**dcmtk程序包简介**  
\*\*\*该文简单列出了dcmtk程序包的简介，包括主要接口类的简单说明，可用工具以及一些例子。下一步工作准备详细分析每个程序包中的接口类功能，并结合源码和dicom文档分析其实现过程。

Dcmtk程序包简介

一、Config程序包

-config目录下的文档：

--config.txt:

指出你编辑的任何.h .c .cc文件首先必须包含该目录下的头文件#include "osconfig.h"

--dirstruc.txt:

给出了dcmtk项目的项目目录结构，这个用cmake会自动生成

--envvars.txt:

这个文件比较重要，它指出了一些运行时环境变量，这些变量可能会影响dcmtk的工具和库的使用，这些变量包括：

---DCMDICTPATH:影响dcmdata

    On Win32 platforms, a built-in dictionary is used by default. If

    the DCMDICTPATH environment variable is set, the applications will

    attempt to load \_additional\_ DICOM data dictionaries specified in

    the DCMDICTPATH environment variable instead. The DCMDICTPATH

    environment variable has the same format as the shell PATH variable

    in that a semicolon (";") separates entries. The data dictionary

    code will attempt to load each file specified in the DCMDICTPATH

    environment variable.

    See also: documentation in dcmdata/docs/datadict.txt

---TCP\_BUFFER\_LENGTH:影响dcmnet

    By default, DCMTK uses a TCP send and receive buffer

    length of 32K. If the environment variable TCP\_BUFFER\_LENGTH is set,

    it specified an override for the TCP buffer length. The value is

    specified in bytes, not in Kbytes.

---TCP\_NODELAY:影响dcmnet

    If this environment variable contains a non-zero number,

    the Nagle algorithm will not be disabled for TCP transport

    connections. Also see documentation for macro

    DONT\_DISABLE\_NAGLE\_ALGORITHM in config/docs/macros.txt

---TMPDIR:影响dcmnet

    Affects the implementation of the tempnam() emulation on

    platforms where tempnam() is not defined. See tempnam(3S) main page

    for a description.

--macros.txt:

这个文件也特别重要，它给出了很多编译时的宏，这些宏可能会影响dcmtk的工具和库的使用。大部分的宏可以用来激活一些实验性的或很少需要的特性，另外有一些是用来取消某些功能。要尽量谨慎使用。详细见文档。

--modules.txt:

这个文件讲述如何自己配置各个模块，不需要掌握。

-config的include目录下的文件

--osconfig.h：这个文件是必须包含在所有.h .c文件中的

其中指出在win32环境下包含"dcmtk/config/cfwin32.h"文件

--cfwin32.h:包含了大量的宏定义。\*\*\*如果需要查找某个宏的定义，可到这个文件中查找\*\*\*

二、ofstd程序包

-ofstd:作为一般目的的类库。

这个模块包含了一般目的的类库，这些类所描述的对象概念并非在Dicom标准中特有。它们广泛的在toolkit中使用。主要包含下面的类

--OFCommandLine：处理命令行参数，头文件在ofcmdln.h。\*\*\*详情需要结合具体的代码来理解\*\*\*

--OFCondition：描述条件码的一般类。头文件在ofcond.h。\*\*\*详情需要结合具体的代码来理解\*\*\*

--OFConsole:是一个singleton(孤立)类\*\*\*不明白\*\*\*。提供线程安全的对标准输出流和错误流的访问。允许以多线程的方式同时创建输出。\*\*\*多线程的东西不太明白，它的作用应该是在多线程中实现指定线程的输出流进行输出操作\*\*\*

--OFList:是一个双向链表模板类，接口是STL list类中的一个子集。头文件在oflist.h。\*\*\*不明白为什么不直接用标准库中的list，兼容性应该更好啊\*\*\*

--OFStack：是一个堆栈模板类，接口是STL stack类中的一个子集。头文件在ofstack.h。\*\*\*不明白为什么不直接用标准库中的stack，兼容性应该更好啊\*\*\*

