DICOMatic 2.0
Conversion Module

User’s Guide

TomoVision

division of
VIRTUAL MAGIC INC.
MONTREAL, CANADA

DICOMatic 2.0: Conversion Module USER'S GUIDE
rev-09/09/08

TomoVision
4559 Pontiac
Montreal, Qc
Canada, H2J 2T2

For questions regarding DICOMatic, phone or fax (514) 522-3559 during Eastern standard time business hours, or email support@tomovision.com.

Information in this document is subject to change without notice and does not represent a commitment on the part of TomoVision or Virtual Magic Inc. It is against the law to copy this software in any fashion or any medium.

TomoVision is a division of Virtual Magic Inc.

DISCLAIMER:

Good data processing procedure dictates that any program be thoroughly tested with non-critical data before relying on it. The user must assume the entire risk of using the program.

Virtual Magic Inc. makes no representation or warranty with respect to the adequacy of readOmatic for any particular purpose or with respect to its adequacy to produce any particular results. In no events shall Virtual Magic Inc., its employees or its suppliers be liable for special, direct, indirect or consequential damages, losses, costs, changes, demands, or claims for lost profits, fees, or expenses of any nature.

The PVRG-JPEG library disclaimer:

The TIFF library disclaimer:

The JasPer JPEG library disclaimer:

Table of Contents

1 Introduction

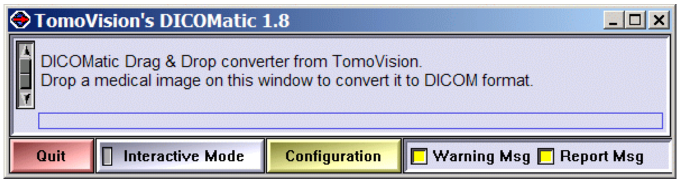
With DICOMatic, you can convert the medical images produced by older scanners to the newer DICOM format.

DICOMatic uses the TomoVision_Convert.dll library to read an image of any format and write it back in a DICOM format.

DICOMatic can be used in two ways:

As a “Drag & Drop” program

You select the icons of the images you want to convert, drag these icons to the DICOMatic window and drop them there.



DICOMatic Drag & Drop converter from TomoVision.
Drop a medical image on this window to convert it to DICOM format.

As a batch converter

You can use DICOMatic in a DOS batch or you can call it from your own application. The command line arguments of DICOMatic enable you to control its operation.

The licenses

Each time DICOMatic convert an image, it consume a “token”. These token are stored in a license “dongle” (USB key). Without tokens, DICOMatic will nevertheless convert your images and generate the DICOM files, but:
- it will also add a watermark to the images
- the Patient Name will be “Demo DICOMatic”
- the Patient ID and Study ID will be “0".

All the other DICOM tags will be created and you should be able to fully evaluate if the conversion process answers your needs.

2 Installation

2.1 First Install

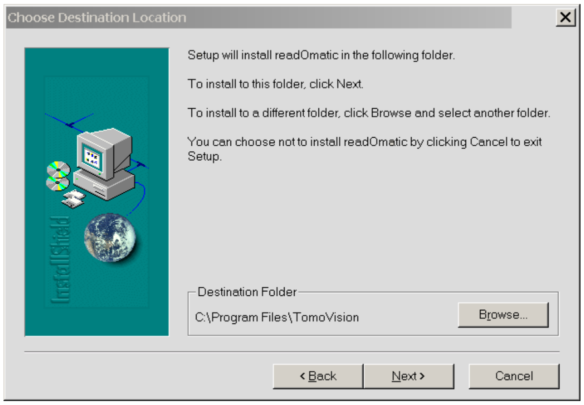
If you have the program's CD, just inserting the CD in the drive should start the installation script. In the absence of the CD, you can install the software directly from the download page at the Web site **http://www.TomoVision.com**.

Note:

To install the software from the Web pages, you will need to have the "WinZip" software on your system. "WinZip" can be found at **http://www.winzip.com**.

To install the software directly from WinZip, click on the "Install" button in the WinZip interface. Alternatively, you can uncompress the files from the ".zip" archive to a temporary directory and click on the "setup.exe" icon to start the installation program.

By default, DICOMatic will be installed in the C:\Program Files\Tomovision directory. If you want the program to be installed somewhere else, you will have the option to select the desired directory during installation. You should not change the program's location after the installation.



Choose Destination Location

Setup will install readOmatic in the following folder.

To install to this folder, click Next.

To install to a different folder, click Browse and select another folder.

You can choose not to install readOmatic by clicking Cancel to exit Setup.

Destination Folder

C:\Program Files\TomoVision

2.2 Updates

New versions of the program are regularly placed on the web site. You should visit the web site periodically to make sure you have the latest version.

To install a newer version of the software, follow the "first Install" instructions, and install the new version in the same directory as the old one. This will overwrite the old program with the new one. You do not need to uninstall the old version.

2.2.1 New DICOM Tables

When DICOMatic is installed, it creates a set of tables in the "DICOM Tables" sub-directory of the installed directory (by default: C:\Program Files\Tomovision). The first time DICOMatic is started, it copies this sub-directory to the user's "Documents ans Settings" directory (on XP: C:\Documents and Settings*user_name*\Application Data\TomoVision, where *user_name* is the current user). If the updated version of the software included an updated version of the tables, you will receive a warning when you start DICOMatic that the tables in the user's directory are out of date. Simply deleting the "DICOM Tables" sub-directory in the user's "Documents and Settings" directory will force the program to re-copy the tables from the installed directory and resolve this issue.

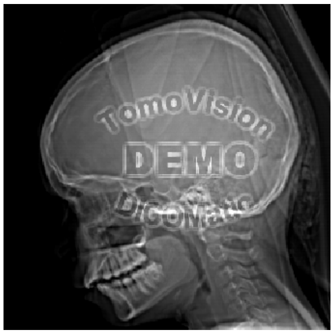
2.2.2 New Conversion Rules file

DICOMatic uses a set of rules, contained in the replacement rule file. By default, this file is called "conversion_rules.txt" and is present in the installed directory (by default: C:\Program Files\Tomovision). If the user has used the "config" option to edit these rules, then a modified copy will also exist in the user's "Documents and Settings" directory (on XP: C:\Documents and Settings*user_name*\Application Data\TomoVision, where *user_name* is the current user). When you start DICOMatic, if the updated version of the software included an updated version of the rules file, you will receive a warning that the rule file in the user's directory are out of date. Simply deleting the file in the user's "Documents ans Settings" directory will force the program to use the rules from the installed directory and resolve this issue.

2.3 Uninstall

To uninstall the software, use the Windows "Add/Remove Programs" application in the normal manner.

3 The Licenses



TomoVision products use 2 layers of protections. A hardware layer, provided by the HASP (Hardware Against Software Piracy) dongles from Aladdin, and a software layer consisting of encrypted license activation codes and a token counter stored in the dongle's memory.

3.1 The HASP dongle

A dongle is a hardware device connected to the USB port. The dongle used by TomoVision has its own timer and memory. All the information needed to license your products is included in the dongle, so it's easy to move your application to a new computer.

If the dongle is not present on your computer, or if its drivers are not installed, the software will run in "demo" mode and some functions will not be available.

HASP dongle for th e USB port

3.2 The License Tokens

DICOMatic licenses work with tokens. Each action of the program will consume tokens. You buy these tokens from TomoVision. The token count is stored in a counter in the dongle's memory. When all the tokens are consumed, and the token count reaches zero, the software will only run in "demo" mode.

You can buy additional tokens and add them to the dongle at any time. TomoVision will generate a license code containing your tokens and send it to you by email. You then use the TomoVision_License program to add these tokens to the dongle's counter.

The licenses codes are specific to a dongle and are only valid once. Using them in TomoVision_License on an another dongle or more than once for their intended dongle will have no effect.
There are 2 types of Tokens, the **30 days tokens** and the **Permanent tokens**.

3.2.1 The 30 days tokens

The dongles are always shipped with 30 days tokens. The reasons for this are twofold:

- As soon as we get a P.O., we ship the dongle with the 30 days tokens. This enables you to start working with the product immediately while the payment is handled by the accounting department of your institution.
- If the dongle gets lost in transit, we can replace it without worrying about multiple licenses being issued, since the tokens in the lost dongle will expire in 30 days anyway.

Also, when you order additional tokens, we will first send you a license code for 30 days tokens. Once we receive payment for your order, we will send you the license code for the permanent tokens by email.

As the name implies, 30 days token are only valid for 30 days. The 30 days are controlled by an internal clock in the dongle and are independent of your computer’s clock. The 30 days are counted from the moment the license was generated at our office.

3.2.2 The Permanent tokens

The license codes for the Permanent tokens are sent by email and you must use the TomoVision_License program to add them to the dongle’s counter.

When you add Permanent tokens to the dongle’s counter, any 30 days tokens that where used will be subtracted from the Permanent token total.

WARNING:

> The permanent tokens must be added to the dongle’s counter with the TomoVision_License program within 30 days of their creation at our office. After that deadline, the TomoVision_License program will reject them and you will have to contact TomoVision for replacement codes.

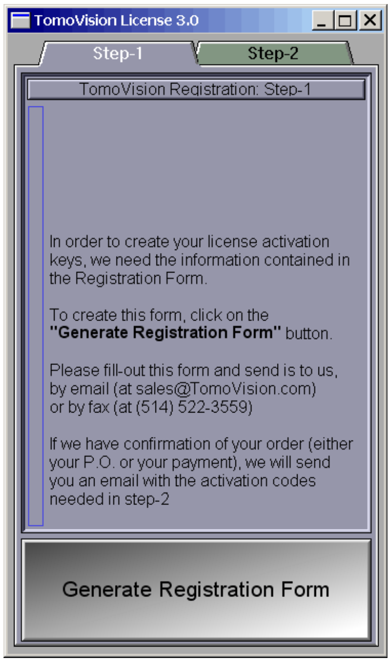
3.3 The TomoVision_License program

The TomoVision_License program is used to read and write license codes in the HASP dongle and to generate a registration form if needed. Each dongle has an unique ID number. This unique ID is used by TomoVision when the license codes are generated, so the licenses are only valid for a specific dongle.

The TomoVision_License program is installed automatically along with your TomoVision application. But you can also download it directly from the “Tools and drivers for the license dongle” page in the Download section of TomoVision’s web site.

Start the TomoVision_License program from the **start\program\TomoVision** menu.

3.3.1 Generating the Registration Form



TomoVision Registration: Step-1

In order to create your license activation keys, we need the information contained in the Registration Form.

To create this form, click on the **"Generate Registration Form"** button.

Please fill-out this form and send is to us, by email (at sales@TomoVision.com) or by fax (at (514) 522-3559)

If we have confirmation of your order (either your P.O. or your payment), we will send you an email with the activation codes needed in step-2

Unless asked for by TomoVision, you do not need to generate a registration form.

If TomoVision asks for it, you should:

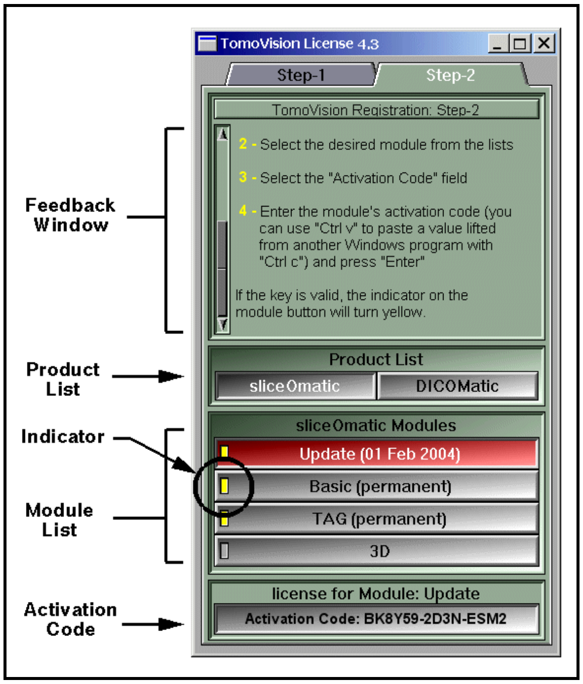
- Make sure the HASP dongle is connected to your computer.
- Start the TomoVision_License program.
- Click on the “Generate Registration Form” button. This will create a registration file in the c:\temp directory. This file will be named: “TomoVision_xxyyyzz_reg.dat”, where xxyyyzz is the current date in the form day/month/year.
- Fill out this form (you can use Windows “Notepad” or any other editor to edit the form).
- Email the form to “sales@TomoVision.com”

3.3.2 Activating the Licenses

The program “TomoVision_License” is used to add the license tokens to the dongle’s counter.

Just drag & drop the license codes file, sent to you by TomoVision, on the License program. All the license codes present in the file will be entered. The indicators on the corresponding buttons in the module list should turn yellow and the counters will be updated.

If the code is invalid or a problem has occurred, a text message will explain the reason in the text feedback window.



TomoVision Registration: Step-2

2 - Select the desired module from the lists

3 - Select the "Activation Code" field

4 - Enter the module's activation code (you can use "Ctrl v" to paste a value lifted from another Windows program with "Ctrl c") and press "Enter"

If the key is valid, the indicator on the module button will turn yellow.

Activation Code: BK8Y59-2D3N-ESM2

3.4 This Module’s token usage

Each image file converted will use one token.

However, if a file contains multiple images, then the module may use more than one token depending on the number of images and their size in pixels. The module will use 1/16 of token for each image of 64x64 pixels or smaller, ¼ of token for images between 64x64 and 128x128 pixels and 1 token each for bigger images. The resulting count will be rounded up to the next integer.

4 How does it work?

The conversion process in DICOMatic is done in 7 steps:

4.1 Identifying the image’s format

Before being able to read an image we need to know its format. Usually, medical images can be divided into 2 categories: fixed format images and ID/value pairs structure images.

4.1.1 Fixed Format

For fixed format images, the image’s header is of fixed length and each information in the header can be found at a precise location. To read the header of these images correctly, we need to know the location of all the important information. Fixed format images are fairly easy to identify: we look for specific byte values at specific locations. For example, 664 bytes after the beginning of the file, a Siemens Somatom Plus image will have the string “SOMATOM PLUS”.

