1.tar.g z
http://www.zipworld.com.au/~retsil/idl/gdlffdicom0.1.1.zip
but will be available from
http://www.anzsnm.org.au/ftp/idl/
sometime in the near future.

GDLffDICOM is an adaptation of IDLffDICOM to achieve several outcomes.

* Better runtime error handling
* Ability to read a broader range of DICOM files
* Limited ability to write DICOM files
* Drop in replacement of IDLffDICOM

Better runtime handling

GDLffDICOM is written completely in IDL. Whilst this decreases performance, it also ensures that all runtime errors are not fatal and can be captured. This is advantageous over IDLffDICOM, which is written in C/C++ and can cause IDL to terminate abnormally if an error is encountered.

Ability to read a broader range of DICOM files

GDLffDICOM can read a slightly borader range of DICOM files than IDLffDICOM. GDLffDICOM is more lenient about accepting non-compliant tags. You can also avoid further parsing errors by using the GDLffDIOM__assoc class.

Limited ability to write DICOM files

GDLffDICOM allows you to associate the image data of a DICOM file with an IDL variable. This provides a reasonably efficient method of altering the pixel data in a DICOM file. Other DICOM tags can be set via the GDLffDICOM::SetValue method and are written to a DICOM file using the GDLffDICOM::Commit function.

Drop-in replacement of IDLffDICOM

The GDLffDICOM method signature is almost identical to IDLffDICOM. This allows you to avoid significant alteration of code if you wish to only use a small portion of GDLffDICOMs features.

Authors Note

GDLffDICOM has acquired its name from the GNU Data Language (GDL) project. Whilst GDLffDICOM might work with GDL it may not exclusively work on either platform. It has only been named with the GDL prefix so it can be effectively maintained by both the IDL and GDL community without namespace issues.