--OFStandard：包含大量帮助函数组成的类，用来包含大量“全局”帮助函数。注意全部都是静态函数。其中的一些函数实现调用了windows API函数，如fileexists()。头文件在ofstd.h。\*\*\*如果需要一些全局函数，可以到这里了找一找。\*\*\*

--OFString:一个简单的string类，实现了std::string的一个子集，没有iterator或trait，在速度上也没有优化。头文件在ofstring.h。\*\*\*不明白为什么不直接用标准库中的string，兼容性应该更好啊\*\*\*

--除了上述的主要类以外，还包含了大量辅助类，用来支撑上述类的功能。\*\*\*详情需要结合具体的代码来理解\*\*\*

三、dcmdata程序包

-dcmdata:一个数据编码/解码库和可用的工具

这个模块包含了一些类来管理Dicom数据结构和文件。同时它也提供了对DICOMDIR文件的支持以满足Dicom storage media（存储介质）的需要。

----------------------------------------------------

主要的接口类有：

--DcmFileFormat：a class handling the DICOM file format (with meta header) 。头文件在dcfilefo.h。\*\*\*详情在单独的文档中分析\*\*\*

--DcmDataset：a class handling the DICOM dataset format (files without meta header) 。头文件在dcdatset.h。\*\*\*详情在单独的文档中分析\*\*\*

--DcmItem：a class representing a collection of DICOM elements。头文件在dcitem.h。\*\*\*详情在单独的文档中分析\*\*\*

--DcmElement：abstract base class for all DICOM elements。头文件在dcelem.h。\*\*\*详情在单独的文档中分析\*\*\*。它的派生类包括：DcmAttributeTag\DcmByteString\DcmFloatingPointDouble\DcmFloatingPointSingle\DcmOtherByteOtherWord\DcmSequenceOfItems\DcmSignedLong\DcmSignedShort\DcmUnsignedLong\DcmUnsignedShort

----------------------------------------------------

----------------------------------------------------

工具：这个模块包含了下面的命令行工具：

dcm2xml: Convert DICOM file and data set to XML

dcmconv: Convert DICOM file encoding

dcmcrle: Encode DICOM file to RLE transfer syntax

dcmdrle: Decode RLE-compressed DICOM file

dcmdump: Dump DICOM file and data set

dcmftest: Test if file uses DICOM part 10 format

dcmgpdir: Create a general purpose DICOMDIR

dcmodify: Modify DICOM files

dump2dcm: Convert ASCII dump to DICOM file

xml2dcm: Convert XML document to DICOM file or data set

\*\*\*暂时不对命令行工具进行详细的分析\*\*\*

----------------------------------------------------

---------------------------------------------------

举例:

--调入一个DICOM文件，输出病人姓名

DcmFileFormat fileformat;

OFCondition status = fileformat.loadFile("test.dcm");

if (status.good())

{

OFString patientsName;

if (fileformat.getDataset()->findAndGetOFString(DCM\_PatientsName, patientsName).good())

{

    cout << "Patient's Name: " << patientsName << endl;

} else

    cerr << "Error: cannot access Patient's Name!" << endl;

} else

cerr << "Error: cannot read DICOM file (" << status.text() << ")" << endl;

--创建一个DICOM dataset数据集，并保存为文件

char uid[100];

DcmFileFormat fileformat;

DcmDataset \*dataset = fileformat.getDataset();

dataset->putAndInsertString(DCM\_SOPClassUID, UID\_SecondaryCaptureImageStorage);

dataset->putAndInsertString(DCM\_SOPInstanceUID, dcmGenerateUniqueIdentifier(uid, SITE\_INSTANCE\_UID\_ROOT));

dataset->putAndInsertString(DCM\_PatientsName, "Doe^John");

/\* ... \*/

dataset->putAndInsertUint8Array(DCM\_PixelData, pixelData, pixelLength);

OFCondition status = fileformat.saveFile("test.dcm", EXS\_LittleEndianExplicit);

if (status.bad())

cerr << "Error: cannot write DICOM file (" << status.text() << ")" << endl;