4.1.2 ID / Value pairs

In an ID/value pairs format image, the location of the information is not fixed but is associated with an ID. To read these images correctly, we need to know the ID of all the important information. Most of these files will have “Magic words”, a string of easy to identify symbols, at or near the beginning of the file. For instance, a TIFF file will start with either “II” or “MM”. Otherwise, to identify these files, we need to read one or more of the ID/value pairs and make sure that they make sense.

4.1.3 File name

Only as a last resort, if the image format has not been recognized, will the program look at the file’s name for a clue. The program may also look at the name of other files in the same directory. For example, one of the Optimed format has files whose name start with “XA” and also has a file named “ARCHIVE.PAT” in the same directory.

4.1.4 Multiple images / Multiple files

Usually each original image file will be converted into one DICOM image file. However in some cases, the information of the original image is spread over multiple files or inversely, multiple images can be stored in a single file.

In the case of multiple images in a single file, if the modality supports it, the program will try to keep this structure in the output DICOM file, otherwise the program will create one DICOM file for each image (The “split_multiframe” parameter in the “.ini” configuration file can be used to force the program to split multi-images files.)

If the information necessary to create the output DICOM image is spread over multiple source files, then the program will attempt to find them. In some cases, this implies that a certain directory structure must be present. For instance, in a typical Bruker Paravision conversion, the program will look for the files “d3proc”, “proc” and “reco” in the directory of the source image as well as the “acqp” and “imnd” files in the directory 2 level up and the “subject” file in a directory 3 level up!

Usually the only way the program has of finding the associated files is with the file names, so if the files have been renamed, the program will not find them.

Note:

If you plan to use DICOMatic from your own application, you may want to remove the source images once they are converted. To help you find these “used files”, you can ask DICOMatic to save the name of all the files it has used for the conversion. (C.f. the “-c” argument in Appendix A)

4.2 Reading the image’s header

The program then reads the header of the source image using the appropriate function from the conversion library.

4.2.1 The Internal Data Structure

The information extracted from the header is used to fill an internal data structure. This data structure will contain information such as the patient name and ID, the exam’s, series’ and image’s numbers, the image metrics (image position and orientation, pixel dimensions...) and every other useful data that can be extracted from the original image. The data structure will be used in later steps to create the DICOM output image.

4.2.2 The special case of NEMA derived images

Before creating the DICOM format, the NEMA group proposed 2 medical image formats: NEMA-1 and NEMA-2 (This is also why the DICOM format is sometimes referred to as DICOM-3). These formats already had the basic structure that will later be used in DICOM. Also, some manufacturers created their own variation on these formats such as: SPI (Siemens and Philips), IS&C (Toshiba and Shimadzu) and Papyrus (University Hospital of Geneva).

If the source image is in a NEMA derived format, in addition to creating the internal data structure, a copy of the source image tags is also kept in memory. Only tags that are currently valid in DICOM are kept. In further discussions we will refer to this list as “the original NEMA tag list”.

4.3 The Conversion Rules

The program will then use the Conversion Rules to fill out any missing information or fix any problems in the internal data structure.

For example, we know that Ge Pace scanners are inverting the acquisition and image numbers. The following rule is then used to restore the proper values:

```
if ( FORMAT == "GE PACE" )
        SWAP ACQUISITION_NUM IMAGE_NUM
```

The Conversion Rules can also be used to create an audit file of the conversions. For example, the following rule will add a description line to the audit file with the date, time and file name for every file converted:

```
if ( 1 )
        AUDIT CURRENT_DATE " " CURRENT_TIME ": file:" FILE_NAME
```

A more detailed discussion of the replacement rule syntax is provided in section 7.

4.4 Reading the image’s pixels

The program then reads the image’s pixel information.

If necessary, the image is reformatted to make it DICOM compatible. For example, some Bruker and Varian MR images can be expressed as 8 or 32 bit values and DICOM only supports 16 bit images for these modalities. Also, DICOMatic does not create images with color tables at this time, so any color table images are converted to RGB images.

4.5 The DICOM stream creation

The next step is the creation of the output DICOM stream.

First we find the modality of the image. Then using this information we read the module table, from the DICOM table directory. This file will have a list of all the attribute tables needed to create the DICOM stream for the corresponding modality. For each of these attribute tables, a matching file is read from the DICOM table directory. Each of these files is a table of DICOM tags. For each of these tags we will have the following information:

- an optional default (=), override (!), append (+) or delete (-) character
- the tag's group
- the tag's element number
- an optional default, override or append value
- the tag's type (1, 1C, 2, 2C or 3)
- an optional tag's "Value Representation" (VR) value
- an optional tag's "Value Multiplicity" (VM) value

If either the VR or VM are absent, they will be fetched from a table internal to the program.

A more detailed discussion of the DICOM tables syntax is provided in section 9

The program will then create a list of all the tags defined for the modality.

An optional "Override file" can also be provided. If it is present, the tags from this file will be added to the list.

DICOMatic will then go through that list, and for each tag in the list, if it is able to find a value for that tag, it will create it in the output image. The value found for each tag will come from:

```
if   an override value exist for this tag, it will be used
else if a value for this tag exist in the original NEMA tag list, it will be used
else if a value for this tag can be derived from the internal data structure, it will be
     used
else if a default value exist for this tag, it will be used.
```

Note:

Some of the original tags will be deleted from the original NEMA tag list if they are present. This will insure that the values for these tags are created from the internal data structure instead of using the values from the original NEMA tags. Otherwise, changes made in the Conversion Rules would not have any effects on these tags. For example, study, series, acquisition and instance numbers will be deleted from the source tags list.

Next, the program will scan the list. It will remove all tags of type=3 that do not have associated values, and give a warning message for all tags of type=1 that do not have associated values.

4.5.1 The private tags

As an option, the program can attempt to keep the private tags from the original image if this image is in a NEMA derived format. However, keeping the private tags is not always possible. Particularly, since DICOMatic creates an “explicit VR” file, the “VR” value of all the private tags must be known. This is the case only if the original image is also “Explicit VR”, or if a table exists giving the VR of the private tags. (See section 8.3).

4.5.2 The DICOM UID

The standard also specifies that each image must have a unique identifier (UID). This UID starts with the code assigned to the creating entity.

The different UIDs assigned to the images by DICOMatic are created from a root UID, the image’s format, the patient’s name & ID, the study’s number, the series number, the acquisition number and the image number.

The different UIDs created by DICOmatic are composed of the following codes:

- Root UID: the value of the root UID can be changed in the DICOMatic.ini file. By default, this is TomoVision’s UID: "1.2.826.0.1.3680043.2.307".
- Code Patient: a code representing the original format, followed by 2 numbers derived from the Patient's name (up to 5 digits) and ID (up to 6 digits).
- Code Study: a code representing the study’s date (up to 4 digits) followed by a number derived from the Study number (up to 6 digits) if present, 1 otherwise.
- Code Series: a number derived from the Series number (up to 3 digits) if present, 1 otherwise.
- Code Image: a number derived from the acquisition, accession, temporal numbers (up to 3 digits) if present, 1 otherwise, followed by a number derived from the image number (up to 4 digits) if present, 1 otherwise.
- Code Frame: the constant 1001

These codes are assembled to form the following UID:

Study Instance UID (0020, 000D)

Root UID + “.” + Code Patient + “.” + Code Study

Series Instance UID (0020, 000E)
Root UID + “.” + Code Patient + “.” + Code Study + “.” + Code Series

Media Storage SOP Instance UID (0002, 0003)
and **SOP Instance UID** (0008, 0018)
Root UID + “.” + Code Patient + “.” + Code Study + “.” + Code Series + “.” + Code Image

Frame of Reference UID (0020, 0052)
Root UID + “.” + Code Patient + “.” + Code Study + “.” + Code frame

Note:

This means that DICOMatic considers that all images in the same study share the same frame of reference.

WARNING:

Since the UIDs in DICOM are limited to 64 characters and the “Code Patient + Code Study + Code Series + Code Image” part of the “SOP Instance UID” can have up to 38 characters, the root UID must not have more than 26 characters. (TomoVision’s UID has 25 characters.)

4.6 Optional edition of the DICOM tags

If the “Interactive” button has been pressed, then an interactive window enabling you to edit the DICOM tags will pop-up. For more information on the interactive tag edition, please refer to section 5.4

4.7 Writing the DICOM file

The program will write the DICOM tags to a DICOM part-10 file. It is called “Part-10" because it respects the part-10 of the DICOM definition.

4.7.1 The File’s name

The ouput file’s name is created by concatenating 3 variables from the internal data structure: the path (FILE_PATH), the name (FILE_NAME) and the extension (FILE_EXT). Unless these have been modified by the rules, the path is the same as the path of the original image, the file's name is the same as the file's name from the original image and the extension is ".dcm".

Example: if the original file is c:\temp\toto.dat, then the converted image will be c:\temp\toto.dcm.

If a file with the same path, name and extension already exists, a counter will be added to the file's name. This counter will increment until an empty slot is found.

Example: converting c:\temp\toto.dat, a second time will produce the file c:\temp\toto (2).dcm

4.7.2 The DICOM transfer syntaxe

The DICOM standard allows a number of transfer syntaxes and compressions schemes. The images created by DICOMatic use the following syntax:

SOP: 1.2.840.10008.1.2.1 = Uncompressed, Explicit VR, Little Endian.

4.7.3 The licenses

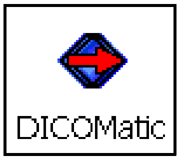
If you do not have a valid license for the modality you are writing, then the program will replace the Patient's name by "Demo DICOMatic", the patient's ID by 1, the Study ID by 1 and it will place a "Demo" watermark on the image.

5 Operation

5.1 Starting the program

5.1.1 From the “Start” menu or the program’s icon

DICOMatic can be started from the menus under “Start, Programs, Tomovision”. Alternatively, you can start the program by clicking on its icon.



5.1.2 From the command line

If you want, you can add one of the optional command line arguments when you start DICOMatic. See Appendix C “Using command line arguments” for more information on this subject.

These arguments are also used if you start the DICOMatic from a DOS batch or from another program.

The main command line arguments for DICOMatic are described in Appendix B.

5.1.3 From a batch or another program

See Appendix B for a detailed description of this procedure.

5.1.4 The DICOMatic.ini file

Some of the program’s configurations can be controlled with the DICOMatic.ini file.

There can be 2 copies of this file: one is in the installation directory (by default: c:\Program Files\TomoVision), the other is in the user’s environment space (on XP: C:\Documents and Settings*user_name*\Application Data\TomoVision, where *user_name* is the current user).

The program will read the copy in the installation directory first, and then, if the file is present, read the copy in the user’s environment space.

The “Config” button can be used to modify the values of the DICOMatic.ini file in the user’s environment space.

For a more detailed discussion of the DICOMatic.ini file, please refer to section 6.

5.1.5 The Conversion Rules file

There can be 2 copies of this file: one is in the installation directory (by default: c:\Program Files\TomoVision), the other is in the user's environment space (on XP: C:\Documents and Settings*user_name*\Application Data\TomoVision, where *user_name* is the current user).

If the program cannot find a copy in the user's environment space, it will use the copy in the installation directory.

For a more detailed discussion of the Conversion Rules file, please refer to section 7.

5.1.6 The DICOM Tables

The content of a DICOM file is dependent on the modality of the images. It is dictated by a series of tables that can be found in the DICOM Table directory.

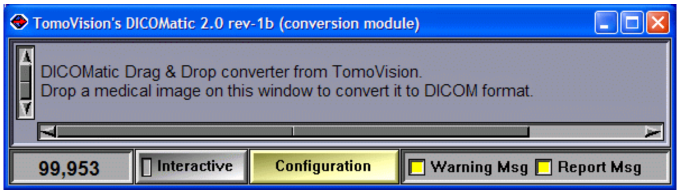
There are 2 copies of the DICOM Tables: one is in the installation directory (by default: c:\Program Files\TomoVision), the other is in the user's environment space (on XP: C:\Documents and Settings*user_name*\Application Data\TomoVision, where *user_name* is the current user).

If the tables do not exist in the user's environment space, the program will create them by copying the tables from the installation directory.

For a more detailed discussion of the DICOM Tables file, please refer to section 8.

5.2 The program’s Interface

DICOMatic’s interface is very simple. The main window consist of a text feedback window, a bar with 3 buttons and a feedback level selector. DICOMatic is a drag & drop program, you interact with it by dropping icons on its window.



```
DICOMatic Drag & Drop converter from TomoVision.
Drop a medical image on this window to convert it to DICOM format.
```

Token Counter The number of license tokens remaining in the license dongle is displayed here.

Interactive If this mode is activated, a DICOM tag edition window will open after the conversion of each image, enabling you to edit the DICOM tags before the image is written to disk.

Configuration Activating this button will open the configuration window. (See section 5.3 for more information on the configuration window.)

Report level This button enables you to select the level of feedback given in the text feedback window during the image’s conversion. You can activate/deactivate:

- The conversion’s “WARNING” messages
- The conversion’s “REPORT” messages.

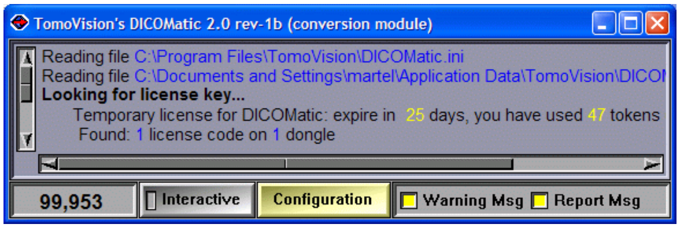
Note:

> You can place the program in trace mode by pressing the control key (“Ctrl”) simultaneously with a number key. Key “0" disables the trace, keys “1" to “5" place the program in trace level 1 to trace level 5.

5.2.1 At startup

First, DICOMatic attempt to read both copies of the “.ini” file.

Then, it queries its licenses and shows the types of licenses currently installed.



```
Reading file C:\Program Files\TomoVision\DICOMatic.ini
Reading file C:\Documents and Settings\martel\Application Data\TomoVision\DICOI
Looking for license key...
      Temporary license for DICOMatic: expire in  25 days, you have used 47 tokens
       Found: 1 license code on 1 dongle
```

5.2.2 Converting Images

Dropping an image on DICOMatic

To convert an image with DICOMatic, just drag & drop the icon of the image over the DICOMatic window.

