

DICOM CTN Release Notes

Stephen M. Moore

Mallinckrodt Institute of Radiology
Electronic Radiology Laboratory
510 South Kingshighway Boulevard
St. Louis, Missouri 63110
314/362-6965 (Voice)
314/362-6971 (FAX)

Version 2.10.0
August 3, 1998

This document contains release information for the DICOM CTN software.

Copyright (C) 1993 - 1997 RSNA and Washington University.
The software and supporting documentation for the Radiological Society of North America (RSNA) 1993 - 1996 Digital Imaging and Communications in Medicine (DICOM) Demonstrations were developed at the

Electronic Radiology Laboratory
Mallinckrodt Institute of Radiology
Washington University School of Medicine
510 S. Kingshighway Blvd.
St. Louis, MO 63110

as part of the 1993 fluoroscopic- 1996 DICOM Central Test Node project, for, and under contract with, the Radiological Society of North America.

THIS SOFTWARE IS MADE AVAILABLE, AS IS, AND NEITHER RSNA NOR WASHINGTON UNIVERSITY MAKE ANY WARRANTY ABOUT THE SOFTWARE, ITS PERFORMANCE, ITS MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR USE, FREEDOM FROM ANY COMPUTER DISEASES OR ITS CONFORMITY TO ANY SPECIFICATION. THE ENTIRE RISK AS TO QUALITY AND PERFORMANCE OF THE SOFTWARE IS WITH THE USER.

Copyright of the software and support documentation is jointly owned by RSNA and Washington University, and free access is hereby granted as a license to use this software, copy this software and prepare derivative works based upon this software. However, any distribution of this software source code or supporting documentation or derivative works (source code and supporting documentation) must include the three paragraphs of the copyright notice.

Version 2.5 New Features (over version 1.4.1) and Information

1. A new *TBL* facility was developed to provide a general *API* to relational databases. A very limited set of functions were implemented (select, update, delete, insert). This version of the software includes a version of *TBL* that uses a Sybase database. Most of the demonstration applications generated for the 1994 RSNA demonstration use this facility and, hence, Sybase. Groups using a different database product should be able to rewrite the *TBL* facility to incorporate their own database. We are currently investigating the issues involved in including a public domain database so that this software can run without any licensing restrictions. Through proper configuration, you can compile and link the software without Sybase. Applications will run but will exit with error messages when they try to access the database. The utility and diagnostic programs do not use a database product and will compile and run without Sybase.
Note: See item 1 under *version 2.6*
2. A new facility (*IDB*) was created to provide an image database. This facility is based on the *TLB* facility and thus does not rely on Sybase directly.
3. A set of configuration routines were implemented in a common package to be used by demon-

stration applications for operations, like mapping AE titles to host names and port numbers.

4. Added a new facility (*FIS*) for providing some information system functions. This provides a rudimentary HIS or RIS, but is not designed to be complete. Hence, the name FIS stands for Fake Information System.
5. Several new demonstration applications were created. These are described in detail in *User's Guide for CTN Demonstration Applications*. In brief, these are:
 - a. `image_server` A new server program that uses the new image database implementation.
 - b. `pmgr_motif` A print manager program with a Motif GUI
 - c. Work List Manager An application for sending work list information to modalities (study scheduled events).
 - d. Results Storage An application for sending results (reports) to workstations.
6. A configuration program with a Motif GUI (*cfg_ctn_tables*) was created to give the user a simple method for configuring applications. This means the user does not have to edit tables by hand or understand the syntax of a database language.
7. The *SRV* facility which provides the basis for our implementation of SOP classes has a partial implementation of C-GET. This coincides with a partial implementation of C-GET in *image_server*. This will be more completely implemented and documented in a future release.
8. The old version of *img_server* and *img_client* have been retained for compatibility with the 1.4.1 release. These will be phased out in the near future. There is no new development for these applications.
9. Many of the CTN demonstrations and configuration programs use Motif. We believe you will need X11R5 to support the Motif applications. Versions 1.2 and higher of Motif should be sufficient to support our applications. (Try what you have.)
10. We did most of our development work using Solaris 2.3. We compiled the code using the Motif software supplied by Sun. We have also compiled the code under SunOS using a third party copy of Motif and under OSF/1 on an Alpha workstation.
11. We added a feature to the image server that is not standard. We map the following attributes to upper case upon insertion to our database and when processing queries from clients.
 - a. Patient Name
 - b. Patient ID
 - c. Accession Number

d. Study ID

12. The Motif based programs were developed using a code generation tool (*uimx*). You will see a number of strange files in the source directories that were generated by our tool. You can ignore those files. You do not need *uimx* to compile these applications.
13. This software supports the image IODs that are documented in Version 3.0 of the DICOM standard, P.S. 3.3-1993. This means that the software includes the definition of NM images that is about to be rewritten. This software does not support the X-ray angio or fluoroscopic image objects.
14. This release of the software does not support DICOM Part 10 files. Information objects (images) are stored using the implicit little-endian transfer syntax with no meta-header information. This is the format of the dataset as it is transmitted across the wire in a C-STORE operation.
15. This release of the code contains some additional tools that can be used for exercising DICOM functionality. There is a Motif-based program for examining image information objects, a Motif-based program for initiating associations and some network software for examining network traffic from an external workstation. There is also a Motif-based program that can perform some simple queries of a server that implements the DICOM query/retrieve (C-FIND) SOP classes. The current implementation supports the patient root information model.
16. Bug reports should be mailed to:
dicom_bugs@wuerl.wustl.edu
General DICOM/RSNA questions should be mailed to:
dicom_help@wuerl.wustl.edu

We received a number of explicit bug reports in the last year. We will list them in this section with comments. If your bug report is not present, it may be due to oversight in documentation. I hope we have addressed all of the reports that we received.

1. Some application makefiles included `-lmalloc`. This is used for debugging in our laboratory and should not have been released.
2. `apps/displays/image_utils.c`
Required initialization of some strings to work around a bug in the AIX ANSI C compiler (c89). The bug occurs when previously defined static strings of length greater than 19 are used to initialize structures. Initializing such strings to "" seems to prevent the bug occurring when the structures that refer to them are subsequently initialized.
3. Some of the user interface functions for the previous his/pacs demonstration used 0 and NULL incorrectly. This has been repaired.
4. `Apps/print_clients/display_icon.c`: Action routines are defined incorrectly. The 4th parameter

should be of type (Cardinal *) not type (int). This applies to functions `my_highlight` and `Multi_select`.