--如何为多个文件创建一般目的的DICOMDIR

DicomDirInterface dicomdir;

OFCondition status = dicomdir.createNewDicomDir();

if (status.good())

{

while ( /\* there are files \*/ )

    dicomdir.addDicomFile( /\* current filename \*/ );

status = dicomdir.writeDicomDir();

if (status.bad())

    cerr << "Error: cannot write DICOMDIR (" << status.text() << ")" << endl;

} else

cerr << "Error: cannot create DICOMDIR (" << status.text() << ")" << endl;

------------------------------------------------------------

四、dcmimgle程序包

dcmimgle是一个图像处理库和可用的工具模块，它包括了对DICOM单色图像的访问和显示。对颜色图像的支持由dcmimage模块提供，对JPEG压缩图像的支持由dcmjpeg模块支持。

主要接口类：

--DicomImage: 为dcmimgle/dcmimage模块提供接口类。主要目的是图像显示。在dcmimage.h中定义。

--DiDisplayFunction: Class to handle hardcopy and softcopy device characteristics file and manage display LUTs (for calibration). 在didispfn.h中定义。  
  
可用工具：

--dcmdspfn: Export standard display curves to a text file

--dcod2lum: Convert hardcopy characteristic curve file to softcopy format

--dconvlum: Convert VeriLUM files to DCMTK display files

举例：

--载入一幅DICOM单帧单色图像，并显示其像素数据。

DicomImage \*image = new DicomImage("test.dcm");

if (image != NULL)

{

if (image->getStatus() == EIS\_Normal)

{

    if (image->isMonochrome())

    {

      image->setMinMaxWindow();

      Uint8 \*pixelData = (Uint8 \*)(image->getOutputData(8 /\* bits \*/));

      if (pixelData != NULL)

      {

        /\* do something useful with the pixel data \*/

      }

    }

} else

    cerr << "Error: cannot load DICOM image (" << DicomImage::getString(image->getStatus()) << ")" << endl;

}

delete image;

五、dcmimage程序包

dcmimage模块为dcmimgle模块提供对彩色图像的支持。对单色图像的支持由dcmimgle提供，对JPEG压缩图像的支持由dcmjpeg模块支持。

主要接口类：

--DicomImage: 在dcmimgle中已介绍。

工具：

--dcm2pnm: Convert DICOM images to PPM/PGM, PNG, TIFF or BMP

--dcmquant: Convert DICOM color images to palette color

--dcmscale: Scale DICOM images

举例：

--载入一幅DICOM单帧图像（单色或彩色），并显示其像素数据。

#include "diregist.h"   /\* required to support color images \*/

/\* ... \*/

DicomImage \*image = new DicomImage("test.dcm");

if (image != NULL)

{

if (image->getStatus() == EIS\_Normal)

{

    Uint8 \*pixelData = (Uint8 \*)(image->getOutputData(8 /\* bits per sample \*/));

    if (pixelData != NULL)

    {

      /\* do something useful with the pixel data \*/

    }

} else

    cerr << "Error: cannot load DICOM image (" << DicomImage::getString(image->getStatus()) << ")" << endl;

}

delete image;

六、dcmjpeg程序包

dcmjpeg提供了一个压缩/解压缩库以及可用工具。该模块包含一些类，可将DICOM图像对象在非压缩和JPEG压缩表示（传输协议）之间转换。无失真和有失真JPEG处理都被支持。这个模块实现了一族codec(编码解码器，由DcmCodec类派生而来)，可以将这些codec在codec list中注册，codec list是由dcmdata模块保存的。

主要接口类：

--DJEncoderRegistration: 一个singleton（孤立）类，为所有支持的JPEG处理注册编码器。在djencode.h中定义。

--DJDecoderRegistration: 一个singleton（孤立）类，为所有支持的JPEG处理注册解码器。在djdecode.h中定义。

--DJCodecEncoder: JPEG编码器的一个抽象codec类。This abstract class contains most of the application logic needed for a dcmdata codec object that implements a JPEG encoder using the DJEncoder interface to the underlying JPEG implementation. This class only supports compression, it neither implements decoding nor transcoding. 在djcodece.h中定义。

--DJCodecDecoder: JPEG解码器的一个抽象codec类。This abstract class contains most of the application logic needed for a dcmdata codec object that implements a JPEG decoder using the DJDecoder interface to the underlying JPEG implementation. This class only supports decompression, it neither implements encoding nor transcoding.