Note:

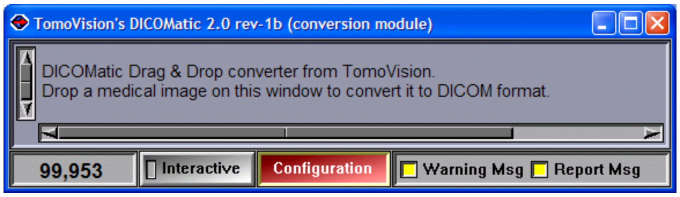
To drag & drop a file, you need to open a Windows directory view where you can see the file's icon, click on this icon with the left mouse button, holding the mouse button while dragging the file's icon over the target program and then releasing the mouse button.

Dropping a directory on DICOMatic

If you drop a directory on DICOMatic, the program will recursively parse the tree structure, starting at the target directory, and convert all the images it recognizes.

5.3 The Configuration window

The behaviour of DICOMatic is controlled by 3 files or group of files: the DICOMatic.ini file, the conversion_rules.txt file and the DICOM Tables. Each of these can be edited interactively through the configuration window.



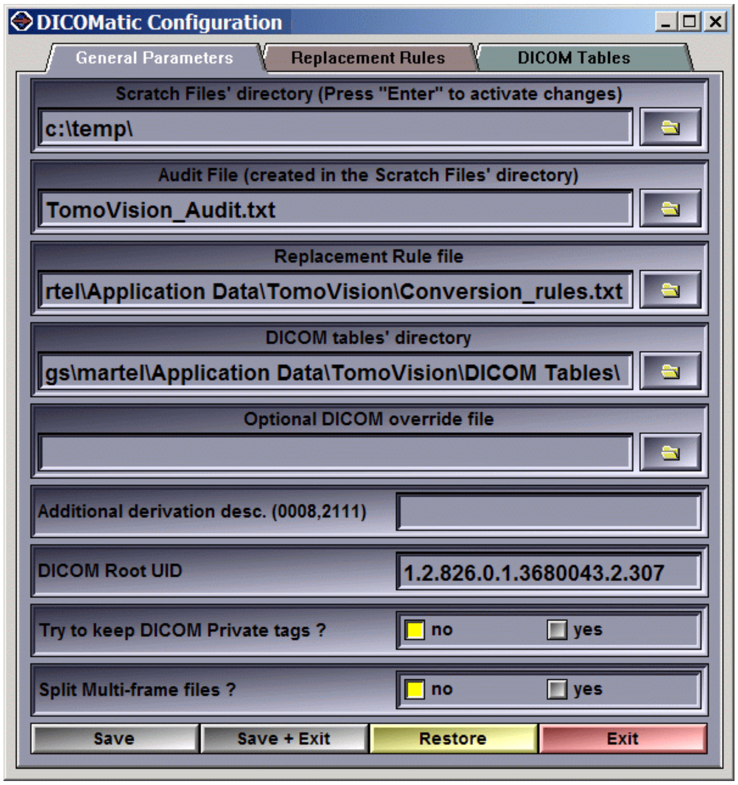
TomoVision's DICOMatic 2.0 rev-1b (conversion module)

DICOMatic Drag & Drop converter from TomoVision.
Drop a medical image on this window to convert it to DICOM format.

99,953

5.3.1 The "General Parameters" page

This page enables you to modify the parameters set by the "DICOMatic.ini" file in the user's "Documents and Settings" directory. The exact syntax and content of the ".ini" file is discussed in more details in Section 6.



DICOMatic Configuration

Scratch Files' directory (Press "Enter" to activate changes): c:\temp\

Audit File (created in the Scratch Files' directory): TomoVision_Audit.txt

Replacement Rule file: rtel\Application Data\TomoVision\Conversion_rules.txt

DICOM tables' directory: gs\martel\Application Data\TomoVision\DICOM Tables\

Optional DICOM override file:

Additional derivation desc. (0008,2111):

DICOM Root UID: 1.2.826.0.1.3680043.2.307

Try to keep DICOM Private tags ? ■ no □ yes

Split Multi-frame files ? ■ no □ yes

Scratch files' directory Any temporary, trace and audit files created by DICOMatic will be created in this directory.

Audit file The "AUDIT" messages created by the Conversion Rules will be appended at the end of this file.

Conversion Rules file The path to the user's "Conversion Rules" file used by the program. By default, this will be the "conversion_rules.txt" file in the user's

“Documents and Settings” directory (“C:\Documents and Settings*user name*\Application Data\TomoVision”)

DICOM tables’ directory The path to the DICOM tables used by the program. By default, this will be the “DICOM Tables” directory in the user’s “Documents and Settings” directory (“C:\Documents and Settings*user name*\Application Data\TomoVision”)

Optional DICOM override file This file, using the DICOM Attribute table syntax (see section 8.2), can be used to modify the DICOM tag values created. It will be read before any other DICOM table. You can use it, for instance, to anonymise your images: just have the “DICOM override file” parameter point to a file with override values for the patient name, ID, and any other TAG you want anonymised.

Additional derivation description DICOMatic will append “converted from xxx by DICOMatic 1.8 rev-zz” to the derivation description tag (0008,2111). If you want to add more information, any text you enter here will be appended after the DICOMatic message.

DICOM Root UID The root UID used by DICOMatic to create most of the image’s UID (see section 4.5.2 for more information on the UID creation). By default, this value is TomoVision’s UID. You can replace it by your enterprise UID if you have one.

Try to Keep DICOM private tags? In some conditions, if the original image has private tags, these can be preserved in the converted image. DICOMatic creates explicit VR images; in order to keep the private tags, the program must be able to know the VR of the private tags. This is the case if the original is also explicit VR, or if you have provided DICOM tables for the private tags (see section 8.3 for more information).

Split Multi-frame files? Some DICOM modalities support multi-frame files. If the original image contains multi-frame, DICOMatic can attempt to create a multi-frame DICOM file.

Save Write the new values to the DICOMatic.ini file in the user’s “Documents and Settings” directory.

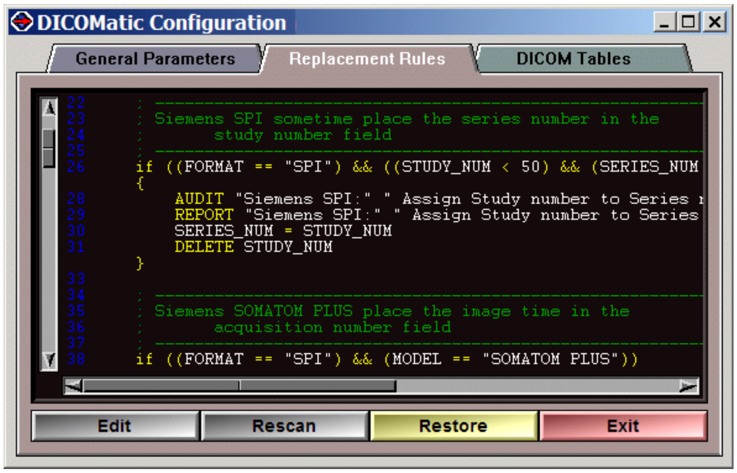
Save & Exit Write the new values and close the configuration window.

Restore Reset all the parameters with the values taken from the DICOMatic.ini file found in the installation directory. Unless you also “Save” these values, they will only be used in the current session of DICOMatic.

Exit Close the configuration window without saving the new values to the .ini file. The changes you made will only be used in the current session of DICOMatic.

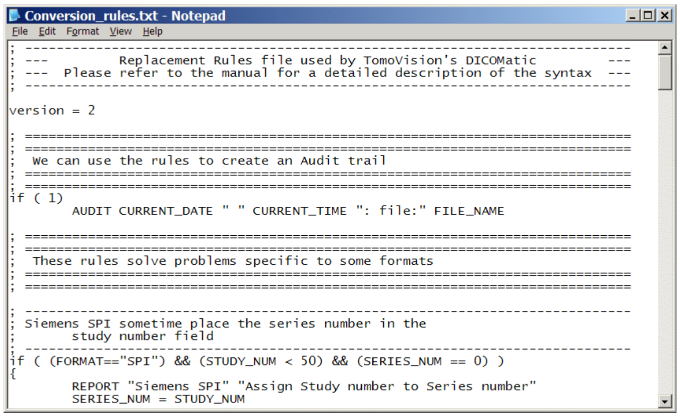
5.3.2 The “Conversion Rules” page

This page enables you to visualize, edit and restore the Conversion Rules file in the user’s “Documents and Settings” directory. The exact syntax and content of the Conversion Rules file is discussed in more details in Section 7.



```
; -------------------------------------------------------------------
; Siemens SPI sometime place the series number in the
;       study number field
; -------------------------------------------------------------------
if ((FORMAT == "SPI") && ((STUDY_NUM < 50) && (SERIES_NUM
{
    AUDIT "Siemens SPI:" " Assign Study number to Series r
    REPORT "Siemens SPI:" " Assign Study number to Series
    SERIES_NUM = STUDY_NUM
    DELETE STUDY_NUM
}

; -------------------------------------------------------------------
; Siemens SOMATOM PLUS place the image time in the
;       acquisition number field
; -------------------------------------------------------------------
if ((FORMAT == "SPI") && (MODEL == "SOMATOM PLUS"))
```

Edit Open a text editor window that will enable you to modify the user’s Conversion Rules file (by default, conversion_rules.txt in the user’s “Documents and Settings” directory). Make sure you “save” the modifications you have done to this file and “rescan” the rules to activate your changes.



```
; ---------------------------------------------------------------------------
; ---         Replacement Rules file used by TomoVision's DICOMatic         ---
; ---  Please refer to the manual for a detailed description of the syntax  ---
; ---------------------------------------------------------------------------

version = 2

; ===========================================================================
; ===========================================================================
;  We can use the rules to create an Audit trail
; ===========================================================================
; ===========================================================================
if ( 1)
        AUDIT CURRENT_DATE " " CURRENT_TIME ": file:" FILE_NAME

; ===========================================================================
; ===========================================================================
;  These rules solve problems specific to some formats
; ===========================================================================
; ===========================================================================

; ---------------------------------------------------------------------------
; Siemens SPI sometime place the series number in the
;       study number field
; ---------------------------------------------------------------------------
if ( (FORMAT=="SPI") && (STUDY_NUM < 50) && (SERIES_NUM == 0) )
{
        REPORT "Siemens SPI" "Assign Study number to Series number"
        SERIES_NUM = STUDY_NUM
```

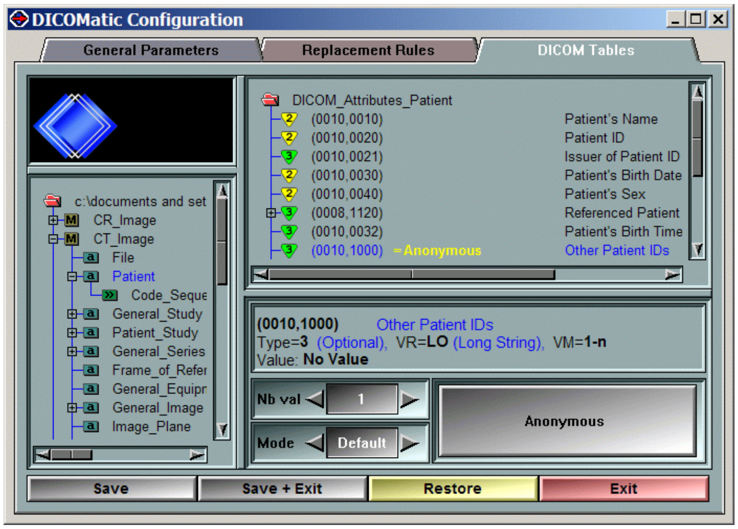
Rescan Delete all current Conversion Rules and re-read them from the user’s Conversion Rules file from the disk.

Restore Replace the content of the Conversion Rules file in the user’s “Documents and Settings” directory by the content of the file found in the installation directory (by default: c:\Program Files\TomoVision\conversion_rules.txt)

Exit Close the configuration window.

5.3.3 The “DICOM Modality” page

This page enables you to visualize, edit and restore the file in the user’s “Documents and Settings” TomoVison Tables sub-directory. The exact syntax and content of these files is discussed in more details in Section 8.



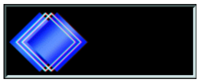
```
c:\documents and set
⊞ M CR_Image
⊟ M CT_Image
    a File
  ⊟ a Patient
      » Code_Seque
  ⊞ a General_Study
  ⊞ a Patient_Study
  ⊞ a General_Series
    a Frame_of_Refer
    a General_Equipr
  ⊞ a General_Image
    a Image_Plane
```

```
DICOM_Attributes_Patient
   2  (0010,0010)                  Patient's Name
   2  (0010,0020)                  Patient ID
   3  (0010,0021)                  Issuer of Patient ID
   2  (0010,0030)                  Patient's Birth Date
   2  (0010,0040)                  Patient's Sex
⊞  3  (0008,1120)                  Referenced Patient
   3  (0010,0032)                  Patient's Birth Time
   3  (0010,1000)  =Anonymous      Other Patient IDs
```

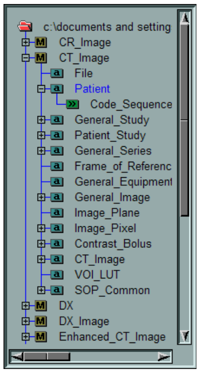
(0010,1000) Other Patient IDs
Type=3 (Optional), VR=LO (Long String), VM=1-n
Value: No Value

The interface of this page is split in 4 re-sizable windows:

Logo window The same interface is used in the “interactive mode”. In that mode, this window is used to display the image. Here, we just display TomoVision’s logo.