5. `apps/print_clients/display_message.c`: The 3rd parameter of the `displayMessage` routine is defined incorrectly. It should be of type (XtCallbackProc) not type (XtCallbackProc (*) ()). Type (XtCallbackProc) is already a pointer to a function
6. `facilities/dulprotocol/dulfsm.c`: Added include file `<sys/select.h>` for AIX compilation.
7. `facilities/dulprotocol/dulprotocol.c`: Added include file `<sys/select.h>` for AIX compilation.
8. `facilities/messages/dicom_messages.h` & `include/dicom_messages.h`
`facilities/print/dicom_print.h` & `include/dicom_print.h`
Required removal of trailing comma from enumerated type definitions. I know ANSI C allows trailing commas for initializers but I don't know if they are allowed for enumerated type definitions. Anyway either this is a bug in the AIX ANSI C compiler or other compilers don't care about the trailing comma! Changing it to remove the comma does no harm.
9. `facilities/objects/dcm.c`
The AIX version of the `swab` function takes parameters of type (short *) which kind of makes sense since you can only swap bytes of 16bit (short)things. The call to `swab` needed to be changed to cast the parameters to type (short*).
10. HPUX does not support the regular expression calls that we used, `re_exec` and `re_comp`. We received some fixes from users, but have not been able to compile them in our laboratory to test them.
11. HPUX does not support the `getwd` function that we used in several places. We replaced it with `getcwd`.
12. `facilities/1st/1stcond.c`: Incorrectly checked for termination of an array of messages.
13. The `img_server` and `img_client` programs work with `ctndisp` to display data as images are sent across the network. These applications use shared memory and semaphore resources that are not released when the programs exit. This was planned, because we wanted to leave the queues in place. To remove the resources, you need to use the system `ipcrm` command or the application `gqkillq`.
14. Changed the typedef `BOOLEAN` to `CTNBOOLEAN`. We were running into other packages that used that type and found it easier to change our definition.
15. Changed some of the macros found in Makefiles to provide consistent naming for where to find things, like `LIBPATH_MOTIF`, `CFLAGS_MOTIF`.
16. Made some of environment variables for Solaris have absolute path names (e.g., `install`). We found conflicts between the UCB version that we are used to using and the Sys V ver-

sions that other people use.

17. facilities/uids/dicom_uids.h:

The struct `UID_DESCRIPTION` has a field called `class`. This created problems when I included the file in C++ files. This has been changed.

18. The new image server does not support image level finds or moves with lists of image UIDS.

19. facilities/srv1.c: `SRV_Request Service Class`

On line xxx there is a malloc call for a `DUL_PRESENTATIONCONTEXT` structure that is assigned to pointer variable `ctx`. A pointer to `ctx` is then passed into `DUL_MakePresentationCtx` which promptly allocates another `DUL_PRESENTATIONCONTEXT` structure and overwrites the first allocated pointer (it thus being lost).

20. facilities/dulprotocol/dulfs.c: `destroy PresentationContextList` had a memory leak in that a sublist was not being destroyed properly when the list was deleted.

21. facilities/dicom/dicom.h

In the header file `dicom.h` `DICOM_CS_LENGTH` is defined as 12. In the final draft DICOM specs I have Part 5: Section 6.2 the Code String type is defined to have a maximum length of 16.

Version 2.6 New Features (over version 2.5) and information

This release is designed to be the first release of CTN software to participants in the 1995 RSNA DICOM demonstration. We have corrected some bugs found during last year's demonstration and have incorporated fixes that we have received from organizations that obtained the CTN software after the 1994 RSNA annual meeting. This section will list the additions to the 2.5 version of the code. There are some more corrections to be made.

1. The TBL facility was augmented to support a public domain database, miniSQL. We include the miniSQL distribution.
2. The IDB facility was updated to better support the DICOM information models. This is designed to improve performance for Query/Retrieve using the Study Root Model.
3. We started to add support for the C-Get operation in version 2.5. There is on going work.
4. There was a bug in our interpretation of RGBA images and how they are displayed. This has been corrected.
5. The GQ facility was modified to be able to access multiple queues from an application

simultaneously. Previous versions allowed access to only one queue at a time, forcing you to close one queue before accessing another queue.

6. The function `DMAN_ClearList` in `facilities/manage/control.c` did not always return a value.
7. In the function `SRV_CGetResponse` in `facilities/services/get.c`, the response `Count` variable was declared incorrectly (both in type and where it was declared).
8. In the function `SRV_NEventReportRequest` in `facilities/services/nevntreport.c`, a call to the user callback function was made with the wrong level of indirection on a pointer to a structure.
9. In the function `processStoreRequest` in `facilities/services/get.c`, the first call to the user callback function was made with two uninitialized variables. The variables are now initialized.
10. In `SRV_ReceiveDataSet` in `facilities/services/srv2.c` there was a small memory leak when the caller of the function did not specify a directory for storage of temporary files.
11. Some field names in the FIS facility were modified due to limitations in the `miniSQL` package.

Version 2.6.1 Release Information

Version 2.6.1 of the CTN software is a minor release which corrects dictionary related items and adds some error correction. Corrections are:

1. Update data dictionary to support elements defined for June 30 Draft for Letter Ballot of NM Image Object. The 2.6 version supports both the obsolete version and the updated version.
2. Update sequence definitions to handle new NM Image Object.
3. Correct error in Information Entity facility where the type for study date and study time was set to type 2 rather than 2C. this would have shown up in `dcm_verify` or when the `image_server` program verified images. In version 2.6, we turned off image verification in the `image_server`. It is turned back on in version 2.6.1.
4. Updated error checking in `IDB_InsertImage`. Better checking allows us to detect failure when trying to insert the image instance.

All documents in the 2.6.1 release will maintain the 2.6 version number. There are no documentation changes.

Version 2.6.2 Release Information

Version 2.6.2 of the CTN software is a minor release which corrects errors as listed below.

1. The *IDB_Select* function had a bug when implementing the Study Root model that caused it to not return any rows.
2. In *apps/assoc_tool*, there were function calls to Motif functions that lacked the proper number of arguments.

