**MINT Reference Implementation**

**Installation Guide**

**Necessary Software:**

**MINT Client**

* Eclipse IDE for Java Developers (<http://www.eclipse.org/downloads/>)
* Java JDK 1.6 or newer (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
* SVN Client (TortoiseSVN recommended - <http://tortoisesvn.net/downloads>)
* Apache Ant (<http://ant.apache.org/>)

**MINT Server**

* Apache Tomcat 6 Server (<http://tomcat.apache.org/download-60.cgi>)
* Optional:
  1. Eclipse IDE for Java Developers (<http://www.eclipse.org/downloads/>)
  2. Java JDK 1.6 or later (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
  3. SVN Client (TortoiseSVN recommended - <http://tortoisesvn.net/downloads>)
  4. Apache Ant (<http://ant.apache.org/>)

**MINT Conformance Tests**

* Python 2.5 or later (<http://www.python.org/download/>)
* Cygwin (<http://www.cygwin.com/>)

**MINT ClearCanvas Workstation Plugin**

* Versioning Client (TortoiseSVN recommended - <http://tortoisesvn.net/downloads>)
* Microsoft Visual Studio 2008 Professional

**Configuration**

1. Checkout source package using the information at the following URL into a directory:
   1. <http://code.google.com/p/medical-imaging-network-transport/source/checkout>
2. Launch *Eclipse*.
3. Right-click on *Project Explorer* and click *Import*.
4. Navigate to the source checkout directory from step (1) and load MINTCommon. Repeat this step for MINTMetadata, MINTServer, MINT2DICOM and DICOM2MINTImport directories.
5. Install Apache Ant on the computer. Follow the URL below for more information on how to install Apache Ant on the computer. Make sure to add ant into the path.
   1. <http://ant.apache.org/manual/install.html>

**Server Setup**

* First step in creating the server is to create the .war file that will be deployed on the Apache Tomcat 6 server. When the command "ant" is used to build the project, the .war file is automatically generated under *MINTServer\build* directory. Alternatively, Eclipse IDE can be used to export a .war file by navigating to *File->Export->Web->WAR File* , though it is **HIGHLY** recommended to use ant to build the project. Both approaches generate *MINTServer.war* file by default, it is not recommended to change the name of the server file for simplicity purposes.
* Next, Apache Tomcat 6 needs to be setup. Follow the instructions on

<http://tomcat.apache.org/tomcat-6.0-doc/setup.html>

for more information on how to setup and start Tomcat. Make sure that the conf/tomcat-users file has been modified to add an administrator account (This is a **VERY** important step, Tomcat manager **cannot** be accessed without an administrator account). It is recommended that the Java Virtual Machine Heap Size is increased to *~ 2048MB*. Once Tomcat is set up, use a web browser to navigate to

<http://localhost:8080/>

and click on Manager to get to the Tomcat Manager page, login using the administrator account credentials. The manager can now be used to deploy the .war file generated, by using "WAR File to deploy" and by navigating to the .war file.

*Note: The .war file can be generated on the client side and transferred to the server, eliminating the need for a duplicate setup on the server. Optionally, the projects can be set up on the server and can generate the .war files locally.*

* Once the .war file is deployed, the Tomcat 6 Manager should display it as one of the running applications. The validity of the server deploy can be verified by navigating to

<http://localhost:8080/MINTServer>

which should display the MINT Server Main Menu, providing options such as

* + View [Studies](http://localhost:8080/MINTServer/studies)
  + View [Changelog](http://localhost:8080/MINTServer/changelog)
  + View [Data Dictionary](http://localhost:8080/MINTServer/types)
  + Study [Create](http://localhost:8080/MINTServer/jobs/createstudy)
  + Study [Update](http://localhost:8080/MINTServer/jobs/updatestudy)
  + *View Studies* link provides a web page with a brief summary of each study uploaded onto the Server. If there are no studies present, only a header of "Studies" is displayed.
  + *View Changelog* link provides a web page with a brief summary of the updates performed on each of the studies.
  + *View Data Dictionary* link provides an XML representation of DICOM tag representations, with VR values and descriptions.
  + *Study Create* can be used to upload a MINT study manually to the Server, by providing paths to the metadata and each of the binary items.
  + *Study Upload* is essentially the same as Study Create, except the Study UUID needs to be provided.

*Note: Apache Tomcat 6 Server uses port 8080. This port may need to be exposed if the server is on a VPN (not on localhost).*

**DICOM to MINT Conversion**

The *DICOM2MINTImport* project provides means to traverse through a directory of DICOM files and creates/updates studies on the listed MINTServer. This project is recommended to be built using "ant" in order to set the appropriate Java heap sizes.