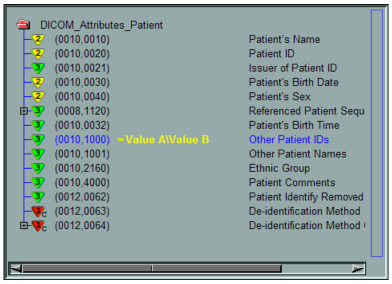


Attribute table There is a hierarchy in the files used to created the DICOM images. For each modality, there is a DICOM_Module_*xx*.txt file (where *xx* is the modality’s abbreviation). This file contains a list of “attribute files”, used to access the DICOM_Attribute_*yy*.txt files (where *yy* is the name of the attribute module). The attribute file contains the actual DICOM tag description. The attribute files can also refer to “macro” files, themselves containing a set of DICOM tag descriptions and possibly refererences to other macro files. The Attribute table selection will show you the hierarchy of these files. The file you select here (attribute or macro) will be used in the following section.



```
c:\documents and setting
⊞ M CR_Image
⊟ M CT_Image
    a File
  ⊟ a Patient
      » Code_Sequence
  ⊞ a General_Study
  ⊞ a Patient_Study
  ⊞ a General_Series
    a Frame_of_Referenc
    a General_Equipment
  ⊞ a General_Image
    a Image_Plane
  ⊞ a Image_Pixel
  ⊞ a Contrast_Bolus
  ⊞ a CT_Image
    a VOI_LUT
  ⊞ a SOP_Common
⊞ M DX
⊞ M DX_Image
⊞ M Enhanced_CT_Image
```

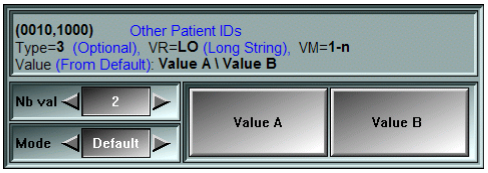
Tag selection The content of the table selected in the previous window will be displayed here. For each tag in the file, a line with the tag number, its optional override/default/append value and a description of the tag will be displayed. If the tag is a sequence, it can be expanded or collapsed. If the sequence contains a macro definition, the reference to the macro, not its content, will be displayed. The tag you select here will be used in the following section.



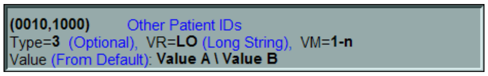
```
DICOM_Attributes_Patient
   2   (0010,0010)                       Patient's Name
   2   (0010,0020)                       Patient ID
   3   (0010,0021)                       Issuer of Patient ID
   2   (0010,0030)                       Patient's Birth Date
   2   (0010,0040)                       Patient's Sex
⊞  3   (0008,1120)                       Referenced Patient Sequ
   3   (0010,0032)                       Patient's Birth Time
   3   (0010,1000)  =Value A\Value B     Other Patient IDs
   3   (0010,1001)                       Other Patient Names
   3   (0010,2160)                       Ethnic Group
   3   (0010,4000)                       Patient Comments
   3   (0012,0062)                       Patient Identify Removed
   1c  (0012,0063)                       De-identification Method
⊞  1c  (0012,0064)                       De-identification Method (
```

Tag edition

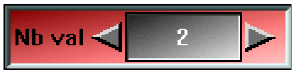
This window enables you to edit the content of the tag selected in the previous window.


(0010,1000) Other Patient IDs
Type=3 (Optional), VR=LO (Long String), VM=1-n
Value (From Default): Value A \ Value B

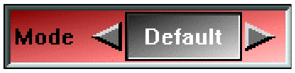
A text window will show you the tag's definition and its current value.


(0010,1000) Other Patient IDs
Type=3 (Optional), VR=LO (Long String), VM=1-n
Value (From Default): Value A \ Value B

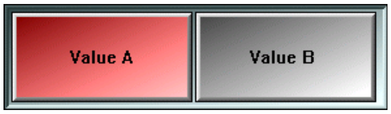
If the tag has a range of possible VM, you can select how many values you want the tag to have. Otherwise this value is not editable.

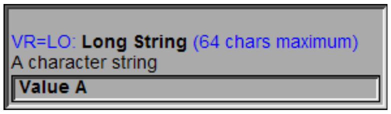


The value you assign to the tag can be an override value (any value for this tag that the image could have will be replaced by the override value you define here), a default value (if this tag is not defined in either the original image or by DICOMatic, the value you defined here will be used) or an append value (the value you define here will be appended at the end of the tag).



The actual values are defined by clicking on its button. The button list will then be replaced by a text window displaying some useful information on the current value's VR syntax. You can type the tag's value in the input field of this window and press Enter.




VR=LO: Long String (64 chars maximum)
A character string
Value A

Save

Write the new values to the selected attribute table in the user's "Documents and Settings" directory.

Save & Exit

Write the new values and close the configuration window.

Restore

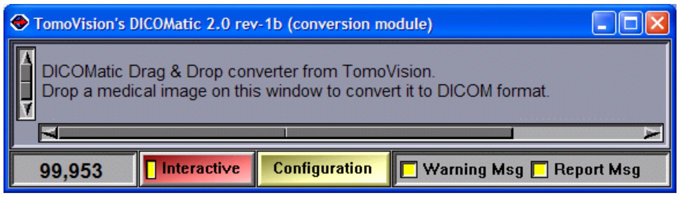
Reset the selected Attribute table with the values taken from the installation directory.

Exit

Close the configuration window.

5.4 The Interactive Option

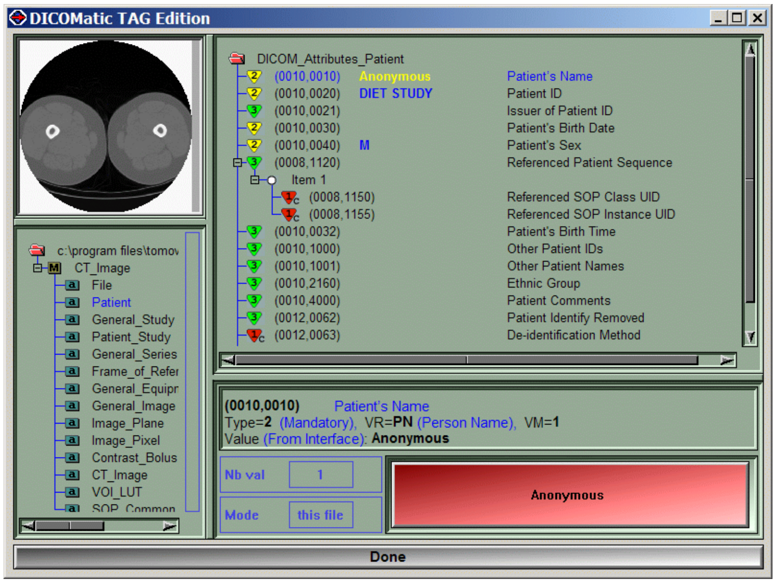
The user has an option to view and edit the content of the DICOM tags just before they are written to the output file. If the “Interactive Mode” button is enabled, the program will open the DICOM TAG edition window.


TomoVision's DICOMatic 2.0 rev-1b (conversion module)

DICOMatic Drag & Drop converter from TomoVision.
Drop a medical image on this window to convert it to DICOM format.

5.4.1 The DICOM tag Edition

The DICOM Edition interface is very similar to the “DICOM table editor” described in the previous chapter. Therefore, we will only describe the differences between the two.


DICOMatic TAG Edition

```
c:\program files\tomov
 CT_Image
   File
   Patient
   General_Study
   Patient_Study
   General_Series
   Frame_of_Refer
   General_Equipr
   General_Image
   Image_Plane
   Image_Pixel
   Contrast_Bolus
   CT_Image
   VOI_LUT
   SOP_Common
```

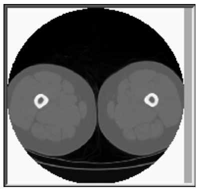
```
DICOM_Attributes_Patient
  2  (0010,0010)   Anonymous     Patient's Name
  2  (0010,0020)   DIET STUDY    Patient ID
  3  (0010,0021)                 Issuer of Patient ID
  2  (0010,0030)                 Patient's Birth Date
  2  (0010,0040)   M             Patient's Sex
  3  (0008,1120)                 Referenced Patient Sequence
       Item 1
         1c  (0008,1150)         Referenced SOP Class UID
         1c  (0008,1155)         Referenced SOP Instance UID
  3  (0010,0032)                 Patient's Birth Time
  3  (0010,1000)                 Other Patient IDs
  3  (0010,1001)                 Other Patient Names
  3  (0010,2160)                 Ethnic Group
  3  (0010,4000)                 Patient Comments
  3  (0012,0062)                 Patient Identify Removed
  1c (0012,0063)                 De-identification Method
```

(0010,0010) Patient's Name
Type=2 (Mandatory), VR=PN (Person Name), VM=1
Value (From Interface): Anonymous

Nb val: 1
Mode: this file

The interface of this page is split in 4 re-sizable windows:

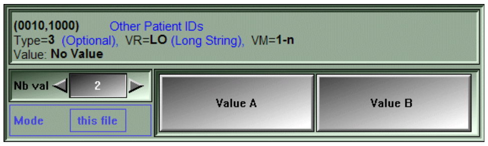
Image preview The current image will be displayed in the top left corner window.



Attribute table This window is very similar to the Attribute table window described in the previous section. The only difference is that only the table for the current image’s modality will be available, and since the macro tables have already been expanded in each attribute files, they will not be shown here.

Tag selection This window is very similar to the Tag selection window described in the previous section. The only differences are that sequences may contain multiple items, and these contain only tags; no references to macro files will be present.

Tag edition This window enables you to edit the content of the tag selected in the previous window. Again, it is very similar to the Tag edition window described in the previous section. The only difference is that the “mode” selection tool will be disabled, since you are editing directly the tag values in the DICOM file.



(0010,1000) Other Patient IDs
Type=3 (Optional), VR=LO (Long String), VM=1-n
Value: No Value

Nb val 2

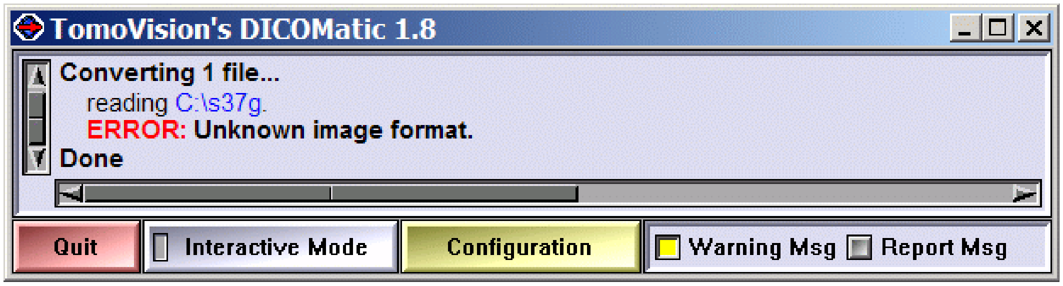
Mode this file

5.5 The conversion feedbacks

While DICOMatic is converting images, it can generate feedback messages, either in the text window of the interface or in an optional “audit trail” file.

5.5.1 The Error Messages

Error messages are used to signal problems that will prevent the program from creating an output DICOM file. Error messages cannot be disabled. When an error is encountered, the DICOM output file is usually not generated.



```
Converting 1 file...
   reading C:\s37g.
   ERROR: Unknown image format.
Done
```

Here are the most common error messages you may encounter:

Unable to open file
By default, DICOMatic places the converted DICOM image in the same directory as the original image. If this directory is write-protected (as in the case of a CD), DICOMatic will be unable to open the directory to create its DICOM file.

Unknown Image Format
The image is not in one of the formats that DICOMatic can convert.

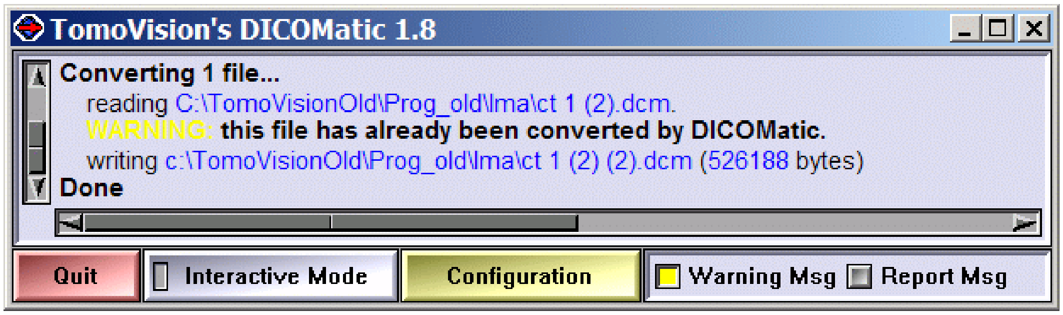
Bad Image Header
The header of the image is corrupted.

Bad Image Data
The image data is corrupted, or it is compressed with a compression scheme that DICOMatic cannot read.

Error commands in the rules files
An error message may be generated from the Conversion Rules with the “ERROR” command.

5.5.2 The Warning Messages

Warning message are used to signal problems that, even though it did not prevent the program from creating an output DICOM file, may result in erroneous data in the converted images. You can disable the displaying of the warning messages from the interface.



```
Converting 1 file...
   reading C:\TomoVisionOld\Prog_old\Ima\ct 1 (2).dcm.
   WARNING: this file has already been converted by DICOMatic.
   writing c:\TomoVisionOld\Prog_old\Ima\ct 1 (2) (2).dcm (526188 bytes)
Done
```

Here are the most common warning messages you may encounter:

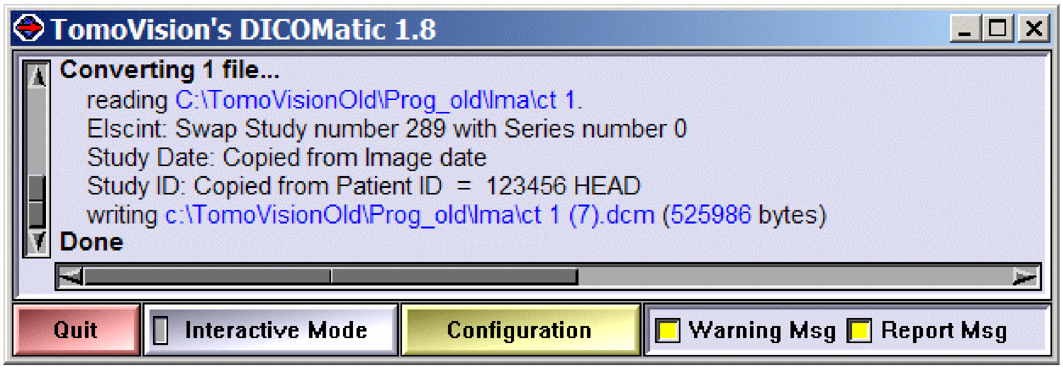
Uncertain orientation
Certain image data do not contain enough information to enable DICOMatic to determine the correct image orientation. If you see this message, please make sure that the horizontal and vertical image direction of the output images are correctly labelled.