All documents in the 2.6.1 release will maintain the 2.6 version number. There are no documentation changes.

Version 2.7 New Features (over version 2.6.2) and Information

Version 2.7 of the CTN software contains bug fixes as detailed below and is our first release of software which reads and writes files that conform to the DICOM file format defined in Part 10 of the DICOM standard. Many/Most applications that have previously read “CTN format files” will now read DICOM Part 10 files as well. Most of these apps will try CTN format first and then will try Part 10 if that fails. You can usually have the app try Part 10 files exclusively with the -t switch. We still have work to do on the big-endian transfer syntax, but we have made progress on the little-endian transfer syntaxes (both implicit and explicit) in the context of Part 10 files.

Other features/bugs fixed in this release are:

1. Included a new application *dcm_print_dictionary*. This application prints the version of the DICOM data dictionary that is maintained by the DCM facility.
2. Fixed a couple of bugs in the IDB select functions when working with the Study Root model. One bug prohibited the user from getting data below the study level; the second bug led to duplicate records being returned.
3. Added support in SQ facility for Modality LUT Sequence, corresponding to element with tag (0028, 3000).
4. Modified the DCM facility when exporting sequences. We have a bug in our software that caused us to compute the length of a sequence incorrectly. When we exported the sequence, this caused big problems. The short term fix was to disable our ability to send sequences with known lengths. We do this now by sending sequences with unknown lengths and appropriate delimiters. As time permits, we will correct the length computation.
5. Some of our Motif apps were incorrectly coded as they invoked the Motif function *XmToggleButtonGadgetSetState* with too few arguments. We fixed the calls.
6. *TBL_Delete* (sybase) would generate a core-dump if the terminator on the delete criteria was not a field item with a NULL field name. Other places in the TBL functions allow us to terminate criteria items with a NULL criteria item (rather than a criteria item with a NULL field name). We modified the delete function to be consistent with the other parts of the system that allowed the list to be terminated in either manner.

7. We updated the data dictionary to support attributes defined for new image objects:
 - Supplement 4 - X-Ray Angiographic Image Objects & Media Storage
 - Supplement 5 - Ultrasound Application, IOD & Transfer Syntax Extensions
 - Supplement 6 - X-Ray Radiofluoroscopic Image Object
 This also involved correcting some entries in the dictionary.
8. Some attributes (smallest pixel value in plane, ...) can be either signed short or unsigned short. When coding or decoding these, our implementation uses a context sensitive lookup (i.e., we check the type of the pixel data). We updated our implementation to include some additional attributes.
9. Added a -o switch to *fillImageDB* so that we could specify an owner at the different levels in the database. Documented the use of *fillImageDB* in the user's guide under the *image_server* discussion.
10. Added two attributes to our database implementation:
 - Series Level: View Position
 - Image Level: Patient Orientation
 This involves placing the attributes in structures defined by the IDB facility, modifying code in the IDB facility and modifying the parse routines for *image_server* and *fillImageDB*. This also means that databases that were created with Sybase or msql scripts before version 2.7 will need to be discarded and recreated with new scripts (as provided in this release).
11. Fixed the *image_server* so that -q really does mean quiet mode. We still had some verbose output when moving images.
12. Added a new app, *dcm_ctnto10*, which converts files from traditional CTN format to DICOM Part 10 format. The warning (should be in caps) is that this is our first attempt at Part 10 files, so there are likely to be some bugs. Please be wary of the output. Also, we strongly believe that we have problems in our big-endian implementation for pixel data, so be extra cautious when converting to big-endian format.
13. Retired the *img_server* and *img_client* applications. These were using our older implementation of a simple database that we wish to retire. That database facility (DB) will be retired as soon as we move the print applications to the new database facility (TBL or something on top of TBL).

Version 2.7.1 Bug Fixes

Version 2.7.1 contains bug fixes and no new features.

1. The TBL facility does not report the database specific error which occurred (e.g., Insert failed because column does not exist). TBL_Insert (msql) has been updated to place additional information on the stack to help trace down insert errors. This should be extended to the other features and to the Sybase implementation as well. No promises are made, however.
2. When the msql scripts for creating FIS tables were written, we had to change some table names because msql (as delivered) has a limit on table (and probably field names). The Sybase scripts FIS_Open code were not updated to be consistent with these changes. That is corrected in this release.
3. DCM_GetElementValue had a bug which occurred when the caller was retrieving data but did not provide buffer space for all of the data (a condition that we expected) and the length of the

data to be retrieved was not an integral multiple of the buffer length. We had a logic error that produced an error under these conditions which is corrected in this release.

4. In *query_client*, the query at the SERIES level did not include an empty attribute for the Series Instance UID (0020, 000E) as required by the standard.
5. *pmgr_motif* was dumping core because we were incorrectly freeing XImage structures that we needed to maintain for use with our image icons.
6. The DB facility is still used to create our print database to be used by *pmgr_motif*. It was hard-coded to limit us to 20 studies, causing problems for RSNA '95 where we want a print database with 30-40 studies. We upped the parameter to 60 studies.
7. We have a bug in our DCM facility that causes us to incorrectly compute group lengths for groups that contain sequences. We were also including group lengths inside of sequences. This means that the images that were shipped with the 2.7 tapes had these errors and/or if you used our software to send images, you would see these problems (as we recompute these things). We have changed our software to turn off the inclusion of group lengths when sending images (but have retained the group length in group 0000). This was first accomplished by changing the IAP facility to use the DCM_NOGROUPLength flag. Since the problem happens with other apps that might rewrite a file, we then changed the DCM software to turn off exporting of group lengths altogether except when exporting the command group (0x0000) as required by Part 7 of the standard.
8. The miniSQL database implementation has a bug in processing strings which include wild-cards and ^. In fact, we may have problems processing any string that includes ^ as part of the query. The symptom that was reported (and that we can repeat) is that a name search for “*^*” returns the same values as “*”. We are investigating the problem (i.e., we have not corrected this one yet) It does not happen in our Sybase implementation.
9. The scripts that we use for creating print databases were updated to reflect the images that were sent with the 2.7 release to vendors. Our print setup software does not handle MONOCHROME1 images properly, so those images are currently not included in the print set.
10. The script *fillImageDBScript* uses *fillImageDB* to load a set of images into the relational database used by the image server. There was an error in the script such that we did not include absolute path names for the image files as they were entered in the image database, making it difficult for the *image_server* application to perform a move (because it did not have the absolute path to the images).
11. The DUL documentation contained some errors. Specifically, we had not fully documented the fields in DUL_ASSOCIATESERVICEPARAMETERS. While we were repairing that documentation, we corrected some other typos in the document.
12. All documents retain the version level 2.7 with the exception of the *User's Guide to the DUL Facility*. It was renumbered 2.7.1.
13. The script *icon_script* checks for the existence of a destination directory and exits if that directory does not exist. There was a typo in the script which reported the wrong directory if that destination directory did not exist. We also updated the script to create the destination directory if it does not exist. However, this should have been done in an earlier step by the script *create_icons*.