工具：

dcmcjpeg: Encode DICOM file to JPEG transfer syntax

dcmdjpeg: Decode JPEG-compressed DICOM file

dcmj2pnm: Convert DICOM images to PGM, PPM, BMP, TIFF or JPEG

**dcmmkdir**: Create a DICOMDIR file

举例：

--用无失真JPEG压缩一幅DICOM图像文件。

DJEncoderRegistration::registerCodecs(); // register JPEG codecs

DcmFileFormat fileformat;

if (fileformat.loadFile("test.dcm").good())

{

DcmDataset \*dataset = fileformat.getDataset();

DcmItem \*metaInfo = fileformat.getMetaInfo();

DJ\_RPLossless params; // codec parameters, we use the defaults

// this causes the lossless JPEG version of the dataset to be created

dataset->chooseRepresentation(EXS\_JPEGProcess14SV1TransferSyntax, &params);

// check if everything went well

if (dataset->canWriteXfer(EXS\_JPEGProcess14SV1TransferSyntax))

{

    // force the meta-header UIDs to be re-generated when storing the file

    // since the UIDs in the data set may have changed

    delete metaInfo->remove(DCM\_MediaStorageSOPClassUID);

    delete metaInfo->remove(DCM\_MediaStorageSOPInstanceUID);

    // store in lossless JPEG format

    fileformat.saveFile("test\_jpeg.dcm", EXS\_JPEGProcess14SV1TransferSyntax);

}

}

DJEncoderRegistration::cleanup(); // deregister JPEG codecs

--解压缩一幅JPEG压缩的DICOM图像文件。

DJDecoderRegistration::registerCodecs(); // register JPEG codecs

DcmFileFormat fileformat;

if (fileformat.loadFile("test\_jpeg.dcm").good())

{

DcmDataset \*dataset = fileformat.getDataset();

// decompress data set if compressed

dataset->chooseRepresentation(EXS\_LittleEndianExplicit, NULL);

// check if everything went well

if (dataset->canWriteXfer(EXS\_LittleEndianExplicit))

{

    fileformat.saveFile("test\_decompressed.dcm", EXS\_LittleEndianExplicit);

}

}      
DJDecoderRegistration::cleanup(); // deregister JPEG codecs

七、dcmnet程序包

dcmnet是一个网络库及可用工具。该模块包含了实现DICOM网络通信的所有函数集，即：DICOM上层有限状态机(DICOM Upper Layer Finite State Machine)，关联控制服务元素（Association Control Service Element, ACSE）以及DICOM消息服务元素（DICOM Message Service Element, DIMSE）。

主要接口：该模块的主要接口包括在文件assoc.h和dimse.h中定义的大量结构和函数。

--assoc.h: 这个文件包含程序，为DICOM应用提供关联管理。它维护描述活动关联的结构，提供对关联特定信息的访问。也提供程序帮助关联协议association negotiation(presentation contexts, abstract syntaxes, transfer syntaxes, maximum PDU length, and other extended negotiation)。该包使用了DICOM上层机制接收/发送关联请求/响应。每一个活动的关联由T\_ASC\_Association结构表示，包含了所有相关的信息。模块前缀ASC\_。

--dimse.h: 这个文件包含程序，为DICOM应用提供dimse层的服务。

工具：

--echoscu: DICOM verification (C-ECHO) SCU

--findscu: DICOM query (C-FIND) SCU

--movescu: DICOM retrieve (C-MOVE) SCU

--storescp: DICOM storage (C-STORE) SCP

--storescu: DICOM storage (C-STORE) SCU

--termscu: DICOM termination SCU

举例：

--一个简单的Echo SCU(Verification Service Class SCU)。大多数错误处理代码省去了，在Win32上的特定代码如WinSock初始化也省去了。

T\_ASC\_Network \*net; // network struct, contains DICOM upper layer FSM etc.