Already converted by DICOMatic
This file has already been treated with DICOMatic.

Warning commands in the rules files
A warning message may also be generated from the Conversion Rules with the "WARNING" command.

5.5.3 The Report Messages

Report message are used to signal unimportant problems that should not affect the output DICOM file. These report messages may be of some interest to the user. You can disable the displaying of the report messages from the interface.



```
Converting 1 file...
    reading C:\TomoVisionOld\Prog_old\Ima\ct 1.
    Elscint: Swap Study number 289 with Series number 0
    Study Date: Copied from Image date
    Study ID: Copied from Patient ID  =  123456 HEAD
    writing c:\TomoVisionOld\Prog_old\Ima\ct 1 (7).dcm (525986 bytes)
Done
```

Report commands in the rules files

A report message may also be generated from the Conversion Rules with the "REPORT" command.

5.5.4 The Audit file

Audit messages are appended directly to an audit file. These messages can only be generated from the Conversion Rules with the "AUDIT" command.

By default, the following lines are present in the "conversion_rules.txt" file, and for each file converted a line with the conversion date, current user and the file name is appended to the audit file.

```
; =====================================================================
; =====================================================================
;  We can use the rules to create an Audit trail
; =====================================================================
; =====================================================================
if ( FILE_EXT != 0 )
       AUDIT "DICOMatic: (" CURRENT_DATE " " CURRENT_TIME
             ", User=" CURRENT_USER ") converting file: " FILE_NAME FILE_EXT
if ( FILE_EXT == 0 )
       AUDIT "DICOMatic: (" CURRENT_DATE " " CURRENT_TIME
             ", User=" CURRENT_USER ") converting file: " FILE_NAME
```

From the file "conversion_rules.txt"

Also, any variable modification done by the Conversion Rules while converting a file will generate a report message and an audit message.

```
; ---------------------------------------------------------------------
; Elscint images do not have a series number, but the
;       study number look more like a series number.
;       Let's swap them
; ---------------------------------------------------------------------
if ( FORMAT == "ELSCINT" )
{
      AUDIT  "    Elscint:" " Swap Study number " STUDY_NUM
             " with Series number " SERIES_NUM
      REPORT "    Elscint:" " Swap Study number " STUDY_NUM
             " with Series number " SERIES_NUM
      SWAP STUDY_NUM SERIES_NUM
}
```

From the file “conversion_rules.txt”

The location of the audit file is controlled by the “DICOMatic.ini” file

```
DICOMatic: (27 Oct 2006 14:15:39, User = martel) converting file: ct_1
    Elscint: Swap Study number 289 with Series number 0
```

Sample audit file

6 The DICOMatic.ini file

The DICOMatic.ini file is used to set a number of the program’s internal variables.

The program can read this file from two different locations. The first is in the installation directory (by default: “C:\Program Files\TomoVision”). The second (if present) is in the user’s “Documents and Settings” directory (by default in XP: “C:\Documents and Settings*user name*\Application Data\TomoVision”). If a parameter is defined in both versions of the file, the one in the user’s “Documents and Settings” version will be used.

6.1 The syntax

- everything on a line following a ";" character is a comment and will not be read by the program
- the lines are composed of a keyword, followed by a "=", followed by an argument.
- arguments containing spaces must be protected by quotes (example: scratch = "c:\name with spaces must be quoted”)
- the keywords are not case sensitive, but the arguments are.

6.2 General configuration parameters

scratch= *dir_name* Set the name of the directory used for temporary and trace files to “dir_name” (default value is c:\temp\).

audit= *file_name* Set the name of the audit trace file. This file will be created in the scratch directory. If this parameter is not defined, no audit file will be created. (default value is “TomoVision_Audit.txt”).

silent= *value* Set the inital condition on the “Warning msg” and “Report Msg” interface (default value is 3).
3: both Warning and Report are enabled
2: only Warning is enabled
1: only Report is enabled
0: both Warning and Report are disabled

rules= file_name Set the name of Conversion Rules file in the user’s “Documents and Settings” directory. (default “conversion_rules.txt”)

6.3 DICOMatic configuration parameters

derivation= *string*
This string is appended at the end of the derivation description (tag (0008,2111))

passthrough= *value*
When value=1, if possible, any private tags present in the source file will be preserved in the output file. If value=0, the private tags will not be copied to the output file.

root_uid= *string*
all UIDs (Unique Identification Numbers) created in DICOMatic start with this prefix value. (default value is 1.2.826.0.1.3680043.2.307)

tables= *name*
Set the name of the subdirectory where the DICOM tables are stored in the user’s “Documents and Settings” directory. (default value is "DICOM Tables")

override= *file_name*
Set the name of an optional “override” DICOM table file. This file, using the DICOM Attribute table syntax (see section 8.2), can be used to modify the DICOM tag values created. It will be read before any other DICOM table. (default none)

split_multiframe= *value*
If the source file contains multiple images, if possible and value=0, create a multi-frame file. Otherwise, create multiple files with one frame per file. (default value is 0)

fixed_seq_size= *value*
DICOM supports 2 ways of specifying the length of sequences and items: with a fixed length (value=1), or with an undefined length and seq/item delimiter tags (value=0). (default value is 0)

create_elem_0000= *value*
In the latest version of the DICOM standard, the tag “0000”, “group length”, has been retired. If you wish to create this element nevertheless, use value=1. (default value is 0)

create_contributing_equip= *value*
Do we create the Contributing Equipment Sequence (0018,A001)? If you wish to create this sequence, use value=1. (Default value is 1)

7 The Conversion Rules file

The program can read this file from 2 different locations. If it is present in the user's "Documents and Settings" directory (by default in XP: "C:\Documents and Settings*user name*\Application Data\TomoVision"), then that version will be used. Otherwise, it will look in the installation directory (by default: "C:\Program Files\TomoVision").

7.1 The syntax

Everything on a line following a ";" character is a comment.

7.2 The Rules definition

Each rule is composed of a test line followed by a statement.

The simplest form of if is:

```
if (condition)
      statement
```

7.2.1 The test line

A test line starts with "if" and is followed by a condition between parenthesis.
Example: `if ( STUDY_DATE < "20000101" )`

The condition is evaluated according to the "C" language conventions; if its value is non-zero, the statement is executed.

7.2.2 The condition

A condition can be as simple as a single argument that will be evaluated to either true (if a variable is defined and is not null) or false.
Example: `if ( STUDY_DATE )` or even: `if ( 1 )`

It can also be a more complex mathematical test using relational operators in the form (argument relation argument). The result of that test will be 1 if the relation is true, and 0 if false.

The supported relational operators are: "==" (equal to), "!=" (not equal to), ">" (greater than), "<" (less than), ">=" (greater than or equal to), "<=" (less than or equal to).

Example:

```
if ( SERIES_NUM >= 100 )
```

Tests can be combined with the operators “&&” (and) and “||” (or).

Example:

```
if ( (FORMAT=="NEMA_1") || (FORMAT=="ISAC") )
```

The 4 mathematical operators “+”, “-”, “*” and “/” can also be used in the evaluation of the tests.

Example:

```
if ( ((SERIES_NUM * 100) + IMAGE_NUM) > 1000 )
```

7.2.3 The statement

A statement can be a simple action. The supported actions are: “REPORT”, “AUDIT”, “WARNING”, “ERROR”, “DELETE”, “SWAP” and assignments “=”.

Example:

```
if ( FORMAT == "GE PACE" )
      SWAP ACQUISITION_NUM IMAGE_NUM
```

A statement can also be another nested rule

Example:

```
if ( (FORMAT=="ISAC") && (MANUFACTURER=="TOSHIBA") )
      if ( MODEL == "Xpress" )
            SWAP SERIES_NUM ACQUISITION_NUM
```

Multiple actions can be associated to the same test with the use of “{“ and “}”.

Example:

```
if ( FORMAT == "ELSCINT" )
{
      REPORT "    Elscint: Swap Study and Series numbers”
      SWAP STUDY_NUM SERIES_NUM
}
```

7.2.4 The arguments

Conditions and statements make use of arguments. An argument can be: a **system variable**, an **image variable**, a **DICOM tag variable**, a **string**, an **integer** or a **float** value.

Each variable has an associated type. Operations and relations are only permitted between compatible types.

7.2.5 The system variables

The **system variables** are independent of the images. The values of these variables are not defined by the source images, nor are they used directly in the output image. These variables are:

Variable’s name	Type	Description
CURRENT_DATE	DATE	The current date and time in “yyyymmdd” and “hhmmss” formats

CURRENT_TIME	TIME	
PROG_NAME	STRING	The name (=DICOMatic), version and revision date of this program
PROG_VERSION	STRING	
PROG_DATE	STRING	
PROG_SERIAL	STRING	The ID of the license dongle
CURRENT_USER	STRING	The login name (environment USERNAME variable)
CURRENT_STATION	STRING	The station name (environment USERDOMAIN variable)
TEMPLATE_DATE	STRING	Template used to convert date and time strings variables. Each image format using strings for their dates and times has a default template string associated with each value. You can override these template by defining the “TEMPLATE_DATE” and “TEMPLATE_TIME” variables
TEMPLATE_TIME	STRING	
TRACE_MODE	INT	The current trace value (from 0 to 5)
AUDIT_MODE	INT	The current Audit mode (0=disabled, 1=enabled)
FILE_PATH	STRING	The path to the output file. By default this is the path of the original image
FILE_NAME	STRING	The name of the output file. By default this is the name of the original image
FILE_EXT	STRING	The extension of the output file. By default this is “.dcm”
USER_0 to USER_9	STRING	User variables. Can be set from the command line with “-0 value” to “-9 value”
FLOAT_0 to FLOAT_9	FLOAT	Local variables that can be used in the rules
INT_0 to INT_9	INT	
STRING_0 to STRING_9	STRING	

7.2.6 The image variables

The values of the **image variables** are either derived from the source image, or used to create specific DICOM tags in the output image. The variables are:

Variable’s name	Type	Associated TAG	Description
FORMAT	FORMAT		Format of the original image (“NEMA_1", “NEMA_2", “DICOM”, “DICOM_FILE”, “ISAC”, “SPI”, “ANALYZE”, “BRUKER”, “ELSCINT”...) (See Appendix A for list)

SCAN_TYPE	SCAN	used in (0008,0008)	Scan type of the image (“AXIAL”, “CORONAL”, “SAGITTAL”, “OBLIQUE”, SCOUT”, DERIVED” and “SCRN_SAVE”)
MODALITY	MODALITY	(0008,0060)	“CT”, “MR”, “US”, “OT”, “NM”...
MANUFACTURER	STRING	(0008,0070)	
MODEL	STRING	(0008,1090)	
PATIENT_NAME	NAME	(0010,0010)	Patient identification information
PATIENT_ID	STRING	(0010,0020)	
PATIENT_SEX	STRING	(0010,0040)	
PATIENT_WEIGHT	INT	(0010,1030)	
PATIENT_DOB	DATE	(0010,0030)	
HOSPITAL	STRING	(0008,0080)	
OPERATOR_NAME	NAME	(0008,1070)	
PHYSICIAN_NAME	NAME	(0008,0090)	
RADIOLOGIST_NAME	NAME	(0008,1050)	
STUDY_DATE	DATE	(0008,0020)	Dates using the DICOM "yyyymmdd" format if assigned to a DICOM tag, the format is “dd mmm yyyy” (where mmm is the month’s initials) if assigned to another string. If used in integer operation, the dates are: (((year*100)+month)*100)+day
SERIES_DATE	DATE	(0008,0021)	
ACQUISITION_DATE	DATE	(0008,0022)	
IMAGE_DATE	DATE	(0008,0023)	
STUDY_TIME	TIME	(0008,0030)	Times using the DICOM "hhmmss.cccc" format. If used in integer operation, the times are: (((hours*100)+min)*100)+sec
SERIES_TIME	TIME	(0008,0031)	
ACQUISITION_TIME	TIME	(0008,0032)	
IMAGE_TIME	TIME	(0008,0033)	
STUDY_DATE_TIME	INT		An integer value derived from the date and time information using: (((((year%100)*12 + month)*31 + day)*24 + hrs)*60 + min)*60 + (int) sec
SERIES_DATE_TIME	INT		
ACQUISITION_DATE_TIME	INT		
IMAGE_DATE_TIME	INT		
STUDY_NUM	INT	(0020,0010)	Hospital specific numbers used to identify the images
STUDY_ID	STRING	(0020,0010)	
SERIES_NUM	INT	(0020,0011)	
ACQUISITION_NUM	INT	(0020,0012)	
IMAGE_NUM	INT	(0020,0013)	
ACCESSION	INT	(0008,0050)	Arbitrary sequence number

STUDY_DESCRIPTION	STRING	(0008,1030)	Different description and comments strings
SERIES_DESCRIPTION	STRING	(0008,103E)	
IMAGE_COMMENTS	STRING	(0020,4000)	
PATIENT_COMMENTS	STRING	(0010,4000)	
PATIENT_HISTORY	STRING	(0010,21B0)	
PATIENT_DIAGNOSTIC	STRING	(0008,1080)	
PIXEL_FILL	INT	(0028,0120)	The pixel intensity value for pixels outside the FOV
RESCALE_INTERCEPT	FLOAT	(0028,1052)	The actual pixel value is computed by: pixel intensity = intercept + slope * value
RESCALE_SLOPE	FLOAT	(0028,1053)	
WINDOW_CENTER	INT	(0028,1050)	Preferences for displaying the images
WINDOW_WIDTH	INT	(0028,1051)	
IMAGE_POS_X	FLOAT	(0020,0032)	Each component of the image position vectors
IMAGE_POS_Y	FLOAT		
IMAGE_POS_Z	FLOAT		
IMAGE_ORIENTATION_H_X	FLOAT	(0020,0037)	Each component of the horizontal and vertical image orientation vectors
IMAGE_ORIENTATION_H_Y	FLOAT		
IMAGE_ORIENTATION_H_Z	FLOAT		
IMAGE_ORIENTATION_V_X	FLOAT		
IMAGE_ORIENTATION_V_Y	FLOAT		
IMAGE_ORIENTATION_V_Z	FLOAT		
PIXEL_SPACING_H	FLOAT	(0028,0034)	Horizontal and vertical pixel spacing (in mm)
PIXEL_SPACING_V	FLOAT		
GANTRY_TILT	FLOAT	(0018,1120)	
TABLE_POSITION	FLOAT	(0020,1041)	
IMAGE_POS_D	FLOAT	from (0020,0032) and (0020,0037)	A value derived from the image's orientation and position
SCAN_SEQUENCE	STRING	(0018,0020)	
SCAN_DIAMETER	FLOAT	(0018,0090)	(in mm)
SCAN_TIME	FLOAT	(0018,1150) (0018,9073)	Exposure time (in sec) or Acquisition duration (in sec)