Version 2.8

Version 2.8 is a cleanup of some of the problems we saw during final testing for RSNA as well as a few new features. There are no major changes in the design.

1. The TBL facility does not report the database specific error which occurred (e.g., Insert failed because column does not exist). TBL_Select (mysql) has been updated to place additional information on the stack to help trace down insert errors. This should be extended to the other features and to the Sybase implementation as well. No promises are made, however.
2. The DUL facility uses a constant to determine the length of the “listen” queue when accepting socket connections. That value had been limited to 5, causing some acceptor applications to miss association requests. We increased the parameter to 50. Note that on Solaris there is another system wide variable that needs to be set using the ndd(1M) command. The listen backlog parameter is called “tcp_conn_req_max” and is in the /dev/tcp set of parameters. The only time we have this problem (and need to increase the parameters) is when we are accepting a lot of connections at the same time (as at RSNA demonstrations).
3. Corrected some incorrect documentation in installation guide.
4. Changed the environments directory layout slightly. We added another layer so you could find all of the files for a particular OS in one directory (e.g., sunos, solaris-2.3, solaris-2.4) rather than having all of the files thrown into a single (large) directory.
5. Modified the definitions of the SRV functions that require user-supplied callbacks. We had several suggestions that we should provide prototypes for the callback functions to aid users in developing code. We took the suggestion and added the callbacks. The only problem that we ran into was in the usage of context variables in our callbacks. Because of the prototypes, we defined these as pointers to void. We wrote our callbacks with these as pointers to void and then did a cast inside of the callback.
6. The document page which described SRV_CMoveResponse was totally incorrect. We fixed that page. While we were at it, we cleaned up some of the documentation in other parts of the SRV functions. These fixes were usually minor ones.
7. We received specific assistance from one user in updating our data dictionary to support items in group 0x0004. We still have not implemented DICOM DIR functions, but the dictionary entries are now there. The same user also sent some comments about reading image files with other transfer syntaxes (like JPEG). We have not made any progress on that front.

Version 2.8.1

Description of 2.8.1

1. We modified the DCM facility by adding a flag to the options of DCM_OpenFile. The DCM_USELENGTHTOEND flag is used when you want to open (older V2) files that use the length to end attribute (0008 0001). A vendor was giving us V2 data with extra bytes on the end. The vendor indicated that they computed length to end and that we should use that rather than the length of the file maintained by the file system. Because of this vendor’s implementation, we assumed that a file that required us to check length to end would also have the group

length attribute (0008 000). Note that all of this goes away when we start doing real DICOM with this vendor.

2. The program *dcm_modify_object* is used to modify attributes in existing files. We advertised the ability to modify the pixel data. However, the previous (incorrect) behavior was to add a second pixel element attribute after the first one. We have corrected this so that we actually replace the old pixel data with the new pixel data.
3. We have code that takes the address of a character array by placing an & in front of the array. i.e.,

```
char c[1024];  
char *b = &c;
```

Most compilers let this slide with a warning, but the SGI compiler generates an error. We fixed the problem in *messages/messages.c*.

4. Coding error in *sq/sequences.c*. The dictionary entry for the sequence NM Number of Frames in Phase had an errant pointer to a variable in a different structure. This meant that any parsing or creation of that sequence would have been in error.
5. Declaration conflict for *re_comp*, *re_exec* in *utility.h*.
Utility.h explicitly defines prototypes for *re_comp* and *re_exec*. *idb.c* included *<unistd.h>* which also defined *re_comp* and *re_exec* on SGI machines. This caused the SGI compiler to pull an error when compiling *idb.c*. We repaired this by not including *<unistd.h>* from *idb.c*. Note: We still need to do some work on the regular expression code since this causes problems on HP and other machines.
6. The *query_client* application did not handle query responses that contained warnings about optional keys not being supported. The behavior was that the *query_client* receiving C-FIND responses but not updating the display after the search. We changed the application to handle status values with warnings as well as statuses of success.
7. The installation process for *facilities* and *apps* has assumed that the target directory (library directory or binary directory) existed before the installation began. This was a bad assumption and not documented. We modified the installation *Makefiles* to check for the target directories and to create them before installation of the libraries or binaries.
8. Server processes (like *simple_storage*) that want to listen on the suggested port 104 did not execute with the *setuid* bit on Solaris systems. This was because the system used shared libraries and Solaris did not consider *libucb.so* to be secure. We fixed this problem by adding another macro in the global makefile called *LIBS_OS_SECURE_UCB* which forces the linker to link with a static copy of *libucb*. Ugly, but it works.
9. Increased a buffer size in *formatStorageAccess* in *apps/cfg_ctn_tables/format.c* so that we did not write over the end of it.
10. Fixed a bug in the *image_server* when storing images as a result of a C-MOVE request. One of the parameters in a store request is the Move Originator Application Entity Title. We had filled in the title of the image server. We corrected that field to contain the title of the application that originated the C-MOVE request.
11. Added code to use the *regcomp* and *regex* functions found on HP-UX and other systems. To trigger this, define the macro *USEREGCOMP* at compile time. We define this as one of our CC flags in the global Makefile. See the OSF environment files for an example.
12. Added a set of environment files for AIX. These were contributed by a user working with the 2.8 release.
13. Fixed a parsing problem in *pmgr_motif* that caused a core dump.