ASC\_initializeNetwork(NET\_REQUESTOR, 0, 1000 /\* timeout \*/, &net);

T\_ASC\_Parameters \*params; // parameters of association request

ASC\_createAssociationParameters(&params, ASC\_DEFAULTMAXPDU);

// set calling and called AE titles

ASC\_setAPTitles(params, "ECHOSCU", "ANY-SCP", NULL);

// the DICOM server accepts connections at server.nowhere.com port 104

ASC\_setPresentationAddresses(params, "localhost", "server.nowhere.com:104");

// list of transfer syntaxes, only a single entry here

const char\* ts[] = { UID\_LittleEndianImplicitTransferSyntax };

// add presentation context to association request

ASC\_addPresentationContext(params, 1, UID\_VerificationSOPClass, ts, 1);

// request DICOM association

T\_ASC\_Association \*assoc;

if (ASC\_requestAssociation(net, params, &assoc).good())

{

if (ASC\_countAcceptedPresentationContexts(params) == 1)

{

    // the remote SCP has accepted the Verification Service Class

    DIC\_US // generate next message ID

    DIC\_US status; // DIMSE status of C-ECHO-RSP will be stored here

    DcmDataset \*sd = NULL; // status detail will be stored here

    // send C-ECHO-RQ and handle response

    DIMSE\_echoUser(assoc, id, DIMSE\_BLOCKING, 0, &status, &sd);

    delete sd; // we don't care about status detail

}

}

ASC\_releaseAssociation(assoc); // release association

ASC\_destroyAssociation(&assoc); // delete assoc structure

ASC\_dropNetwork(&net); // delete net structure

八、dcmpstat程序包

dcmpstat: 一个描述状态(presentation state)库和可用工具。This module contains classes that implement a high-level API for the DICOM Softcopy Grayscale Presentation State Storage SOP Class. It also contains various support classes that are used by DICOMscope, a free DICOM viewer that has been developed as a demonstrator for presentation states. See <http://dicom.offis.de/dscope>。

主要接口：

--DVPresentationState: 一个灰度软拷贝描述状态。这个类管理着一个描述状态对象的数据结构。描述状态可以创建、读、写和更改。在dvpstat.h中定义。

--DVInterface: 这个接口类是用来帮助软拷贝描述状态浏览器工作的。这个类管理着数据库机制，允许开始和停止网络交互，并访问图像和描述状态。在dviface.h中定义。

--DVPSStoredPrint: the representation of a Stored Print object。在文件dvpssp.h中定义。

工具：

dcmmkcrv: Add 2D curve data to image

dcmmklut: Create DICOM look-up tables

dcmp2pgm: Read DICOM image and presentation state and render bitmap

dcmprscp: DICOM basic grayscale print management SCP

dcmprscu: Print spooler for presentation state viewer

dcmpschk: Checking tool for presentation states

dcmpsmk: Create DICOM grayscale softcopy presentation state

dcmpsprt: Read DICOM images and presentation states and render print job

dcmpsrcv: Network receive for presentation state viewer

dcmpssnd: Network send for presentation state viewer

举例：

--给一幅DICOM图像创建一个缺省的描述状态

DcmFileFormat infile;

DcmFileFormat outfile;

if (infile.loadFile("image.dcm").good())

{

DVPresentationState pstate; // presentation state handler

if (pstate.createFromImage(\*infile.getDataset()).good())

{

    // serialize presentation state into DICOM data set structure

    if (pstate.write(\*outfile.getDataset(), OFFalse).good())

    {

      // and write to file

      outfile.saveFile("gsps.dcm", EXS\_LittleEndianExplicit);