CT_TUBE_VOLTAGE	FLOAT	(0018,0060)	CT and X-Ray Modalities's tube current (in Kv), current (in mA) and exposure (in mAs)
CT_TUBE_CURRENT	FLOAT	(0018,1151)	
CT_EXPOSURE	FLOAT	(0018,1152)	
MR_FIELD_STRENGTH	FLOAT	(0018,0087)	MR Modalities' magnetic field (in Tesla)
MR_TR	FLOAT	(0018,0080)	MR Modalities' recovery, echo and inversion times (in mSec)
MR_TE	FLOAT	(0018,0081)	
MR_TI	FLOAT	(0018,0082)	

7.2.7 The DICOM tag variables

The rules are applied to the images just after the original image's header is read by the program, and before the converted DICOM image is created. This means that when the rules are applied, if the original image itself did not contain DICOM tags, no DICOM tag variables exist. In other words, the DICOM tag variables are only defined for images that are in DICOM format, an ancestor of DICOM (NEMA-1, NEMA-2) or a derivative of DICOM (IS&C, SPI and Papyrus). In your rules, you can assign values to DICOM tags. If this tag exists, you will overwrite its value. If it did not exist you will create it. But if you attempt to use a value from a non existing DICOM tag, you will receive a syntax error.

The syntax is: (gggg,eeee) where gggg is the group number (4 hexadecimal digits) and eeee the element number (4 hexadecimal digits)

All DICOM tags will be considered as STRING variables. Numerical values will be transformed to strings. so the test ((0028,0012) == "1") is valid while ((0028,0012) == 1) is not

Note:

((0028,0012) == 0) tests for the presence of the tag, it does not compare its value to zero.

7.2.8 The variable type compatibilities

FORMAT can be compared ("==" and "!=" only) with:
a STRING (example: `if (FORMAT=="SPI")`)

SCAN can be compared ("==" and "!=" only) with:
a STRING (example: `if (SCAN_TYPE=="AXIAL")`)

MODALITY can be compared ("==" and "!=" only) with:
a STRING (example: `if (MODALITY=="CT")`)

DATE can be compared with:
0 (present or not) (example: `if (IMAGE_DATE != 0)`)
another DATE (example: `if (IMAGE_DATE > SERIES_DATE)`)
a STRING (in the "yyyymmdd" format) (example: `if (PATIENT_DOB>“20000101”)`)

TIME can be compared with:
0 (present or not),
another TIME
a STRING (in the "hhmmss" format)

NAME can be compared with:
0 (present or not),
another NAME
a STRING (in the DICOM format, "family[^surname]")

STRING can be compared ("==" and "!=" only) with:
0 (present or not),
another STRING

Note:

> when comparing strings, you are testing for the presence of string #2 in string #1, not for strict equality

DICOM tags can be compared with:
0 (present or not),
another DICOM tag (example: `if ( (0008,0021) < (0008,0020) )`)
a STRING (example: `if ( (0008,0070) == “Elscint” )`)

INT and FLOAT can be compared with one another

7.2.9 The actions

There are only 7 possible actions supported in the rules:
AUDIT, ERROR, REPORT, **WARNING**, **SWAP**, **DELETE** and "**=**" (assignment).

- **AUDIT** The arguments following the key word "AUDIT" will be formatted as text, concatenated, and written to an audit file.
  Example: `AUDIT  "file:" FILE_NAME " date:" CURRENT_DATE`

- **REPORT** The arguments following the key word "REPORT" will be formatted as text, concatenated, and written to the DICOMatic window.
  Example: `REPORT "Elscint: Swap" STUDY_NUM " with " SERIES_NUM`

- **WARNING** The arguments following the key word "WARNING" will be formatted as text, concatenated, and written to the DICOMatic window with a yellow “WARNING:” heading.
  Example: `WARNING "Invalid value for SERIES_NUM: " SERIES_NUM`

- **ERROR** The arguments following the key word "REPORT" will be formatted as text, concatenated and written to the DICOMatic window with a red “ERROR:” heading. The conversion process will be aborted for this image.
  Example: `ERROR "Unable to convert!"`

- **SWAP** The content of the first argument will be swapped with the content of the second argument. Both arguments must be of the same type!
  Example: `SWAP STUDY_NUM SERIES_NUM`

- **DELETE** The content of the first argument will be deleted
  Example: `DELETE PATIENT_NAME`
  or: `DELETE (0020,0052)`

- **=** The results of an assignment will depend on the type of the variable on the left side of the “=”:

  For STRING and DICOM tag variables, the content of the variable on the left side will be replaced with a concatenating of all the arguments on the right side. Arguments on the right side that are not strings will be converted to strings before being concateneted.
  Example: `FILE_PATH = FILE_PATH "\" PATIENT_NAME`

  INT and FLOAT variables can be assigned the result of a mathematical expression involving other variables. The supported mathematical operators are “+”, “-”, “*” “/” and “@” (“@” is used for the integer modulo operator). Operators are evaluated from left to right. You can use parenthesis to change the order of evaluation.
  Example: `IMAGE_NUM = (SERIES_NUM * 100) + IMAGE_NUM`

  When dates are used in integer operation, they are transformed in a number with the operation:
  (((year * 100) + month) * 100) + day
  So, if you need to extract these values you can use the rules:
  INT_0 = STUDY_DATE @ 100 ; extract the day

INT_1 = (STUDY_DATE / 100) @ 100 ; extract the month
INT_2 = ((STUDY_DATE / 100) / 100) ; extract the year

When times are used in integer operation, they are transformed in a number with the operation:
(((hours * 100) + min) * 100) + sec

For all other types, the variable will be assigned the value of the argument on the right side.
Example: `STYDY_DATE = IMAGE_DATE`

8 The DICOM tables

The program keeps these tables in 2 different locations. It will use the tables located in a sub-directory of the user's "Documents and Settings" directory (by default in XP: "C:\Documents and Settings*user name*\Application Data\TomoVision"). If this directory does not exist or is empty, the program will create it by coping the "DICOM_Tables" sub-directory from the installation directory (by default: "C:\Program Files\TomoVision").

The content of the output DICOM file is determined by the modality of the data in the source file. For each of the supported modalities, an IOD (Information Object Definition) Module table specifies a number of mandatory, conditional and optional attribute modules tables that are used to create the file. The IOD Module Tables are defined in Annex A of part 3 of the DICOM standard (PS 3.3 Annex A).

Each of these attribute tables, in turn, specifies the tags that are used in the output file. Each of these tags is uniquely identified by a group and element number. Depending on the modality, each tag is assigned a "type". The "type" can be "1", "2", or "3". Also, the "type" of a tag may be conditional to some other information in the image. In these cases, the type will have a value of "1C" or "2C".

Type	Meaning (see also PS 3.5 Section 7.4)
1	**Mandatory** This tag and its associate value must be present in the DICOM file. If DICOMatic is unable to assign a value to this tag, a warning message will be displayed.
1C	**Mandatory if other conditions are met**[1]
2	**Mandatory, but can be empty** This tag must be present in the DICOM file. However, if its associated value is unknown, the tag can be written to the file without any values
2C	**Mandatory if other conditions are met, but can be empty**[1] If the conditions are not met, the type 2C elements will NOT be present in the dataset.

[1] At the present time, DICOMatic is unable to evaluate if conditions are met or not. Instead, the type 1C will be treated as type 1 and 2C will be treated as type 2.

3	**Optional and if present, can be empty** This tag is optional. If its associated value is unknown, DICOMatic will not write the tag to the file.

The tables of attributes for each module are defined in Annex C of part 3 of the DICOM standard (PS 3.3 Annex C).

You can edit these tables to add or remove tags you do not want, or to assign override or default values to specific tags.

8.1 The DICOM IOD Module Tables

The tables used by DICOMatic are derived from the IOD Module Tables defined in the DICOM document, section PS 3.3 Annex A. Of course, to make them machine readable, some formatting and syntax changes have to be applied to the tables.

8.1.1 The IOD Module

For each modality supported by the program, there is an IOD Module table. The file names of these tables are in the form: "`DICOM_Module_`**`xx`**`.txt`" (where "**xx**" is replaced by the modality's name. At the present time, the supported modalities are: CR, CT_Image, DX, Enhanced_CT, Enhanced_MR, MR, NM, OT, PET, SC, US_Image, US_Multi-Frame_Image, VL, VL_Photographic_Image, XA and XA_Image.

These files contain a list of the attributes tables needed to create a valid DICOM file of that modality.

Each valid non-commented line in this file contains the name of an attribute table. The program will create the file name "`DICOM_Attributes_`**`yyyy`**`.txt`" (where "**yyyy**" is replaced by the table's name extracted from the line) and read the file matching that name.

Everything on a line following a ";" is a comment. Blank lines are ignored.

The "Usage" and "Reference" information present in the tables are there only for information purpose and are not used by the program.

8.1.2 The “Conditional” and “User Optional” information

By giving the user the power to modify the tables, DICOMatic has lost some of the ability to use conditions when creating the module’s attributes. There is no easy way for the user to specify a condition such as “*Required if the Imaging Subject is a Specimen*” in a form that can be taken from the tables and used by the program without using complex scripting language. So, no “conditions” (“C”) usage information are supported in the IOD Module tables. It is up to the user to comment or un-comment the lines of the attribute tables he wants to see used by the program. The same is also true of the “User Optional” (“U”) usage information, it is implemented by simply commenting or un-commenting the desired attribute tables.

8.1.3 Modifying the IOD tables

You can edit the IOD module files to remove or add attribute files. For intance, if you want to “anonymize” the images converted by DICOMatic, you could set an override value to the appropriate tags in the attribute files, or create a new “anonymous” attribute table that regroups all the tags you want to override. If this file is named “DICOM_Attributes_Anonymous.txt”, then just adding the line “Anonymous” in one of the IOD module table will anonymize all the files of that modality created by DICOMatic.

8.1.4 Inserting a new IOD table

The only occasion you may have to “insert” a new IOD table is if you want DICOMatic to read and modify a DICOM file of a modality not already covered by the program.

DICOMatic already has a number of modality IOD tables defined. However, if you need to convert a DICOM file whose modality is not covered, you can add new tables. DICOMatic uses two information to recognize the modality of a DICOM file, the modality’s UID (tag (0008,0016)) and the modality code (tag (0008,0060)).

The IOD table’s name used by DICOMatic will be created in the following manner:

The modality UID (tag (0008,0016)) is examined to see if the UID is recognized by the program. The recognized UID and their matching IOD table’s name are:

SOP	UID	Filename: DICOM_Module_*xx*.txt where *xx* is:
CR IMAGE	1.2.840.10008.5.1.4.1.1.1	CR_Image
CT IMAGE	1.2.840.10008.5.1.4.1.1.2	CT_Image
ENHANCED CT IMAGE	1.2.840.10008.5.1.4.1.1.2.1	Enhanced_CT_Image
MR IMAGE	1.2.840.10008.5.1.4.1.1.4	MR_Image
ENHANCED MR IMAGE	1.2.840.10008.5.1.4.1.1.4.1	Enhanced_MR_Image
NM IMAGE	1.2.840.10008.5.1.4.1.1.20	NM_Image
PET IMAGE	1.2.840.10008.5.1.4.1.1.128	PT_Image
US IMAGE	1.2.840.10008.5.1.4.1.1.6.1	US_Image
US MULTIFRAME IMAGE	1.2.840.10008.5.1.4.1.1.3.1	US_Multi-Frame_Image
SECONDARY CAPTURE IMAGE	1.2.840.10008.5.1.4.1.1.7	SC_Image

If the UID is not one of these, then the IOD Module table name is created using the following syntax:

`DICOM_Module_`**`xx`**`.txt`

where “**xx**” is the 2 character modality code as specified by the “modality” tag (0008,0060).

If no table is found with that name, then the default table “Other Modality” table is used:

`DICOM_Module_OT_Image.txt`

8.2 The DICOM Attribute tables

The tables used by DICOMatic are derived from the Attribute tables defined in the DICOM document, section PS 3.3 Annex C. Of course, to make them machine readable, some formatting and syntax changes have to be applied to the tables. In addition, to help the users, the VR and VM information have been written for each tag in the tables. These are extracted from the “Data Dictionary” part of the DICOM document (section PS 3.6).

8.2.1 The Attribute tables syntax

Everything on a line following a “;” is a comment. Blank lines are ignored.

Each valid non blank line will contain either included macro file instructions or a series of informations, separated by spaces or tabs and using the following structure:
- an optional control character
- one or more optional nesting level indicator character
- the group/element pair between parenthesis
- an optional value between quotes
- an optional Type definition
- an optional VR definition
- an optional VM definition
- an optional description

The starting control character, either “=”, “-“, “+” or “!”, is optional. It is used to specify that the tag defined on this line should be assigned a default, override or an append value or if the tag should be removed.

The nesting level indicator is the character “>”. Tags that are inside a sequence will be preceded by this character. If the sequence is inside another sequence, then 2 characters are used, etc.

Every tag is defined by a group/element pair. The syntax for these is (xxxx,yyyy) where xxxx is the group number and yyyy is the element number (both in Hexadecimal, no spaces allowed). (example: `(0018,106E)`)

The group/element pair can be followed by an optional value between quotes (see the default override and append section).