Version 2.8.2

1. This version now compiles under HPUX (9.x).
2. Some of the application Makefiles listed libraries in the wrong order. They listed the Motif and X11 libraries before the CTN libraries which actually used them. This caused the HPUX and Irix linkers to complain.
3. Added code to use the regcomp/regexec functions for regular expression matching. This was necessary for HP systems. This code is turned on by defining the macro USEREGCOMP in the global makefile. That is, -DUSEREGCOMP. See the hpux environment files for an example.
4. All other docs retain the 2.8.1 release number.

Version 2.8.3

1. Added prototype for definition of callback function in DCM_ExportStream. This satisfies C++ compilers.
2. Added parse/build functions for the VOI LUT sequence. Changed the structure used for the VOI LUT sequence. We had previously defined a structure, but had no code to parse these sequences.
3. Added new functions in SQ facility to release structures and the list created by SQ_ParseSequence. The implementation still needs some work, but this gives the user a uniform method for releasing these structures.
4. Satisfied a compile problem in the GQ facility under Linux. Harry Kobetitsch/Picker International submitted an explanation and a fix because *there is a different argument list for semctl than in the traditional semctl*. This does not imply that we support Linux. I do not have access to a Linux machine to test the code.
5. Added some code in headers to satisfy C++. This consisted of getting *extern "C" {}* wrapped around the code with appropriate ifdefs.
6. Added -x flag to simple_storage so that the default directory for creating images could be changed with a command line argument.
7. Corrected some errors in the MSG facility for optional and required values in N-Action, N-Create and N-Delete Responses. We had some attributes that were coded as required when they should have been optional. We also had some attributes in the structures that did not belong. These changes caused us to correct related errors in *print_server*, *pmgr_motif*, *print_client*, *print_mgr*.
8. Fixed a bug in the DCM facility that caused us to not write the group length for group 0002. This meant that the *dcm_cntto10* application wrote illegal DICOM part 10 files that could not be ready by our own software (*dcm_dump_file*).
9. Fixed documentation in *send_image* so that comments in the code and in printed documents matched the implementation.
10. Relaxed a test when exporting dcm objects in *facilities/objects/dcm.c*. This test kept us from exporting 0-length sequences.
11. Added code in *facilities/services/send.c* to eliminate the sending of runt PDVs. This is accom-

plished by calculating the length of the object before we export it into the PDV. You turn this on with a compile-time option (-DCTN_NO_RUNT_PDVS). With the current implementation, there will be a slight performance hit, so I would recommend turning this on only if you need it.

Version 2.8.4

The main reason for this release is to publish a version of the software that runs under Linux. There are some other changes noted below.

1. This version compiles under Linux. I used:
Caldera Network Desktop - release 1.0
Red Hat Linux - release 2.1
for my development. My thanks to Henry Hollenberg for providing a test environment.
2. Added more macros to compile environment because the Linus system had flex and not lex. In the environment file, defined LEX to be the command that invoked lex (or flex). In the Makefile, defined LIBS_LEX to be the library needed to link applications that used lex (-ll or -lfl).
3. Did some work on adding multi-thread support on the network software. This is being tested under Solaris but is not considered complete. You have to define a macro to turn this on (-DCTN_USE_THREADS), so it won't cause problems for you if you don't enable it.
4. Release Notes and Installation Guide renumbered to version 2.8.4. All other docs stay at 2.8.3.

Version 2.8.5

This version contains several bug fixes. There are no new major features.

1. Corrected a logic error in *dcm_x_disp* that caused the program to ignore command line values for Window Center/Width if the image did not originally contain those parameters.
2. Modified *objects/dcm.c* so that 0-length sequences can now be parsed successfully. Prior implementations required sequence items of known length to have at least one attribute.
3. Corrected an error in *IDB_Delete* where the function did not properly iterate through the database to delete items below the starting level of the delete operations.
4. Added ability in *UID_NewNumber* to request a new patient number (not a UID).
5. *dcm_map_to_8* repaired to properly pad pixel data in images with an odd number of pixels.
6. Corrected bug in *DMAN* facility so that the image_server could correctly interpret -n flag and override default node name.
7. Removed references to obsolete studies in *create_icons* script.
8. A number of cosmetic changes to make the code compile on Windows NT using Visual C++. We were trying to get rid of most if not all of the warning messages generated by the VC++ compiler.
9. *Release Notes*, *Installation Guide* and *Conformance Statement* renumbered to version 2.8.5. All other docs stay at 2.8.3.

Version 2.8.6

Miscellaneous bug fixes and changes to distribution of miniSQL.

1. The author of miniSQL has changed the licensing terms for his software. In brief, academic institutions may still use it for free, but others are required to pay a small licensing fee. We have also decided not to include the software with our distribution. The installation guide will tell you how to obtain the software.
2. Changed macros SUCCESS, INFORM, WARNING, ERROR, FATAL in *dicom.h* to CTN_SUCCESS... We ran into conflicts with similar declarations by other systems. This was long overdue. If your code uses the older macros, the easiest way to find the places is to let the compiler do it. Uncomment the definitions of the old macros in *dicom.h* to force compile time errors (rather than the link time errors because it thinks that ERROR is an undefined function).
3. Added some additional error messages in *image_server* when an image cannot be stored by the server. Before, there were cases where the user had to guess why *image_server* rejected an image.
4. Corrected a table problem in the MSG facility that caused the CTN to send an incorrect attribute in an N-EVENT-REPORT-RQ message. We had been (incorrectly) transmitting (0000, 0003) Requested SOP Class UID and are now transmitting (0000, 0002) Affected SOP Class UID.
5. A number of (cosmetic) type changes in local variables to get types to match up with function prototypes. We have started to convert the code for MSC, and that compiler seems to be less forgiving than our usual compilers.
6. Accepted a number of changes in the display programs (ctn_x_disp, ctndisp) from Roddy McColl. These programs now work with 24 bit color displays. Our original code only worked with 8 bit (greyscale or pseudocolor).
7. Corrected one case where the *image_server* program could go into an infinite loop if the peer sent an illegal command.
8. *Release Notes* and *Installation Guide* renumbered to version 2.8.6. All other docs stay at previous levels. With this release we will make it a policy to only renumber a document when it changes. This means that document numbers will only match release numbers when a document changes. Many of the documents have not had real changes in quite some time.
9. Included new material in *Release Notes* on Linux installation. We had contributions from two independent sources (Bill Chimiak and Henry Hollenberg) who performed installs and gave us notes/scripts that make the job simpler on Linux.

