# DICOM

**Digital Imaging and Communications in Medicine** (**DICOM**) is a standard for handling, storing, printing, and [transmitting](http://en.wikipedia.org/wiki/Data_transmission) information in [medical imaging](http://en.wikipedia.org/wiki/Medical_imaging). It includes a [file format](http://en.wikipedia.org/wiki/File_format) definition and a network [communications protocol](http://en.wikipedia.org/wiki/Communications_protocol). The communication protocol is an application protocol that uses [TCP/IP](http://en.wikipedia.org/wiki/TCP/IP) to communicate between systems. DICOM files can be exchanged between two entities that are capable of receiving image and patient data in DICOM format.

DICOM is known as [NEMA](http://en.wikipedia.org/wiki/National_Electrical_Manufacturers_Association) standard PS3, and as [ISO standard](http://en.wikipedia.org/wiki/ISO_standard) 12052:2006 "Health informatics -- Digital imaging and communication in medicine (DICOM) including workflow and data management".

## python

<https://pyscience.wordpress.com/2014/09/08/dicom-in-python-importing-medical-image-data-into-numpy-with-pydicom-and-vtk/>

I’ll be showing how to use the pydicom package and/or VTK to read a series of DICOM images into a NumPy array. This will involve reading metadata from the DICOM files and the pixel-data itself.

Anyone in the medical image processing or diagnostic imaging field, will have undoubtedly dealt with the infamous [Digital Imaging and Communications in Medicine (DICOM) standard](http://en.wikipedia.org/wiki/DICOM) the de-facto solution to storing and exchanging medical image-data.

Applications such as [RadiAnt](http://www.radiantviewer.com/) or [MicroDicom](http://www.microdicom.com/) for Windows and [OsiriX](http://www.osirix-viewer.com) for Mac, do a great job of dealing with DICOM files. However, there are as many flavors of DICOM as there are of ice-cream. Thus, when it comes to programmatically reading and processing DICOM files things get a little hairier depending on whether the files store the pixel data in a compressed form or not.

### GDCM

**Grassroots DICOM** or **GDCM** (originally called **GNU DiCoM**; the name was changed at a request for integration in [ITK](http://en.wikipedia.org/wiki/Insight_Segmentation_and_Registration_Toolkit), followed by a change in license), is a [cross-platform](http://en.wikipedia.org/wiki/Cross-platform) library written in C++ for [DICOM](http://en.wikipedia.org/wiki/Digital_imaging_and_communications_in_medicine) medical files. It is automatically wrapped to Python/C#/Java & PHP (using [SWIG](http://en.wikipedia.org/wiki/SWIG)). It supports RAW, JPEG (lossy/lossless), J2K, JPEG-LS, RLE and deflated. It also comes with DICOM Part 3,6 & 7 of the standard as XML files. It can be used to build a JPIP or WADO server.

<http://gdcm.sourceforge.net/wiki/index.php/Main_Page>

### pydicom

<https://github.com/darcymason/pydicom>

pydicom is a pure python package for working with [DICOM](http://medical.nema.org/) files. It was made for inspecting and modifying DICOM data in an easy "pythonic" way. The modifications can be written again to a new file.

As a pure python package, pydicom can run anywhere python runs without any other requirements, although [NumPy](http://www.numpy.org) is needed if manipulating pixel data.

pydicom is not a DICOM server, and is not primarily about viewing images. It is designed to let you manipulate data elements in DICOM files with python code.

Limitations -- the main limitation of the current version is that *compressed* pixel data (e.g. JPEG) cannot be altered in an intelligent way as it can for uncompressed pixels. Files can always be read and saved, but compressed pixel data cannot easily be modified.

### pynetdicom

*Pure python implementation of the DICOM network protocol*

<https://pypi.python.org/pypi/pynetdicom/0.8.0>

pynetdicom is a pure python package implementing the DICOM network protocol. Working with pydicom, it allows DICOM clients (SCUs) and servers (SCPs) to be easily created. DICOM is a standard (<http://medical.nema.org>) for communicating medical images and related information such as reports and radiotherapy objects.

The main class is AE and represents an application entity. User typically create an AE object, specifying the SOP service class supported as SCP and SCU, and a port to listen to. The user then starts the application entity which runs in a thread. The use can initiate associations as SCU or respond to remote SCU association by the means of callbacks.

# Liver segmentation, 3D reconstruction

## 3D Splicer

<https://www.orthanc-server.com/>

## Ginkgo CADx

**Ginkgo CADx** is a multiplatform (Windows, Linux,[[1]](http://en.wikipedia.org/wiki/Ginkgo_CADx#cite_note-1) Mac OS X) DICOM viewer (\*.dcm) and dicomizer (convert different files to DICOM). Ginkgo CADx is licensed under [LGPL](http://en.wikipedia.org/wiki/LGPL) license, being an [open source](http://en.wikipedia.org/wiki/Open-source_software) project with an [Open core](http://en.wikipedia.org/wiki/Open_core) approach. The goal of Ginkgo CADx project is to develop an open source professional [DICOM](http://en.wikipedia.org/wiki/DICOM) workstation.[[2]](http://en.wikipedia.org/wiki/Ginkgo_CADx#cite_note-2)

<http://ginkgo-cadx.com/en/>

## OsiriX

Mac Dicom viewer

http://www.osirix-viewer.com/index.html

# Visualization

## VTK - Visualization Toolkit

VTK is an open-source software system for image processing, 3D graphics, volume rendering and visualization. VTK includes many advanced algorithms (e.g., surface reconstruction, implicit modelling, decimation) and rendering techniques (e.g., hardware-accelerated volume rendering, LOD control).

general visualization toolkit for 3D data

http://www.vtk.org/VTK/resources/software.html

**X11\_Xt\_LIB could not be found**

**sudo apt-get install libxt-dev doxygen**

Building with cmake-gui

<https://mail.python.org/pipermail/pythonmac-sig/2004-October/011838.html>

necessary to activate the python wrappings in cmake

Change to the Python wrapper directory and install the Python wrappers.

copy  
Wrapping/Python

to

export PYTHONPATH=${PYTHONPATH}:/usr/local/lib/python2.7/site-packages/vtkWrapper

# Python paths

export PYTHONPATH=${PYTHONPATH}:/usr/local/lib/python2.7/site-packages/vtk

export PYTHONPATH=${PYTHONPATH}:/usr/local/lib/python2.7/site-packages/vtkWrapper

export PYTHONPATH=${PYTHONPATH}:/usr/local/lib/python2.7/site-packages/vtkWrapper/vtk

export PYTHONPATH=${PYTHONPATH}:/usr/local/bin

## Documentation

<http://www.vtk.org/doc/nightly/html/>

<http://www.barre.nom.fr/vtk/links.html>