The type definition uses the syntax: “`type=`**`xx`**”, where “**`xx`**” can be“1", “1C”, “2", “2C” or “3". If a tag is defined in multiple attribute tables, it will be assigned the lowest value (1 < 1C < 2 < 2C < 3). If the type is omitted, it will be assumed to be 3.

The VR definition uses the syntax: “`VR=`**`xx`**”, where “**`xx`**” is the VR (Value Representation) of this tag. The possible VR values are specified in the DICOM document, PS 3.5 section 6.2.

The VM definition uses the syntax: “`VM=`**`x`**” or “`VM=`**`x-y`**”, where “**`x`**” is a number and “**`y`**” can be a number or “n”. The VM (Value Multiplicity) term is defined in the DICOM document, PS 3.5 section 6.4.

The description definition uses the syntax: “`desc="`**`string`**` "`”, where “**`string`**” is the description string between quotes.

Example from the “`DICOM_Attributes_NM_Detector.txt`” file:

```
(0054,0022)           type=2   VR=SQ   VN=1   ; Detector Information Sequence
> (0018,1180)         type=3   VR=SH   VN=1   ; Collimator/Grid Name
> (0018,1181)         type=2C  VR=CS   VN=1   ; Collimator Type
...
> (0054,0220)         type=3   VR=SQ   VN=1   ; View Code Sequence
>>  include "Code Sequence Macro" ; Table 8.8-1
>>  (0054,0222)       type=2C  VR=SQ   VN=1   ; View Modifier Code Sequence
>>>   include "Code Sequence Macro" ; Table 8.8-1
```

Note:

> The “VR”, “VM” and description definitions for non-private tags are optional since the program already know these from its internal tables. Their presence in the tables is for information purpose. They are not used by the program.

8.2.2 Default, Override and Append Values

You can add a default, override or append value to a tag by preceding the tag with the "=", "!" or “+” control character and adding a value after the tag. The default value will be used if no value is derived from the original file. The override value will be used regardless of the original file's values. The append value will be added after any existing values of the tag.

The value must be between quotes "" (example: `"23.5"`). You can use the '\' character to define multiple vales (example: `"1.0\1.0\5.5"`).

Depending on the tag’s VR, the appended values will be either a new value added after the other existing values if the tag supports multi-values (example: VR=IS), or a string appended at the end of any existing string if the tag is a string value (example: VR=LT).

Goal	Syntax
To set "value" as the **default** value for tag (xxxx,yyyy)	= (xxxx,yyyy) "value"
To set "value" as the **override** value for tag (xxxx,yyyy)	! (xxxx,yyyy) "value"
To append "value" at the end of any other data for tag (xxxx,yyyy)	+ (xxxx,yyyy) "value"

Example: To set a default institution name

```
= (0008,0080)  "My Hospital" type=3 VR=LO VN=1 desc="Institution Name"
```

Example: To override the pixel padding value

```
! (0028,0120)  "-1000"
```

Example: To append a new patient name after any existing “other names”

```
+ (0010,1001)  "Smith^Joe" type=3 VR=PN VN=1-n ; Other Patient Names
```

8.2.3 Removing a tag

You can cause a DICOM tag to be removed from the file created by DICOMatic by preceding the tag with the "-" control character

8.2.4 Sequences

DICOMatic support sequences. The elements in the sequence are preceded by one or more nesting level indicator characters “>”. The nesting level of the tags in the sequence must be one level higher than the sequence tag.

The “End of Sequence” tag is not required to terminate a sequence.

The “Item” and “End of Item” tags can be used to bracket items in a sequence. They are only needed if you want to create multiple items for this sequence, otherwise the level indicator “>” is enough.

Example: with a single item in the sequence

```
(0040,0275)    type=3   VR=SQ  VN=1 ; Requested Attributes Sequence
> (0040,1001)  type=1C  VR=SH  VN=1 ; Requested Procedure ID
> (0040,0007)  type=1C  VR=LO  VN=1 ; Schedule Procedure Step Description
> (0040,0008)  type=1C  VR=SQ  VN=1 ; Schedule Procedure Code Sequence
>>  Include "Code Sequence Macro"
> (0040,0009)  type=1C  VR=SH  VN=1 ; Schedule Procedure Step ID
```

Example: with 2 items in the sequence (We need Item and Item Delimitation tags)

```
(0018,6011)    type=3   VR=SQ  VN=1 ; Sequence of Ultrasound Regions
> (FFFE,E000)                       ; Item #1
> (0018,6018)  type=1   VR=UL  VN=1 ; Region Location Min x
...
> (0018,6043)  type=3   VR=SL  VN=1 ; TM-Line Position y1
> (FFFE,E00D)                       ; Item Delimitation
> (FFFE,E000)                       ; Item # 2
> (0018,6018)  type=1   VR=UL  VN=1 ; Region Location Min x
...
> (0018,6043)  type=3   VR=SL  VN=1 ; TM-Line Position y1
> (FFFE,E00D)                       ; Item Delimitation
```

Note:

> When DICOMatic creates sequences of attributes, it uses the first item of the sequence defined in the attribute table as a template for any items it has to create. The only reason you may want to specify multiple items in a sequence in the attribute tables is if you wish to have default or override values specific to each item.

8.2.5 Include

You can also include macro tables with the line:

```
Include "name"
```

Where "**name**" is the name of the table you want to include, composed of words delimitated by spaces. The spaces in the name will be replaced by "_" and that name will be appended to the string "DICOM_Include_" and followed by the string ".txt" to create the name of the file that will be included by the program.

Macro files use the same syntax as the attribute files. They can include other macro files.

Example: To include the file "DICOM_Include_Code_Sequence_Macro.txt"

```
include "Code Sequence Macro"
```

8.2.6 Writing Private Tags to the output file

You can also add private tags to the existing list. However, if you add private tags, you will also need to define their VR, or add them in a private tag table (see section 7.3) so that its VR value will be known to DICOMatic.

Example: to add a private tags with default values

```
= (0009,0010) "My_Company"  type=3  VR=LO
= (0009,1001) "-123.45"     type=3  VR=FL  desc="My private value"
```

8.2.7 DICOMatic and the conditional types

By giving the user the power to modify the tables, DICOMatic has lost some of the ability to use conditions when creating the attributes. There is no easy way for the user to specify a condition such as "*Required if synchronization channel is encoded* *in a waveform*” in a form that can be taken from the tables and used by the program without using complex scripting language.

So, when creating the tags, except for some well known tags, the condition is assumed to be true. A type “1C” will be treated as “1" and “2C” as “2". However, when testing the file for coherence and reporting on missing tags, the conditions will be assumed to be false.

8.3 The Private Tags tables

DICOMatic will read all the private tags from a DICOM file, but when it writes the tags back to the output file, since the mode is “Explicit VR”, the program can only write the tags for which it knows the VR. For most Implicit VR private tags the VR is unknown and the program will be unable to write them even if you specify the "Keep_Private_Tag" option.

If you want the private tags to make it through DICOMatic, you must define them in a table.

Private tags are recognized by 2 sets of information: their creator string and their group/element numbers. For each private group (all odd number groups are private), the elements between 0010 and 00FF are used to define creator strings. The elements between 1000 and FFFF are parsed using the first 8 bits of the element number to refer to the creator. For example if we have:

```
(0009,0010) “A”
(0009,0011) “B”
```

then the tag (0009,1001) would refer to the element 1001 of the creator “A”, and (0009,1101) to the element 1101 of the creator “B”.

You need to create one file for each different creator string in the private tags. That file will include all the private tags you want the program to recognize.

Note:

> Private tag tables are only needed if you want DICOMatic to preserve private tags from an “Implicit VR” file. They are not needed for “Explicit VR”.

8.3.1 The Private Tags table name

The file name is composed of part of the creator string, converted to upper cases, preceded by “DICOM_Private_” and followed by “.txt”. (Example: `“DICOM_Private_PHILIPS MR.txt”`).

The file name does not need to include the entire creator string, but just enough to differentiate the creators. For example, the file name `“DICOM_Private_PHILIPS_MR.txt”` will be used to locate private tags for the creators “PHILIPS_MR”, “PHILIPS_MR/PART” and “PHILIPS_MR/LAST”.

Note:

The characters \ / : * ? ” < > and | are not allowed in a file name under Windows. If present in the creator string, they will be replaced by “_” in the file name. However, the space character will not be replaced, so the strings “Philips MR” and “Philips_MR” are not equivalent.

Note:

For DICOMatic the creator string is not case sensitive. The string “Philips_MR” and “PHILIPS_MR” are equivalent. Both will be matched to the file “DICOM_Private_PHILIPS_MR.txt”

8.3.2 The Private Tags table syntax

Everything on a line following a “;” is a comment. Blank lines are ignored.

Each information on a line must be separated from another by 1 or more spaces or tab characters. The informations present on each line are:

- an optional control character
- the group/element pair between parenthesis
- a VR definition
- an optional VM definition
- an optional description

The starting control character,“-”, is optional. It is used to specify that the tag should be removed.

Every tag is defined by a group/element pair. The syntax for these is (xxxx,yyyy) where xxxx is the group number and yyyy is the element number (both in Hexadecimal, no spaces allowed). (example: `(0018,106E)`)

The VR definition uses the syntax: “`VR=`**`xx`**”, where “**`xx`**” is the VR (Value Representation) of this tag. The possible VR values are specified in the DICOM document, PS 3.5 section 6.2.

The VM definition uses the syntax: “`VM=`**`x`**” or “`VM=`**`x-y`**”, where “**`x`**” is a number and “**`y`**” can be a number or “n”. The VM (Value Multiplicity) term is defined in the DICOM document, PS 3.5 section 6.4.

The description definition uses the syntax: "`desc="`**`string`**`"`", where "**`string`**" is the description string between quotes.

example: from the "DICOM_Private_Philips MR Imaging DD.txt" file

```
(2005,1005) VR=CS desc="Synergy Reconstruction Type"
(2005,1020) VR=SL desc="Number of Chemical Shifts"
(2005,10a1) VR=CS desc="Syncra Scan Type"
```

Appendix A: The different Image Formats

Even though most manufacturers now offer the DICOM format, there is still a multitude of image formats in circulation. Here is a description of the formats that DICOMatic is able to understand and convert. For each of these, the value of the Conversion Rules variable “FORMAT” is specified.

The DICOM format

The DICOM (Digital Imaging and Communication in Medicine) format, created by NEMA (National Electrical Manufacturers Association), is now the de facto standard for medical images. All recent scanners output their images directly in this format. DICOM supports 2 types of image compression: RLE (Run Length Encoding) and JPEG (JPEG, JPEG-LS and JPEG 2000).

Why would you need to use DICOMatic to convert images from DICOM to DICOM? Over the years, some manufacturers have created images that have problems (errors in their tags or in the compression). DICOMatic should be able to correct most of these errors. DICOMatic will also uncompress[2] the images it reads.

FORMAT = “DICOM_FILE” for files respecting the DICOM part 10 syntax
FORMAT = “DICOM” for other DICOM files
FORMAT = “NEMA_1” or “NEMA_2” for older NEMA files

CT and MR Image formats

The Bruker Paravision format

The Bruker data set is split in multiple files. The “2dseq” file contains the image data. The “reco” “d3proc” and “procs” files, in the same directory as the “2dseq” file, as well as the “imnd”, “acqp” and “subject” files higher in the hierarchy, contain the header information. To create a good DICOM header, DICOMatic attempts to read as many of these files as it can. You must therefore have the complete Paravision tree structure on your disk and drop one of the header files on DICOMatic (either the “d3proc” or the “procs” file).

FORMAT = “BRUKER”

[2] JPEG lossy decompression is done with the public domain Portable Video Research Group PVRG library, JPEG 2000 decompression is done with the JasPer library.

The Dichte pQCT format

This format is used by the Dichte for their pQCT scanner

FORMAT = “DICHTE_PQCT”

The Elscint format

Elscint images start with the first 2 characters: 0xce, 0xfa. The image data in an Elscint image is compressed.

FORMAT = “ELSCINT”

Elscint DICOM images with Lossless Rice compression

These images are used internally by Elscint to store the data. They were not meant to exist outside the console and have probably been copied directly from the SGI disk without using the “export” utility.

FORMAT = “DICOM”

The Hitachi MPR/AIRIS format

These images created by some Hitachi MR scanners start with the 4 characters “HDL1". The images can be either compressed or uncompressed. If the images are uncompressed and not a pilot, then only the data interior to the circular field of view is present in the file. These images use a format similar to the Philips HDL1 format.

FORMAT = “HDL1” or “HITACHI_HDL1”

The Imatron format

The Imatron images have a variable size header and a block of non-compressed images.

FORMAT = “IMATRON”

The IS&C format

The IS&C (Image Save and Carry) format (also called ISAC) is used by Japanese scanners (Toshiba and Shimadzu). Images extracted from an IS&C disk can sometimes be presented in multiple parts. The “.INF” file contains a NEMA header and the “.IMG” file contains the image data. To convert these images, both parts must be in the same directory. Drop the “.INF” header file on DICOMatic, it will automatically fetch the corresponding “.IMG” image and merge it with its “.INF” header in a DICOM image. The Toshiba “.IMG” file can be compressed.

FORMAT = “IS&C” or “ISAC”

The GE 9800 Format

The 9800 format was used in really old CT scanners.

FORMAT = “GE_9800”

The GE Signa_3, Signa_4 and Signa_5.x formats

These formats are used extensively by a number of CT and MR scanners from GE. The Signa_4 and Signa_5 formats start with the magic word: “IMGF”. The Signa_4 & Signa_5 images offer 4 modes for storing the images: Uncompressed, Compressed, Packed and Compacted.

FORMAT = “GE_SIGNA_3”
FORMAT = “GE_SIGNA_4”
FORMAT = “GE_SIGNA_5”

The GE SYTEC Format

This format is specific to the SYTEC scanner.

FORMAT = “GE_SYTEC”

The GE PACE Format

The PACE format was developed by Yokogawa for the PACE scanner. The PACE format starts with the magic word “!STD”.

FORMAT = “GE_PACE”

The Philips HDL1 format

These images use a format similar to the Hitachi MPR/AIRIS format.

FORMAT = “HDL1” or “PHILIPS_HDL1”

The Philips JUSTFF format

These images created by some Philips CT scanners start with the 6 characters “JUSTFF".

FORMAT = “PHILIPS_JUSTFF”

The Philips MRH1 format

These images created by some Philips MR scanners start with the 4 characters “MRH1". At the present time, DICOMatic cannot read these images.

FORMAT = “PHILIPS_MRH1”

The Philips SPI format

Philips EasyVision and GyroScan scanners use a file format similar to the Siemens SPI format.

FORMAT = “SPI”

The Picker IIF format

These files start with the string “IF” and can contain multiple images per files. This format is used by the Nordstar scanner

FORMAT = “PICKER_NORDSTAR”

The Picker IQ/PQ format

The image data in a Picker image can be either compressed or uncompressed.

FORMAT = “PICKER_IQ/PQ”

The Shimadzu SCT format

The image data in a Shimadzu image is compressed.

FORMAT = “SHIMADZU_SCT”

The Siemens SPI format

Most of the Siemens scanners used to be able to export images either in a device-specific native format, or in a more “standard” SPI format. The SPI format is based on the NEMA-1 (and later NEMA-2) format. SPI images generated by the Somatom scanners can also be compressed.

FORMAT = “SPI”
FORMAT = “SPI_COMPRESSED”

The Siemens Native formats

If a Siemens image is not in DICOM or SPI format, it is probably in one of the following native formats:

- **Numaris-1**
  FORMAT = “SIEMENS_NUMARIS_1”
- **Numaris-2**
  FORMAT = “SIEMENS_NUMARIS_2”
- **Vision**
  FORMAT = “SIEMENS_VISION”

- **Somatom Plus**
  FORMAT = “SIEMENS_SOMATOM_PLUS”
- **Somatom Plus-4**
  FORMAT = “SIEMENS_SOMATOM_PLUS_4”
- **Somatom AR**
  FORMAT =“SIEMENS_SOMATOM_AR”
- **Somatom DR**
  FORMAT = “SIEMENS_SOMATOM_DR”

The SMIS format

This is a format used by the Surrey Medical Imaging System (now MRRS).

FORMAT = “SMIS”

The Toshiba Flexart format

These files, starting with the string “VF” are used by Toshiba on their Flexart MR scanners. At the present time DICOMatic is unable to extract the pixel’s positions and dimensions from these files. As a consequence of this, the converted images will be of the modality “Secondary Capture” (SC). The patient’s information will be extracted from the “patient_database” file if it is present in the parent directory.

FORMAT = “TOSHIBA_FLEXART”

The Varian format

If the pixel data is 32 bits (float or integer), it will be converted to 16 bits short integer in the DICOM images.

FORMAT = “VARIAN”

Ultrasound image formats

The ATLI format

This format from ATL (Advanced Technology Laboratories) has a very small header (starting with the characters “ATLI”) followed by a JPEG image.

FORMAT = “ATL_I”

The DEFF format

The DEFF (Data Exchange File Format) format was created by ATL (Advanced Technology Laboratories). This format is based on the TIFF image format[3] with a number of private tags.

FORMAT = "DEFF"

The DSR format

The DSR (Digital Storage and Retrieval) format was created by HP and also used by Agilent Technologies. This format is based on the TIFF image[3] format with a number of private tags.

FORMAT = "DSR"

The Medison format

DICOMatic will convert images written in the Medison Kartesian format.

FORMAT = "MEDISON"

The Kretz File format

There are 2 variations of this format. The "native" format is present in the files with the "V00" extension, and there is also a DICOM variation.

FORMAT = "KRETZ_FILE"

Nuclear Medicine Image formats

The ADAC format

DICOMatic will convert images written in the ADAC format.

FORMAT = "ADAC"

The Ge Advance PET

DICOMatic will convert images written in the Ge Advance PET format.

FORMAT = "GE_GEMS_PET"

The Interfile format

The Interfile format has been developed for the transfer of images in the nuclear medicine community. This format is composed of an ASCII header

[3] TIFF decompression us done with Sam Leffler's the public domain library from Silicon Graphics.

followed by the image. The header is composed of a series of key-value pairs.

FORMAT = “INTERFILE_3.2”
FORMAT = “INTERFILE_3.3”

The Toshiba NEMA NM

This is a NEMA format from Toshiba with information in the private groups.

FORMAT = “TOSHIBA_NEMA_NM”

Application-specific and other image formats

The following formats are used by specific applications or institutions:

The Analyze format

This format is used by the Analyze software of the Mayo Clinic. The Analyze images have 2 components: a “.hdr” header file and a “.img” raw data file. To convert these images, you must drop the header file on DICOMatic.

FORMAT = “ANALYZE”

The Camtronics XA format

This format is used by Camtronics for their XA images. To convert these images, the 1Kb “header” file must be present in the same directory as the images.

FORMAT = “CAMTRONICS”

The IGB format

This format is used by the “Institut de Genie Biomedical” in Montreal. This format is composed of an ASCII header followed by the image.

FORMAT = “IGB”

Lead Technologies / iScan optical scanner

This scanner is used to digitize film. The technology seem to be used by a few manufacturers. Among these is the iScan software from CompuRad. The resulting images are in 2 prts: a TIFF image and a header file. To convert these images, you need to drop the header file on DICOMatic.

FORMAT = “LEAD_OPTICAL”

The Materialise format

This format is used by the Materialise software.

FORMAT = “MATERIALISE”

The Optimed XA format

This format is used by Optimed for their XA images. To convert these images, the “ARCHIVE.PAT” file must be present in the same directory as the images.

FORMAT = “OPTIMED_XA”

The Papyrus format

The Papyrus format is used by the Papyrus software. It is based on DICOM and the images start with the magic word “PAPYRUS”.

FORMAT = “PAPYRUS_2”
FORMAT = “PAPYRUS_3.0”
FORMAT = “PAPYRUS_3.1”

The RAW format

This format is used by TomoVision to add a header to raw images or to modify the header information of known images. The RAW header is created with the free “Raw_Header” program from TomoVision. To convert these to DICOM, both the original image and the “.raw” header must be in the same directory. You must drop the ”.raw” header on DICOMatic. It will automatically fetch the corresponding original image and merge it with its header in a DICOM image.

FORMAT = “RAW”

Other “Secondary Capture” formats

The JPEG format

Since the DICOM images can be compressed using the JPEG image format, DICOMatic can read and convert JPEG images. Since the JPEG header does not contain enough information to ascertain the image’s origin, JPEG images will be considered “SC” (Secondary Capture) with a modality of “OT” (Other).

FORMAT = “JPEG”
FORMAT = “JPEG-2000” for JPEG2000 code stream

FORMAT = “JP2” for JP2 wrapper

The TIFF format

Since a few Ultrasound formats are based on the TIFF image format, DICOMatic can read and convert TIFF images. Since the TIFF header does not contain enough information to ascertain the image’s origin, TIFF images will be considered “SC” (Secondary Capture) with a modality of “OT” (Other).

FORMAT = “TIFF”

Appendix B: Using DICOMatic in a batch

To use DICOMatic in a batch or to call it from an external applications, you need to know what arguments it is expecting in its command line, and what return values you can expect from it.

DICOMatic command line arguments

The main command line arguments for DICOMatic are:

DICOMatic [-a *input*] [-b *output*] [-t *val*] [-v] [-z] [-q] ...

where:

Trace & debugging information

-t *val* Trace mode, val must be between 1 and 5. This will force DICOMatic to create a trace file in the C:\temp directory. This file can be used by TomoVision for debugging purpose. Level 5 generates a much more detailed trace than level 1.

-v Pop a window with the version and compilation date of the program.

Input / output files

-a *input_file* input file (or folder). Only 1 file or folder per call. if * is a folder, DICOmatic will recursively parse the tree structure and convert all the files it found. If the file name contains spaces, it must be protected by quotes.
Example: DICOMatic -i “c:\my dir\name with spaces”

-b *output_file* output name (by default, the output name is the input file name with the ".dcm" extension) If the file name contains spaces, it must be protected by quotes.
Example: DICOMatic -i test -o “c:\my dir\name with spaces”

-c *used_file* name of the “used files” file. The program will write to this file the path of all the files it has used in creating the converted image. This may be useful if you want to clean-up the directory after the conversion.

-o *override_file* name of the DICOM override file.

-d *path* Redirect the output file to the specified “path”. By default the files are created in the directory of the input file. With this argument you can send the output file to the “path” directory.

Valid license information

-z Enable NetHASP licenses.

-q Query Licenses, the program returns the licenses flag. This flag is a 32 bit integer where each of the supported formats has a bit. The list of supported formats and their associated license number is available from TomoVision.

-l *license* Get the expiration date for a specific license where “license” is the license number.

DICOM parameters

-f *string* "from" the argument will be added to DICOM derivation tag (0008:2111). If the argument contains spaces, it must be protected by quotes.
Example: DICOMatic -f “converted in house”

-m *modality* Default modality to use when converting this file.
Example: DICOMatic -m CT

-p “Private tags”. The presence of this argument will cause DICOMatic to let private tags (odd groups) present in DICOM source images through to the converted DICOM images if possible.

-r *root_UID* Use “root_UID” in all the DICOM UID parameters (by default, TomoVision’s UID (1.2.826.0.1.3680043.2.307) is used).

User defined rules variables

-0 *string* “string” will be associated to the system variable USER_0
to
-9 *string* “string” will be associated to the system variable USER_9

Windows parameters

-x *pos dim* X position & dimension of DICOMatic window

-y *pos dim* Y position & dimension of DICOMatic window

DICOMatic return codes

If the program successfully converts the image, it will return 0. If not, the following return values describe the problem the program has encountered:

1 The image format is not known to DICOMatic

3 DICOMatic was unable to open the input file

4 DICOMatic was unable to open the output file

5 Unable to read the image’s header: Either the image header is corrupt, or the reading was aborted by an “ERROR” command in the rules.

6 Unable to read the image’s data: The image seems to be corrupt

8 A 3D data set contains too many images for DICOMatic

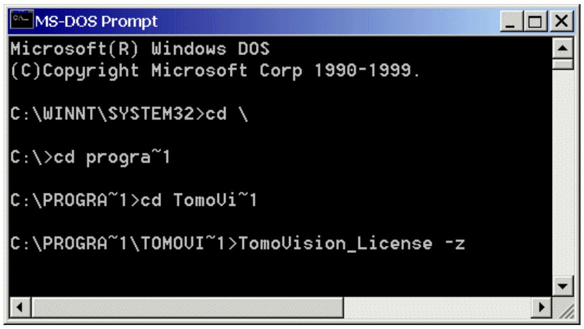
>10 No valid license was found for the image format. (return value - 10) is the actual format number. This is just a warning, since the image was created with a watermark anyway.

Appendix C: Using command line arguments

You may need to use command line arguments to enable some of the options of a program. There are 3 ways to do this:

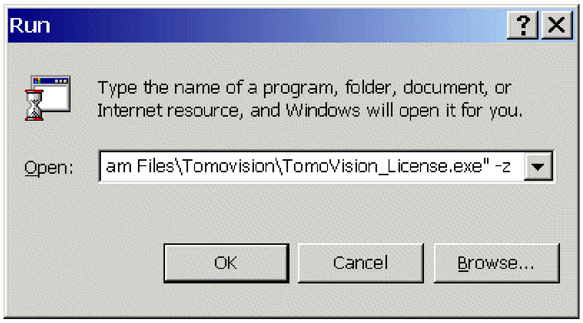
Start the program from a DOS Prompt

You can type in directly the filename of the executable program, followed by the options, in a DOS Prompt window.



```
Microsoft(R) Windows DOS
(C)Copyright Microsoft Corp 1990-1999.

C:\WINNT\SYSTEM32>cd \

C:\>cd progra~1

C:\PROGRA~1>cd TomoVi~1

C:\PROGRA~1\TOMOVI~1>TomoVision_License -z
```

Start the program from the “Run” window

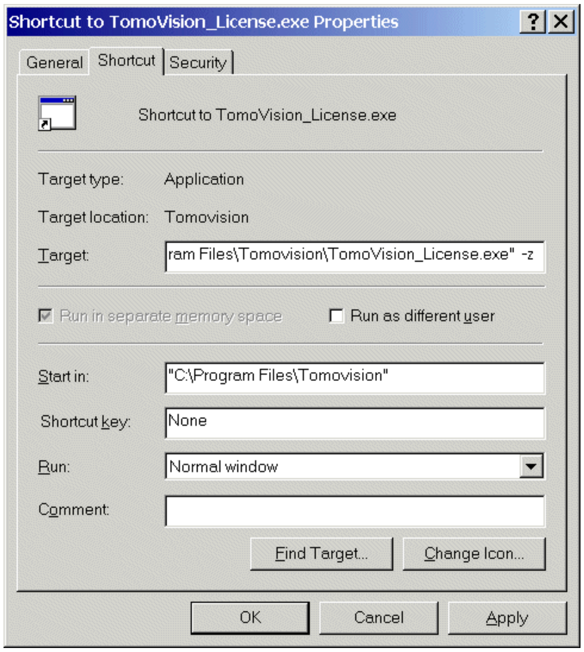


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: am Files\Tomovision\TomoVision_License.exe" -z

From Windows “Start” menu, select the “Run” option to open the “Run” window. In this window, you can use the “Browse...” button to browse the file structure until you find the program (by default TomoVision’s programs are installed in C:\Program Files\Tomovision). The program’s path and name will be placed in the “Open” field. You can then add the arguments you wish to use after the quotes in the “Open:” field (-z in this example).

Start the program from a shortcut



Shortcut to TomoVision_License.exe

Target type: Application

Target location: Tomovision

Target: ram Files\Tomovision\TomoVision_License.exe" -z

☑ Run in separate memory space ☐ Run as different user

Start in: "C:\Program Files\Tomovision"

Shortcut key: None

Run: Normal window

Comment:

With this technique, the command line arguments will be present each time you start the program from the shortcut. First, create a shortcut to the executable program by right-clicking on its icon and selecting the “Create Shortcut” option. Then, edit the “Target” field of the shortcut properties by right-clicking on the shortcut and selecting the “Properties” option. Under the “Shortcut” tab, add the arguments after the quotes in the Target field.