    }

}

}

--应用一个描述状态中的灰度变换管道给一幅DICOM图像

DcmFileFormat imagefile;

DcmFileFormat gspsfile;

if (imagefile.loadFile("image.dcm").good() &&

    gspsfile.loadFile("gsps.dcm").good())

{

DVPresentationState pstate; // presentation state handler

if (pstate.read(\*gspsfile.getDataset()).good()) // parse gsps object

{

    // attach presentation state to image data

    if (pstate.attachImage(&imagefile, OFFalse).good())

    {

      const void \*pixel; // pointer to pixel data, one byte per pixel

      unsigned long width;   // width of image bitmap

      unsigned long height; // height of image bitmap

      if (pstate.getPixelData(pixel, width, height).good())

      {

        /\* do something useful with the pixel data \*/

      }

      pstate.detachImage(); // release connection between GSPS and image

    }

}

}

九、dcmsign程序包

dcmsign是一个数字签名库和可用工具。这个模块包含了一些类，以创建DICOM数据集中的数字签名，并验证和删除签名。这个模块需要扩展的OpenSSL库的支持。

主要接口：  
--DcmSignature: this class provides the main interface to the dcmsign module - it allows to create, examine and verify digital signatures in DICOM datasets or items. The methods in this class do not handle digital signatures embedded in sequence items within the dataset, other than providing helper functions that allow to locate and attach the sub-items separately. 在dcsignat.h中定义。

--SiSecurityProfile: 所有安全框架的抽象基类。abstract base class for all security profiles. 在sisprof.h文件中定义。

--SiCertificate: a class representing X.509 public key certificates. 在sicert.h文件中定义。

--SiPrivateKey: a class representing a private key. 在siprivat.h文件中定义。

--SiMAC: a base class for all classes that implement hash functions. 在simac.h文件中定义。

工具：

dcmsign: Sign and Verify DICOM Files

举例：

--验证一个DICOM文件中的所有签名。

DcmFileFormat fileformat;

if (fileformat.loadFile("test.dcm").good())

{

int counter = 0;          // counts the signatures in the DICOM file

int corrupt\_counter = 0; // counts signatures that failed verification

DcmDataset \*dataset = fileformat.getDataset();

DcmStack stack;           // stores current location within file

DcmSignature signer;      // signature handler

DcmItem \*sigItem = DcmSignature::findFirstSignatureItem(\*dataset, stack);

while (sigItem) // browse through items that contain digital signatures

{

    signer.attach(sigItem); // each item may contain multiple signatures

    for (unsigned long l=0; l < signer.numberOfSignatures(); ++l)

    {

      if (signer.selectSignature(l).good())

      {

        ++counter;  
        if (signer.verifyCurrent().bad()) // verify signature  
           corrupt\_counter++;  
      }  
    }  
    signer.detach();  
    sigItem = DcmSignature::findNextSignatureItem(\*dataset, stack);  
}  
if (counter == 0)   
      cerr << "no signatures found in dataset." << endl;  
else   
      cerr << counter << " signatures verified in dataset, "   
           << corrupt\_counter << " corrupted." << endl;  
}  
--给一个DICOM文件增加签名。  
DcmFileFormat fileformat;  
if (fileformat.loadFile("test.dcm").good())

{  
DcmDataset \*dataset = fileformat.getDataset();  
SiCreatorProfile profile; // select the "RSA Creator Profile"  
SiRIPEMD160 mac;           // use RIPEMD160 as MAC algorithm  
DcmSignature signer;       // signature handler  
SiCertificate cert;        // our certificate  
if (cert.loadCertificate("certificate.pem", X509\_FILETYPE\_PEM).bad())  
{  
    cerr << "unable to load certificate" << endl;  
    return;  
}  
SiPrivateKey key; // private key, must be unencrypted here  
if (key.loadPrivateKey("privkey.pem", X509\_FILETYPE\_PEM).bad())  
{  
    cerr << "unable to load private key" << endl;  
    return;  
}  
signer.attach(dataset); // connect handler to data set  
if (signer.createSignature(key, cert, mac, profile).good())  
{  
    fileformat.saveFile("test\_signed.dcm"); // write back  
}  
}  
十、dcmsr程序包  
dcmsr是一个结构化报表库和可用工具。这个模块包括一些类来读、写、创建、修改、访问、打印和显示DICOM结构化报表文档。所支持的SOP类列表由DSRTypes::E\_DocumentType提供。