Version 2.8.7

Miscellaneous bug fixes and library updates.

1. Repaired bug 96-26. The image display programs need to allocate color map entries using X Window calls. They also need to compute gray scale values based on the number of entries

that can be allocated. In cases where we could only allocate one color map entry, we had a divide-by-0 error. We now print an error message (and assume that the user will close some other application that is taking the color map entries).

2. Repaired bug 96-27. The MAX PDU value established during association negotiation is used to limit the amount of data sent in a P-DATA PDU. We have software that checks to see if we received a PDU that violated this constraint. We also applied the same test to other PDUs, but should not have done so. We now only do this for P-DATA PDUs. <p> More specifically, a vendor told us of someone else's implementation they had seen in their labs that had a small max value (2K). However, it had to work with A-Associate AC PDUs or A-Associate RQ PDUs that were greater than 2K in length.
3. Repaired bug 96-28. *query_client* makes a patient level query and requests the Patient Birthdate. If that attribute was not returned by the query SCP, *query_client* would ignore the response and not place results in the user interface. Since Patient Birthdate is an optional attribute for query SCPs, we have corrected the behavior of the application. It now continues and prints a blank on the user interface where we would have printed the Patient Birthdate. If the SCP provides a Patient Birthdate, *query_client* will still display it.
4. A number of updates were applied to the data dictionary to bring it up to date with the version of the DICOM Standard to be published in early 1997 (DICOM 3.0 - 1996). This included adding entries, correcting typos in some of the attribute names, and retiring attributes. We changed our philosophy on retired attributes. We now try to maintain the VR in the dictionary and place RET in the description. This will allow us to still perform dictionary lookups and get the proper VR for attributes which we will be seeing in older images.
5. The IE facility had some errors in its table definitions. This facility is going to be phased out over time.
6. Modified the DCM_DumpElements function to take a count argument. This allows us to dump binary elements with a VM of > 1. We used to just print the first value. Modified the dcm_dump_file to use this new argument. *This change in signature is a problem if you ignore compiler warnings when building your software that uses this subroutine. We modified a lot of our code to get the correct number of arguments when calling this function.*
7. Added functionality in FIS which will be used to support storage commitment requests and general work requests. This release has the updates in the subroutine libraries, but we are not yet releasing the applications that do the work.
8. Modified the schemas in the FIS to support storage commit requests and work queues. We added three new schemas. You will want to recreate your databases when building this release.
9. Added functions in the *utility* facility to support timing measurements. These are some helper functions so we don't have to remember the unix time conversions each time we want to time an operation. Still need to update the documentation on this facility.
10. Modified the *send_image* program. A new option (-p) allows you to send images with the pixel data replaced by a single unsigned short pixel. This is useful for tests where we wanted to get header information into our databases but did not want to send all of the pixel data to do this.
11. Corrected an error in *load_control*. It was not parsing input files correctly for the case of loading print server configuration information

Version 2.8.8

Miscellaneous bug fixes and library updates.

1. Made a number of changes to FIS facility and the *cfg_scripts* that create the FIS tables in the miniSQL and Sybase databases. We added some attributes to the Results and Interpretation tables and some event notifications in the Detached Results and Detached Interpretation Management classes supported by FIS.
2. Added *FIS_DumpRecord* function to FIS facility. This dumps a record using formatted output to a file.
3. Added *FIS_SelectOne* function to FIS facility. This function searches some FIS tables using criteria other than the unique IDs found in the tables. The first implementation allows us to search the Study Table by Accession Number.
4. Made a change in the *SRV* facility so that it better supports the Detached Results Management Meta SOP class.

Version 2.8.9

1. Fixed a bug in *dcm_cntnto10*. The previous version ignored the switches that were supposed to allow the user to modify the byte order in the output file. This was corrected.
2. Updated *dcm_dump_element* to handle sequences. If asked to dump a sequence, *dcm_dump_element* dumps the sequence to a file in CTN format. The user can then examine the file using other CTN tools.
3. Fixed a bug in the mSQL version of the *TBL* facility. We were not properly detecting NULL columns in some rows. This was causing a segmentation fault, most notable in the application *cfg_ctn_tables*.
4. Modified the *dcm_print_dictionary* program to accept a command line switch to direct it to print the dictionary of UIDs that is maintained.
5. Updated the definitions in the *DCM* facility to support the Position Emission Tomography objects. These were defined in Supplement 12. We are working from a draft for ballot dated June 6, 1996. We have also updated the *SRV* facility and *simple_storage* and *image_server* applications to store these objects.
6. Updated the definitions in the *DCM* facility to support the Radiotherapy Objects defined in Supplement 11. We are working from a Final Text version, dated June 4, 1997. We have also updated the *SRV* facility and *simple_storage* and *image_server* applications to store these objects.
7. Updated the *IDB* facility during image inserts. We found a vendor was passing "\" in one of the attributes that we place in the database. That was their convention for a NULL attribute with a VM of 2. We replaced that with "" when we perform the database insert because the "\" was confusing the insert command into mSQL.
8. Modified the *DCM* facility for reading Part 10 files. After reading file meta header information, corrected how we set our flags for reading the rest of the file. For some cases, we should have been assuming Little Endian byte ordering and explicit Value Representation.
9. Updated the dictionary of UIDs in the *UID* facility to include the PET and Radiotherapy objects.
10. Added the function *UID_ScanDictionary* which gives the user the ability to scan the dictio-

nary of UIDs that we maintain. Modified the application *dcm_print_dictionary* to take another flag to print that dictionary instead of the attribute dictionary.

11. Modified *dcm_dump_element* to dump sequences. When the user asks for an attribute that is a sequence, the attribute is written out in CTN format. You can then use the usual CTN tools to view that file or extract the attribute that you need. Modified the *send_image* program. A new option (-t) allows you measure the time required to send the data and compute a transmission rate.

Version 2.9.0

This release marks the first addition of new capability in the CTN software in some time. In addition to bug fixes, we have added some applications.