The project runtime requires a number of arguments to be provided by the user, in the following sequence:

1. once|daemon - Specifies if the directory should be traversed once or as a daemon process.
2. xml|gpb - Specifies the DICOM metadata output format.
3. {PATH TO DIRECTORY} - The path to the directory of DICOM files.
4. {URL} - The URL path to the MINTServer.
5. binThreshold - An optional, non-negative integer the specifies the minimum size of a DICOM tag that should be stored in an external file rather than inline in the metadata file.
6. nodelete - An optional argument that specifies to not delete the DICOM files converted.
7. forcecreate - An optional argument that specifies to never update existing studies.

The project provides feedback to the user of the program, indicating if a study already exists on the server and whether it is being updated, how many files have been generated, etc.

**MINT to DICOM Conversion**

The *MINT2DICOM* project provides means to convert a MINT study on a MINT Server to the DICOM standard. This project currently does not have ant build, and can simply be run through Eclipse.

To run this project:

* Click on *Run Configurations* (next to Run).
* Select *Java Application* and click *New.*
* On *Main* tab, select *MINT2DICOM* as the project.
* If the *Main class* has not automatically been found, search and select *MINT2DICOM* (or type org.nema.medical.mint.MINT2DICOM).
* This project has arguments that need to be provided by the user:
  + {URL} - The URL to the study on the server, ending with /DICOM/.

(e.g. [http://10.181.5.201:8080/MINTServer/studies/95f976dd-d6b6-46a6-a1b5- 4f62375d0556/DICOM/](http://10.181.5.201:8080/MINTServer/studies/95f976dd-d6b6-46a6-a1b5-4f62375d0556/DICOM/))

* + {PATH TO DIRECTORY} - The output directory where the DICOM files will be output to.
  + true|false - Indicate whether bulk loading is to be used.

**Conformance Testing**

**Setting up the environment:**

The MINT Conformance tools require [Python2.5](http://www.python.org) or later. Currently the tools are run directly from the MINTConformance package in a subversion trunk (installation scripts will be provided later). In order to run the tools, the **PYTHONPATH** environment variable must be set to point to the python source code. The python application must also be in your **PATH**. For example, if **MC\_PATH** is the path to your subversion trunk, then set your PYTHONPATH as follows:

**Unix**

PATH=${PATH}:/pub/python/bin  
 PYTHONPATH=$MC\_HOME/MINTConformance/src

**Windows**

set PATH=%PATH%;C:\Python2.6  
 set PYTHONPATH=%MC\_HOME%\MINTConformance\src

**Conformance Tools**

**MINT Study Compare Shell Script**

The [MintStudyCompare.sh](http://code.google.com/p/medical-imaging-network-transport/source/browse/trunk/MINTConformance/src/org/nema/medical/mint/MintStudyCompare.sh) tool is a simple shell script that compares two MINT studies by running the Unix *diff* command on the metadata.xml files and the *cmp* command on each of the binary items. This tool checks that the MINT studies are exactly equal.

Because we are running directly from the subversion trunk instead on an installation, the permissions on the tools are not executable. Launch the tool by using the Unix *sh* command to parse the file. To run this script on Windows, it requires a [cyqwin](http://www.cygwin.com/) command shell.

Running the tool with no arguments displays the usage:

> sh $MC\_HOME/MINTConformance/src/org/nema/medical/mint/MintStudyCompare.sh  
Usage: MintStudyCompare.sh <mint\_study1.xml> <mint\_study2.xml>

To test the tool, you can use the sample data located in the MINTConformance/testdata directory. In the example below, we run the tool with the identical MINT studies so no differences are found. As with most Unix applications, no news is good news. Therefore, no output means the metadata and binary files are exactly equal.

> cd $MC\_HOME/MINTConformance/testdata/MINT/7cc45edf-d5a6-4264-b8ae-1d24d857f04b/DICOM  
> sh $MC\_HOME/MINTConformance/src/org/nema/medical/mint/MintStudyCompare.sh metadata.xml metadata.xml

To see differences, we can compare an older MINT study to a newer one. The example below shows both metadata changes and binary differences.

> cd $MC\_HOME/MINTConformance/testdata/MINT  
> sh $MC\_HOME/MINTConformance/src/org/nema/medical/mint/MintStudyCompare.sh metadata.1.xml 7cc45edf-d5a6-4264-b8ae-1d24d857f04b/DICOM/metadata.xml  
1,190c1,190  
< <?xml version="1.0" encoding="utf-8" ?>  
< <StudyMeta studyInstanceUID="1.2.392.200036.9116.2.2.2.1762893313.1029997326.945873">  
<     <Attributes>  
<         <Attr tag="00080020" vr="DA" val="20020711" />  
<         <Attr tag="00080030" vr="TM" val="144114.000" />  
<         <Attr tag="00080050" vr="SH" val="1126" />  
...  
./binaryitems/2.dat 7cc45edf-d5a6-4264-b8ae-1d24d857f04b/DICOM/binaryitems/2.dat differ: byte 759, line 4

**MINT Study Compare Python Application**

The [MintStudyCompare.py](http://code.google.com/p/medical-imaging-network-transport/source/browse/trunk/MINTConformance/src/org/nema/medical/mint/MintStudyCompare.py) tool is a smart python application that compares two MINT studies by parsing the metadata.xml files and doing a byte by byte comparison for each of the binary items. This tool checks that the MINT studies are equivalent and is tolerant to minor formatting differences and attribute sorting.