主要接口：  
--DSRDocument: Interface class for 'dcmsr' (DICOM Structured Reporting Documents). This class supports reading, writing, creation, printing and rendering of DICOM SR documents (according to DICOM PS 3.x-2004, formerly known as Supplement 23). The list of supported SOP classes is available in file "dsrtypes.h". 在dsrdoc.h中定义。  
--DSRDocumentTree: 管理SR文档树的类。在dsrdoctr.h中定义。  
--DSRContentItem: Interface class for content items. This class allows to access the document tree nodes without using any pointers. 在dsrcitem.h中定义。  
--DSRCodedEntryValue: Class for coded entry values. 在dsrcodvl.h中定义。

工具：  
dsr2html: Render DICOM SR file and data set to HTML   
dsr2xml: Convert DICOM SR file and data set to XML   
dsrdump: Dump DICOM SR file and data set   
xml2dsr: Convert DICOM SR file and data set to XML

举例：  
--载入一个DICOM结构化报表，并以HTML格式显示其内容。  
DcmFileFormat fileformat;  
OFCondition status = fileformat.loadFile("test.dcm");  
if (status.good())  
{  
DSRDocument document;  
status = document.read(\*fileformat.getDataset());  
if (status.good())  
{  
    status = document.renderHTML(cout);  
    if (status.bad())  
      cerr << "Error: cannot render SR document (" << status.text() << ")" << endl;  
} else  
    cerr << "Error: cannot parse SR document (" << status.text() << ")" << endl;  
} else  
cerr << "Error: cannot read DICOM file (" << status.text() << ")" << endl;  
--创建一个DICOM结构化报告，并将其存为文件。  
DSRDocument document;  
document.setPatientsName("Doe^John");  
/\* ... \*/  
document.getTree().addContentItem(DSRTypes::RT\_isRoot, DSRTypes::VT\_Container);  
document.getTree().getCurrentContentItem().setConceptName(DSRCodedEntryValue(/\* some code \*/));  
document.getTree().addContentItem(DSRTypes::RT\_hasObsContext, DSRTypes::VT\_Code, DSRTypes::AM\_belowCurrent);  
/\* ... \*/  
DcmFileFormat fileformat;  
OFCondition status = document.write(\*fileformat.getDataset())  
if (status.good())  
{  
status = fileformat.saveFile("test.dcm", EXS\_LittleEndianExplicit);  
if (status.bad())  
    cerr << "Error: cannot write DICOM file (" << status.text() << ")" << endl;  
} else  
cerr << "Error: cannot write SR document (" << status.text() << ")" << endl;  
--读取文档树的属性，并直接修改。  
DSRDocument document(DSRTypes::DT\_KeyObjectDoc);  
/\* ... \*/  
document.getTree().addContentItem(DSRTypes::RT\_isRoot, DSRTypes::VT\_Container);  
DSRCodedEntryValue \*codePtr = document.getTree().getCurrentContentItem().getConceptNamePtr();  
if (codePtr != NULL)  
codePtr->setCode("113000", "DCM", "Of Interest");  
/\* ... \*/  
document.getTree().addContentItem(DSRTypes::RT\_contains, DSRTypes::VT\_Image);  
DSRImageReferenceValue \*imagePtr = document.getTree().getCurrentContentItem().getImageReferencePtr();  
if (imagePtr != NULL)  
{  
imagePtr->setValue(DSRImageReferenceValue(UID\_UltrasoundMultiframeImageStorage, /\* image UID \*/));  
imagePtr->setPresentationState(DSRCompositeReferenceValue(UID\_GrayscaleSoftcopyPresentationStateStorage, /\* GSPS UID \*/));  
imagePtr->getFrameList().addItem(2);  
imagePtr->getFrameList().addItem(5);  
}

/\* ... \*/  
十一、dcmtls程序包  
dcmtls是网络库的安全扩展。This module contains classes that implement DICOM network communication tunneled through a Transport Layer Security (TLS) connection, conforming to the DICOM "Security Enhancements One" extension (formerly Supplement 31). This module requires the external OpenSSL library.