1. Added a new suite of programs under the guise of an *image archive*. Complete details are included in the postscript documentation. We incorporated the Storage Commitment class and improved query capabilities by adding some attributes to query responses. We also added an application that will purge the archive of images. This set of programs is a direct replacement for the *image_server* which will no longer be supported. See the documentation for full details.
2. Changed the algorithm for generating permanent file names in the *DMAN* facility. This is used by the image archive to store images. The old mechanism (used by the *image_server* application) grouped all images in one study under a common directory. The new system creates one further directory level to store each series in a different directory. If you use the path names stored in the database to open the images, your application code will not change. If you wrote other applications that "knew" where the images were stored, you will have to make some modifications.
3. Modified the TBL and IDB facilities to better handle databases (mSQL) that don't support the full SQL standard. In SQL, you can update a counter in a field (e.g., $v = v + 5$). mSQL does not support that yet, so we had to create a solution for that. This was all done to support the *image archive* when it is acting as an SCP for queries. We wanted to provide the Study, Series and Image counts that are part of the Query SOP classes.
4. Repaired bug 97-2. We were mixing types of *long* and *int* in some of the subroutine libraries. This was causing problems on 64 bit machines.
5. Added a new set of configuration scripts for version 2 of mSQL. John Sanders of UNM sent in changes to support the new version of mSQL. We haven't tried version 2 in our lab yet. The new scripts are in *....cfg_scripts/msql2*. If there are problems with the scripts, blame us for transcription errors.
6. Repaired bug 97-5. We accepted Stephan Popp's suggested fix. When we parse a file with explicit VR and our dictionary says that the type is context sensitive (meaning US or SS), we will accept the VR encoded by the user.
7. Update *contributed/linux-chimiak/linux.configure* to include libraries ICE and SM during the build process.

Version 2.9.1

This version was an internal release.

Version 2.9.2

1. Incomplete updates for the conversion to Windows 95/NT. Some files require simple changes to include directives. Other modules (like network I/O) require larger changes. Documentation is incomplete, but you can get a start by reading `.../doc/windows.txt`.
2. Updates to data dictionary to support Visual Light Supplement 15 and Waveform Supplement 30.
3. Updates to the DCM facility to be able to read DICOM Part 10 files with compressed data. This still needs some work to effectively handle the compressed data. At this version, we can at least read files with compressed pixels and dump the header.
4. Updates to the SQ facility to handle sequences defined in the Waveform Supplement. Still considered a works in progress.
5. Updates to `dcm_modify_elements`. Modified the application so that the output is always written in CTN format, default Little-Endian transfer syntax. Fixed the problem with original pixel data not being deleted if the user tried to add pixel data.
6. Add a security override to `archive_server` that allows the user to allow a client to connect to multiple servers on a node with a single entry in the Security Matrix.
7. Fixed a bug caused by the interaction of `mysql` and IDB facilities. Searches on patient names containing '^' were failing because '^' is a special character in regular expression matching, and we were not escaping it properly before passing it on to `mysql`. This problem does not show up in `sybase`. The fix is a new compile time macro: `TBL_REQUIRES_HAT_ESCAPE` that is needed if you are using `mysql`. This will be defined in `$DICOM_MAKE`.
8. Modifications in `apps/displays` to support display with MS Windows. Changed some of the files so that they did not rely directly on X-11. Still needs more work to make this all compatible with Windows.
9. Modify `archive_server` so that it takes another switch (`-r`) for reduced capability. This allows the application to run without using the FIS database.
10. Correct a bug in the `archive_server`. Storage commit requests were received but not placed in the work queue for processing.
11. Begin updates to the `idb` facility to store a new table in the image database. When implemented, this table will allow us to set limits on the size of the database so applications can monitor the system.
12. Add the function `TBL_SetOption` to TBL implementations. This function will take a character string and allow users to configure the TBL facility (in lieu of using environment variables.)
13. Modified the DCM facility to take another flag when opening Part 10 files. Now we try various combinations of file names to match what we have seen on ISO 9660 disks.
14. Added the function `DCM_GetElementValueOffset`. This allows the caller to specify an offset into the data before we start to retrieve it. It is especially useful for multiframe images (uncompressed).
15. Added special code to DCM facility for parsing/interpreting Waveform Data. The document is still in draft format, and the code is likely to change.
16. Added `-t` flag to `dcm_ctnto10`. This allows us to read Part 10 files as input and change the transfer syntax on output.
17. Added the application `dcm_dump_compressed`. This application reads DICOM Part 10 files with compression and dumps the compressed blobs to separate files.

18. Correct a bug in archive_agent. It was not releasing the association after processing a work record.
19. Update the print_server application. We should not have been returning Print Job information as a result of the N-Action print request because we do not support the Print Job class.

Version 2.9.3

This version of the software was released for vendor testing prior to the 1998 ACC DICOM demonstration.

1. Modified the applications fillImageDB and archive_server to support DICOM Part 10 files. The archive_server does not store files in DICOM Part 10 format. The fillImageDB program can be used to load the database with existing Part 10 files. When a remote application requests these images, the archive_server will open them and transmit them.
2. Modified the IDB facility and image_archive application to keep track of the size of the image database (that is, the number of bytes in the image files in the database). If the database exceeds some limit, image_archive will begin to purge studies until the database shrinks sufficiently. This is not the default behavior and needs to be turned on with a compile time MACRO. If you are used to the previous behavior, you should not see any changes. If you want to try this, the macro to define in \$(DICOM_MAKE) is -DCTN_IDBV2.
3. Added IAP_InfoObject to IAP library. This function is similar to IAP_SendImage, but works on DICOM Part 10 files rather than files stored in standard CTN format.
4. Changed signatures of some of the library functions so that some of the char * arguments are now const char *.
5. This feature was actually changed in 2.9.2, but we did not note it. The DCM facility is used to import/export information objects. When the user imports objects with elements that are not in our data dictionary, we do not know what to do about interpretation and byte order. When the user exports the data, the DCM facility just copies those elements as they were read. This is a problem when the data is read in one byte order and exported in a different byte order. The DCM facility used to do this silently. Now we print a warning message. We still need to provide an appropriate fix. This was not a big problem for the archive_server when it only supported the default little-endian transfer syntax. It still only supports that transfer syntax for network operations, but now you can use fillImageDB to place files in the database with a big-endian transfer syntax.

Version 2.9.4

This version of the software contains minor bug fixes.

1. Modified a number of the facilities. They define functions XXX_Message which are supposed to map a condition value to an ASCIZ string. Tables in these functions were not terminated properly; this could lead to an exception if you tried to lookup a condition value that was not

defined.

2. Corrected the dictionary entry for (0028, 2110) Lossy Image Compression. The proper VR is CS.
3. Updated simple_storage to support more SOP classes. Updated that application to print a warning message if it cannot map an SOP class to a directory name (MR, CT, ...). The program used to just hang when that happened.
4. Added -k switch to simple_storage. When you use this switch, files are killed (unlinked) immediately after the objects are received and acknowledged.

Version 2.9.5

This version of the software contains minor bug fixes.

5. Corrected a subroutine library bug that manifested itself in archive_server. Because of the interaction of the DCM and IAP facilities, we were not properly parsing Part 10 files with sequences when we were preparing to send them out of the system (in response to a C-Move request). You had to have loaded Part 10 files into the archive by hand using the 2.9.3/2.9.4 releases to have seen this bug. If you loaded the archive by sending images to it (send_image or other means), this problem would not occur.

Version 2.9.6

This version of the software contains minor bug fixes.

1. Corrected a bug in IDB facility. We were not returning a value for the number of images in a series in a series level query.
2. Corrected a memory leak in the PRN facility. The functions made a call to SQ_BuildSequence to create a DCM_ELEMENT structure and then a call to DCM_AddElement. This results in a memory leak. Short term fix is to add a function DCM_AddSequenceElement and have caller free the element allocated by DCM_BuildSequence.
3. Corrected a number of memory leaks that showed up in the application pmgr_motif. Some were in the application. Most were in our DICOM facilities (as described below).
4. Repaired memory leaks which were a result of the design of the SRV_N.. functions. The functions sent messages to a peer application and received a response (command and dataset). The functions were not deleting the dataset after the response was processed. This correction means that functions/applications that call SRV_N...Request functions need to examine the response data set in the callback function. We had left a hole that allowed the user to see this after the SRV_N...Request function was complete, but this was really a memory leak.
5. Repaired a memory leak in ICON_GetStudyOffset. A list was not destroyed properly.
6. Repaired memory leaks in PRN_BuildObject. We had some problems with freeing memory when working with sequences. Added a PRN_FreeStructure function that frees a structure created by PRN_ParseObject.
7. Added a function DCM_AddSequenceElement. This function is documented to take ownership of the list structure which was passed in the caller's DCM_ELEMENT argument. The

function DCM_AddElement can be used to add sequences to DCM objects, the but side effects of what happens with a sequence are not well defined.

8. Added the function DMAN_SelectImageDestinations. This function takes a source AE title as input and returns a list of possible destination applications based on information in the VideoImageDest table.

Version 2.9.7

Internal release.

1. Previous releases introduced a Limits table in the IDB facility to allow us to track the size of the image database. We have changed the units of the size variables in that table so that all values are in kilobytes. Previous releases mixed KB and bytes.
2. Update *manage* for Windows compatibility. Still not complete.
3. Add error comment fields to C-Get and C-Move response structures.
4. Updated some more facilities/applications so they compile under Windows.

Version 2.9.8

This release contains more work on multi-threaded code and Windows applications.

1. We are working on making the TBL and IDB routines thread safe (and are probably being overly cautious). We are also making some of the applications multi-threaded, like the *archive_server*. Making the servers multi-threaded means that we need to have some tables open in more than one thread. Multiple TBL calls to open a table used to result in an error. Now, we allow multiple callers (multiple threads) to open a table and keep a reference count for each open.
2. Update thread library to include Win32 threads.
3. Update the header extraction function in the *image_archive*. We received some images with single quote characters which confused the database functions. We updated the parse functions to strip those characters before we call the IDB insert routines.
4. Add code and switches to allow *archive_server* to run multi-threaded rather than forking a new process for each association. This version works in multi-threaded mode, but may have some problems with the original fork model. That means you will want to build the libraries for this version with the CTN_USE_THREADS macro defined and use the -t switch on *archive_server*.
5. We have a bug report of incorrect field lengths inside of TBL_Insert. We have made some changes to fix the problem, but the fix is incomplete.

Version 2.10.0

This release contains enough new features to warrant an increment in the minor version number. We have continued the work on porting the CTN tools to Windows (using MS SQL Server as the DB) and on making the code thread-safe.

Some of the thread changes are Solaris specific. We will incorporate fixes for other operating systems as we get reports and indications how to do so. You can turn off the thread support as described below.

1. All documentation is now available in pdf format as well as postscript. The easiest way to view the documentation is to point your web browser to the file `.../doc/doc_index.html`. This file contains links to all of the pdf files in the same directory.
2. We have added the THR facility to support threads. The current functionality is to support a common way to support mutex operations. If you want to use threads/mutexes, define the symbol `CTN_USE_THREADS` in `$DICOM_MAKE`. When you want to use threads, all applications should call *THR_Init* before they call any CTN libraries and *THR_Shutdown* just prior to exiting. We have added these calls to the applications we distribute. If you compile with threads and use applications that do not require threads, you still need to call these functions; the lower level CTN functions need to use the threads library and cannot tell if your application is multi-threaded.
3. Fix a problem in the COND facility. We were obtaining a mutex a second time when the stack overflowed, causing the application to block.
4. Define the MACROS `CTN_MALLOC` and `CTN_FREE` to help us track down some allocation problems. These are used in our facilities. There is no requirement that you use them.
5. Changed the rename code in the DMAN library for files in Microsoft environment. We used to map “.” to “z”. Now we map that to “_”.
6. Fixed an error in the SRV facility. When we created files in the Windows environment (as a storage SCP), we left a flag out of the open statement. Files were writable by our app, but were listed as readonly by the file system.
7. Modified the functions *THR_ObtainMutex* and *THR_ReleaseMutex* to take a facility number. This gives more flexibility and prevents one facility from locking out all other facilities.
8. This is a report of a problem we are having with `archive_server` in the Windows NT 4.0 environment. When we receive an image from an external source, we determine where to store the image and use the C run time library to determine if the directory exists. We are finding that the call to “stat” or “_stat” under Windows fails after we accept a number of associations and try to store a number of images (~40 associations, ~5000 images). We are running NT 4.0 with SP3. Maybe someone else has seen the same problem and can provide some advice.