Launch the tool by using the *python* command to parse the file. This tool should run either in a Unix or Windows command shell.

Running the tool with no arguments displays the usage:

> python $MC\_HOME/MINTConformance/src/org/nema/medical/mint/MintStudyCompare.py  
Usage: MintStudyCompare.py [options] <mint\_study1.xml> <mint\_study2.xml>  
  -v: verbose  
  -l: lazy check (skips binary content)  
  -h: displays usage

The verbose flag turns on output to show how many tags and bytes were compared. Otherwise nothing is displayed when there are no differences. You can optionally skip checking the binary to save time when debugging.

> cd $MC\_HOME/MINTConformance/testdata/MINT/7cc45edf-d5a6-4264-b8ae-1d24d857f04b/DICOM  
> python $MC\_HOME/MINTConformance/src/org/nema/medical/mint/MintStudyCompare.py -v metadata.xml metadata.xml  
        12 study attribute(s) compared.  
         2 series compared.  
        14 series attribute(s) compared.  
       102 normalized instance attribute(s) compared.  
         4 instance(s) compared.  
        24 instance attribute(s) compared.  
         4 binary item(s) compared.  
   2097152 byte(s) compared.

To see all the differences the application should be run twice, switching the command line arguments. This will find all the tags that may be present in one study but not the other.

> cd $MC\_HOME/MINTConformance/testdata/MINT  
> python $MC\_HOME/MINTConformance/src/org/nema/medical/mint/MintStudyCompare.py 7cc45edf-d5a6-4264-b8ae-1d24d857f04b/DICOM/metadata.xml metadata.2.xml  
Series (1.2.392.200036.9116.2.2.2.1762893313.1029997326.946350) Normalized Instance Attribute : tag=00080008 vr=CS val=ORIGINAL\PRIMARY\AXIAL != tag=00080008 vr=CS val=ORIGINAL  
Series (1.2.392.200036.9116.2.2.2.1762893313.1029997326.946350) Normalized Instance Attribute : tag=00200020 vr=CS val=L\P != tag=00200020 vr=CS val=L  
Series (1.2.392.200036.9116.2.2.2.1762893313.1029997326.946350) Normalized Instance Attribute : tag=00200037 vr=DS val=1.000000\0.000000\0.000000\0.000000\1.000000\0.000000 != tag=00200037 vr=DS val=1.000000  
...  
2.dat size : 524288 != 524287  
        12 study attribute(s) compared.  
         2 series compared.  
        14 series attribute(s) compared.  
       102 normalized instance attribute(s) compared.  
         4 instance(s) compared.  
        24 instance attribute(s) compared.  
         4 binary item(s) compared.  
   1572864 byte(s) compared.  
17 difference(s) found.

**MINT DICOM Compare Python Application**

The [MintDicomCompare.py](http://code.google.com/p/medical-imaging-network-transport/source/browse/trunk/MINTConformance/src/org/nema/medical/mint/MintDicomCompare.py) tool is a smart python application that compares a DICOM study to a MINT study by parsing the data elements in the DICOM and comparing them to the metadata and binary items in the MINT study.

There are some subtle differences between the DICOM and MINT representations if you are using another tool for comparison. Some binary VRs are represented as text in MINT for clarity ("SS", "US", "SL", "UL", "FL", "FD", "OF", "AT") so the values will not compare directly to the DICOM binaries. Arrays of these primitive types are represented as strings with the values separated by a "\" delimiter (ie. 208.55017\166.19896). Note that the precision for floating point representations is set to 6.

Launch the tool by using the *python* command to parse the file. This tool should run either in a Unix or Windows command shell.

Running the tool with no arguments displays the usage:

> python $MC\_HOME/MINTConformance/src/org/nema/medical/mint/MintDicomCompare.py  
Usage: MintDicomCompare.py [options] <dicom\_study\_dir> <mint\_study.xml> <data\_dictionary.xml>  
  -v: verbose  
  -l: lazy check (skips binary content)  
  -h: displays usage

The verbose flag turns on output to show how many tags and bytes were compared. Otherwise nothing is displayed when there are no differences. You can optionally skip checking the binary to save time when debugging. The data dictionary is required for looking up implicit VRs based on the group-element tag.

> cd $MC\_HOME/MINTConformance/testdata  
> python $MC\_HOME/MINTConformance/src/org/nema/medical/mint/MintDicomCompare.py -v DICOM MINT/7cc45edf-d5a6-4264-b8ae-1d24d857f04b/DICOM/metadata.xml DICOMDictionary.xml  
- Study Instance UID 1.2.392.200036.9116.2.2.2.1762893313.1029997326.945873  
 - Series Instance UID 1.2.392.200036.9116.2.2.2.1762893313.1029997326.946350  
  - SOP Instance UID 1.2.392.200036.9116.2.2.2.1762660474.1026398161.357037  
+++ 00280002 Value : != 1  
- Study Instance UID 1.2.392.200036.9116.2.2.2.1762893313.1029997326.945873  
 - Series Instance UID 1.2.392.200036.9116.2.2.2.1762893313.1029997326.946350  
  - SOP Instance UID 1.2.392.200036.9116.2.2.2.1762660474.1026398161.357037  
+++ 00280006 Value :  != 0  
- Study Instance UID 1.2.392.200036.9116.2.2.2.1762893313.1029997326.945873  
 - Series Instance UID 1.2.392.200036.9116.2.2.2.1762893313.1029997326.946350  
  - SOP Instance UID 1.2.392.200036.9116.2.2.2.1762660474.1026398161.357037  
+++ 00280010 Value :   
...  
         4 instance(s) compared.  
        32 tag(s) skipped.  
       300 text tag(s) compared.  
         4 binary tag(s) compared.  
   2097152 byte(s) compared.  
        32 difference(s) found.

**MINT ClearCanvas Workstation Plug-in**

ClearCanvas Workstation is a DICOM Image Viewing Solution. The plug-in was chosen to be developed for ClearCanvas Workstation as it is an open source application, and is intended to demonstrate how an existing application can consume the information provided by MINT.

**Downloading the Source and Building**

1. Checkout the reference MINT source from the following repo: <http://medical-imaging-network-transport.googlecode.com/svn/trunk>.  
      - For the remainder of this instruction, {MINTROOT} refers to the local directory the MINT source was checked out into.
2. Checkout the ClearCanvas source. The SVN repository is: **svn://svn.clearcanvas.ca/source/Xian/Tags/RisViewer2.0SP1**  
   For the remainder of this instruction, {CCROOT} refers to the local directory the ClearCanvas source was checked out into. For authentication, Username = "opensource", Password = "opensource".
3. Apply the patch found in the *{MINTROOT}/ClearCanvas/****ClearCanvasPatch*** from the checked-out MINT source to *{CCROOT}*. This will make the necessary changes to core ClearCanvas to allow the MINT plugin to work. The patch can be applied using *TortoiseSVN*.
4. Copy the MINTLoader and MINTUnitTest directories from the *{MINTROOT}/ClearCanvas/****ClearCanvasPlugin*** into *{CCROOT}*.
5. Open *{CCROOT}/ImageViewer/ImageViewer.sln* in Visual Studio 2008 (other versions are currently not supported).
6. Add the *{CCROOT}/MINTLoader/MINTLoader.csproj* project to the solution.
7. Copy the files in *{MINTROOT}/ClearCanvasCodec/* into *{CCROOT}/ReferencedAssemblies/Codecs*, replacing all files.
8. If a security question pops up asking "Open project for Browsing / Open project Normally", select "Open Project Normally" and deselect "Ask for every project".
9. Build the solution.

**Running ClearCanvas**

1. Make sure a *MINT Server* with loaded studies is running.
2. Make the *ClearCanvas.Desktop.Executable* project the *Startup Project* and *run*. When it runs, a server will be displayed on the *Servers List* on the left called "**Medical Imaging Network Transfer**".
3. Right-click and edit as follows:
   * **General/Server Name:** Medical Imaging Network Transfer
   * **General/Host:** Enter the URI of the MINT Server (e.g. http://{hostname}[:port]/MINTServer)
   * **General/Location:** leave empty.
   * **General/Finder Name:** MINT
   * **DICOM/AE Title:** MINT
   * **DICOM/Port:** ignored, use a non-zero number.
   * **ClearCanvas Image Streaming/Enabled:** uncheck.
   * Click **OK** to save the settings.
4. Select the "Medical Imaging Network Transfer" Server. Click "Search" (near the top, right of center). Click "Yes" on the warning.
5. A list of studies will show up. Select one and click the right-most green checkbox icon on the studies list. This will load the study.

*Note: The ClearCanvas plugin is able to use bulk loading or single file loading. To change between these modes, the plugin code needs to be modified. Specifically,* ***MINTLoader\MINTStudyLoader.cs*** *program's* ***OnStart*** *method needs to be modified, where if* ***UseBulkLoading*** *is set to* ***true****,* ***bulk loading*** *is used, and when it is set to* ***false****,* ***single file loading*** *is used.*