主要接口：  
--DcmTLSTransportLayer: factory class which creates secure TLS transport layer connections and maintains the parameters common to all TLS transport connections in one application (e.g. the pool of trusted certificates, the key and certificate to be used for authentication and the list of ciphersuite to be used for association negotiation.)。在tlslayer.h文件中定义。  
--DcmTLSConnection: this class represents a TLS (Transport Layer Security) V1 based secure transport connection.

举例：  
--TLS的关联请求应用  
T\_ASC\_Network \*net;        // network initialization code not shown,  
T\_ASC\_Parameters \*params; // we just assume these pointers to be valid

// create TLS object that initializes the random generator through a file

// "random.dat" containing random data (1 kByte is sufficient).

DcmTLSTransportLayer \*tLayer = new DcmTLSTransportLayer(

DICOM\_APPLICATION\_REQUESTOR, "random.dat");

if (TCS\_ok != tLayer->setPrivateKeyFile("privkey.pem", SSL\_FILETYPE\_PEM))  
{  
cerr << "unable to load private key" << endl;  
return;  
}

if (TCS\_ok != tLayer->setCertificateFile("certificate.pem", SSL\_FILETYPE\_PEM))  
{  
cerr << "unable to load certificate" << endl;  
return;  
}

// enable the TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA ciphersuite

tLayer->setCipherSuites(SSL3\_TXT\_RSA\_DES\_192\_CBC3\_SHA);

// accept any certificate from the remote site (not recommended)

tLayer->setCertificateVerification(DCV\_ignoreCertificate);

// register and activate TLS layer

ASC\_setTransportLayer(net, tLayer, 1);

ASC\_setTransportLayerType(params, 1);

十二、dcmwlm程序包

dcmwlm是一个设备工作表(Modality Worklist)数据库服务器。这个模块包含类，作为一个SCP，为实现DICOM Modality Worklist Management Service的应用提供支持。基于这些类的SCP可以从C-Find-RSP返回消息中找到相关信息。

主要接口：

--WlmActivityManager: This class encapsulates data structures and operations for basic worklist management service class providers. 在wlmactmg.h文件中定义。

--WlmDataSource: This class encapsulates data structures and operations for connecting to an arbitrary data source in the framework of the DICOM basic worklist management service. 在wlds.h文件中定义。

--WlmDataSourceFileSystem: This class encapsulates data structures and operations for connecting to a file-based data source in the framework of the DICOM basic worklist management service. 在wldsfs.h文件中定义。

--WlmFileSystemInteractionManager: This class encapsulates data structures and operations for managing data base interaction in the framework of the DICOM basic worklist management service. 在wlfsim.h文件中定义。

工具：

wlmscpfs: DICOM Basic Worklist Management SCP (based on data files)

举例：

For an example of how to use the main interface classes of this module, see file 'wlmscpfs.cc' (containing the main function of the corresponding tool) and file 'wlcefs.cc' (making use of the WlmActivityManager class that manages all activities a corresponding SCP has to manage).

十三、dcmqrdb程序包

dcmqrdb是一个图像数据库服务器。This module contains a simple image archive that manages a number of storage areas and allows images to be stored in these storage areas using the DICOM Storage Service Class. It also allows image attributes to be queried and images to be retrieved using the DICOM Query/Retrieve Service Class.

工具：

dcmqridx: Register a DICOM image file in an image database index file

dcmqrscp: DICOM image archive (central test node)

dcmqrti: The Terminal Initiator Telnet Client Program

文件：下面文件提供进一步信息。

--dcmqrcnf.txt: