Merge OEM
6509 Airport Road
Mississauga, ON, Canada L4V 1S7
Tel. +1-905-672-2100
Fax +1-905-672-2307

MergeCOM-3™ DICOM C/C++ Toolkit Release Notes

Please read the platform notes included for your platform first. It describes the system configuration on which this tool kit will run. Verify that your system configuration is appropriate. It also contains a road map to the rest of the MergeCOM-3 DICOM Toolkit Documentation.

PLEASE NOTE:

If you cannot find the answer to your questions in any of the documentation, contact Merge OEM via e-mail at:

mc3support@merge.com

Please include any relevant logs, API usage descriptions, or other data that may be helpful in diagnosing the problem in your e-mail along with the platform and release that you are using.

Release	Date	Description
4.1.0	3-Nov-2009	***Enhancements in version 4.1.0*** Added support for the following DICOM Supplements to the data dictionary: 43, Storage of 3D Ultrasound Images 119, Instance and Frame Level Retrieve SOP Classes 126, Colon Computer-Aided Detection SR SOP Class Made changes to the data dictionary to address the following change proposals: 750, 844, 849, 870, 875, 889, 895, 922, 923, 924, 925, 928, 932, 933, 935, 936, 937, 938, 939, 940, 943, 945, 948, 949, 950 Added support for all valid reject reasons in MC_Reject_Association(). Added support to limit the size of the printed values for the output of MC_List_Message() and T2 logging. This feature is configurable through the following configuration parameters in mergecom.pro: LIST_VALUE_LIMIT, LIST_SQ_DEPTH_LIMIT and LIST_UN_ATTRIBUTES. Added support for specifying paths relative to the initialization file for the location of the configuration files in merge.ini. Changed the code to tolerate reading DICOMDIR files that have the Referenced Lower-Level Directory Entity (0004,1420) attribute missing or empty in IMAGE records. Added sample applications for C-GET SCU and SCP services. ***Updates in version 4.1.0***

		Fixed a crash in MC_Get_Next_Encapsulated_Value_To_Function() call when no MC_Get_Encapsulated_Value_To_Function() was called first. Fixed the incorrect printing of extended negotiation information in T3 log messages. Fixed an issue that could cause MC_Cleanup_Memory to free up less memory than it could. Fixed a problem in MC_Set_Next_Value_From_String() that resulted in the wrong value count when the value set was an empty string (""). Fixed some errors and inconsistencies in the Reference Manual and the User's Manual. Changed the decoder to correctly deal with unknown length sequences attributes that have UN value representation. Previously the toolkit failed to decode such attributes if they were defined in the data dictionary. Fixed MC_Set_Value_From_Buffer() to reject uneven length buffers for VR's that do not have a padding defined. Fixed an issue on Linux platforms that caused memory corruption when accepting or requesting associations while the process had more than 1024 file descriptors open. Fixed a crash that occurred when AUTO_ECHO_SUPPORT was enabled and two echo requests were received in parallel. Removed a limitation that caused the toolkit to reject setting more than 2^16 values for an attribute. Fixed a problem that resulted in incorrect encoding after changing the VR of an attribute from OW to US.
4.0.0	1-May-2009	***Enhancements in version 4.0.0*** Added support for the following DICOM Supplements to the data dictionary: 123, Structured Display 125, Breast Tomosynthesis Image Storage SOP Class 128, Cardiac Stress Testing Structured Reports 132, Surface Segmentation Storage SOP Class 133, Color Palette Storage, Query and Retrieval 137, MPEG2 MP@HL Transfer Syntax

		141, Enhanced MR Color Image Storage SOP Class Made changes to the data dictionary to address the following change proposals: 363, 568, 817, 832, 837, 850, 851, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 871, 872, 873, 874, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 892, 893, 894, 897, 898, 899, 900, 901, 902, 903, 907, 908, 910, 911, 912, 914, 915, 916, 917, 918, 919, 920, 921, 926 Turning on the T1_MESSAGE configuration option will now cause detailed memory usage reporting to be logged when MC_Report_Memory() is called. ***Updates in version 4.0.0*** Fixed an issue in which tabs in the mergecom.app file could lead to MC_Open_Association() returning MC_CONFIG_INFO_MISSING. Fixed a crash in MC_Delete_Current_Value() if MC_Get_Next_Value_To_Int() had previously been called repeatedly to, for example, traverse the items in a sequence, and finally returned MC_NO_MORE_VALUES. Fixed an issue where MC_RLE_Compressor/Decompressor did not support the photometric interpretation YBR_FULL. Fixed an issue in MC_RLE_Decompressor() in which some multi-byte pixel images (16-bit or RGB) failed to decompress. Fixed an issue in which the Group 0 Length Element was generated only when EXPORT_GROUP_LENGTHS_TO_NETWORK was turned on. This Element is now always generated. Fixed an issue on Linux introduced in 3.9.0 in which MC_Read_Message() would always wait for one second irrespective of the timeout parameter passed in. Fixed an issue in which MC_Open_File() would crash for some DICOMDIR files when DICOMDIR_STREAM_STORAGE was turned on. Fixed an issue introduced in 3.8.0 IB2 (secure buffer manipulation) in which MC_List_File() would produce repeated blocks of hex output for the Part 10 File Prefix.
3.9.2	3-Feb-2009	Fixed an issue in which calling MC_Abort_Association() from one thread could crash when another thread attempted to continue processing a message on the same association.

		Fixed a crash that occured when validating or creating a message or file object for a service and command that had private attributes defined.
3.9.1	14-Jan-2009	Updates when porting to 64-bit Solaris 10 on Intel.
3.9.0	31-Oct-2008	***Enhancements in version 3.9.0*** Added support for the following DICOM Supplements to the data dictionary: • 117, Enhanced PET Image Storage SOP Class • 122, Specimen Identification and Revised Pathology • 130, Ophthalmic Refractive Structured Reports Made changes to the data dictionary to address the following change proposals: 128, 725, 767, 769, 781, 800, 801, 803, 805, 806, 807, 808, 809, 810, 815, 816, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 834, 839, 840, 841, 842, 843, 845, 846, 847, 848 Added more 1C and 2C type condition checks to validation. Improved validation performance and significantly reduced the overhead associated with validation. Integrated the latest Pegasus libraries. Now the toolkit distributes with Pegasus version 2.00.559. ***Updates in version 3.9.0*** Fixed an issue introduced in 3.8.0 IB2 (secure buffer manipulation) in which the log file could not be accessed by any viewer while the application was running. Fixed MC_Set_Value_Representation so that large LUTs (>64kb) encoded as VR US in Implicit VR Transfer Syntax can be reset to OW in order to be valid for re-encoding in Explicit VR. Fixed file writing to not pass the first 132 bytes separately, in order to prevent files from being written in two fragments on Windows regardless of the value of the WORK_BUFFER_SIZE option. Fixed an issue in which UT (unlimited text) attributes larger than OBOW_BUFFER_SIZE would be split into multiple values when LARGE_DATA_STORE was FILE. Fixed an issue in which odd size frames were being padded by the built in Standard and RLE decompressors. The correct approach, according to DICOM, is to pad the entire Pixel Data attribute if the pixels are 8-bit and the number of rows, columns, and frames are all odd. Fixed an issue in which the RLE decompressor would crash if the image was multiframe with no offset table and, according to DICOM, one of the compressed frames was padded to achieve even

		length. Fixed the listing of Pixel Data attribute VR to list the actual VR instead of OW. This fix affects all MC_List... calls and mc3list utility. Changed association negotiation to abort the association if an unknown transfer syntax is detected by the requester in an association response. Fixed an issue that caused MC_Abort_Association to have no effect if another thread was waiting for a message on the association. Fixed an issue that caused MC_FreeService to return MC_SERVICE_IN_USE even if the service was not in use. This happened if another service with the same SOP class and transfer syntax list was used by a service list object. Fixed the documentation for MC_Register_Application that incorrectly listed MC_NO_LICENSE and MC_INVALID_LICENSE as possible return values. Fixed info.pfl for missing attributes and other minor typos. Fixed division by zero issue when the Number Of Frames attribute had a value of zero. Now the toolkit tolerates such values and assumes that the pixel data has one frame. Fixed a problem with Patient level moves in the Query/Retrieve provider sample application. Also, filtering added to remove duplicates from Patient level query results.
3.8.0 IB6	30-Apr-2008	Fixed an issue in MC_List_Message when printing hex views of unprintable characters in attribute values, causing mc3list to terminate - from an exception on Windows and as assert on Solaris. Fixed a synchronization issue that could cause MC_Open_Association to fail if multiple threads were creating or updating the remote application object information simultaneously. Fix to allow MC_Duplicate_Message to duplicate a message with attribute that contains empty value. Fixed memory leak when decompressing a 10 bit grayscale JPEG_EXTENDED_2_4 image.
3.8.0 IB5	2-Apr-2008	Fixed MC_Set_Value_Representation to allow an unknown VR attribute with empty value to change to other VR. Fixed a synchronization issue that could have caused memory corruption when, in the same application, an SCU and an SCP used the same service list and requested/received

		multiple associations at the same time.
3.8.0 IB4	27-Mar-2008	Fixed a problem introduced in 3.8.0 IB2 that caused a C-language character string terminator (NULL) to appear in AE Titles during association negotiation, causing problems when communicating with some implementations, including ADVT. Fixed a crash in MC_FreeSyntaxList that occurs when the application creates and frees a custom transfer syntax list without creating a service list. Fixed small memory leaks associated with custom transfer syntax list objects and application objects. Fixed a problem that prevented the toolkit to receive N-EVENT_REPORT messages for the Unified Procedure Step service.
3.8.0 IB3	26-Feb-2008	Fixed defect where JPEG 2000 compressed image without offset table and less than 2K per frame was not decompressed properly. Fixed a problem introduced in the previous release that causes mc3icomb utility to hang.
3.8.0 IB2	31-Jan-2008	***Enhancements in Version 3.8.0*** Added support for the following DICOM Supplements to the data dictionary: • 74, Utilization of Worklist in Radiotherapy Treatment Delivery • 96, Unified Worklist and Procedure Step • 107, Verification of Substance Administration and Substance Information Query • 110, Ophthalmic Coherence Tomography (OCT) Storage SOP Class • 114, DICOM Encapsulation of CDA • 116, 3D X-Ray Made changes to the data dictionary to address the following change proposals: 617, 665, 668, 670, 681, 683, 684, 685, 688, 692, 695, 698, 699, 703, 710, 715, 716, 720, 721, 727, 734, 744, 747, 748, 749, 754, 755, 763, 764, 766 Integrated the latest Pegasus libraries and added support for dispatcher registration code setting. This fixes a crash that happens when using Pegasus libraries on Windows Server 2003 with DEP (Data Execution Prevention) turned on. Changed MC_Standard_Compressor and MC_Standard_Decompressor to allow lossless Jpeg compression/decompression of images with the number of bits stored between 2 and 16. Previously

		only images with bits stored of 8, 10, 12 and 16 were accepted. Added the current thread ID to the name of temporary files created on Windows. On Windows platforms, changed calls to C Runtime buffer manipulation routines to their equivalent secure versions. Note that now the static version of the toolkit is dependent on versions of the Microsoft C Runtime that implement these secure routines (MS Visual Studio 2005 or later). Added a new function MC_Thread_Release() that can be called by the application to release resources allocated for a specific thread. This call is needed to avoid a memory leak on Linux and Unix platforms with Pegasus compressor when using with multiple threads. Added synchronization to MC_Cleanup_Memory() to avoid memory corruption when applications call MC_Cleanup_Memory() from different threads. Sample application prnt_scu (prnt_scu.c) now accepts media type files. ***Defects Resolved in Version 3.8.0*** Fixed problem in which MC_Get_Next_Value_To_String() returned MC_NO_MORE_VALUES when retrieving key attributes in DICOMDIR records after sorting the DICOMDIR with MC_Dir_Sort(). Fixed crash when private services were defined having more than 35 characters. Fixed problem in the breakup of bulk attributes for storage in memory that led to the error message "Odd length buffer passed to FlushStreamBuffer". Fixed a problem in validation that resulted in MC_MSGFILE_ERROR status being returned when element 0 in a private block was defined in the dictionary. Fixed mc3icomb utility for missing items in the output when more than one item was specified in the [EXISTING_ITEM_SECTION] of the configuration file. Fixed linear/circular buffer interfacing with Pegasus library to resolve Pegasus related decompression problems. The toolkit previously had some inconsistent usage of the linear/circular buffer interfacing with the Pegasus library, which resulted in unsuccessful decompression of JPEG images. Some cleanups were also made to simplify buffer management when interfacing with the Pegasus library. JPEG encapsulated images with extra padding at the end of each fragment would previously cause the parser to fail decoding and return an error. Fixed problem where an association abort from the remote peer was not accurately reported by MC_Read_Message(); the status returned was MC_NETWORK_SHUT_DOWN. MC_Read_Message() now returns MC_ASSOCIATION_ABORTED in this situation. Fixed several problems in the Query/Retrieve SCP and SCU sample applications that resulted in

		zero matches for image level queries and a memory corruption when executing queries with date or date range matching.
3.7.0 IB14	23-NOV-07	***Updates in Version 3.7.0 IB14*** Changed the way synchronization is handled to avoid frequent creation and deletion of synchronization objects and to avoid potential memory issues after calling MC_Library_Release().
3.7.0 IB13	4-OCT-07	***Updates in Version 3.7.0 IB13*** Fixed defect where the log file was not backed up properly when it became full, even though LOG_FILE_BACKUP was configured ON. The actual behavior of this backup varied from platform to platform. On Windows it did not work at all. Log file backup worked during library initialization only.
3.7.0 IB12	19-APR-07	***Updates in Version 3.7.0 IB12*** Fixed defect related to private attributes that were not assigned to the first private block in a private group. The MC_Get_pValue_... related functions were returning that the tags did not exist in the message. Fixed defect where when the T2 level logging was enabled, the MC_Get_Next_... routines would not properly return the first value of a tag within a message received over the network. Resolved problem with the mergecom.pro file where the JPEG 2000 Multi-Component transfer syntax UIDs were entered improperly. Updated the PDF documentation to have bookmarks. On Windows, removed the dependency introduced within 3.7.0 to the msvcr71.dll DLL file. The mc3adv.dll file is now compiled with the /MT option so that the contents of the C runtime is compiled into the DLL file. The DLL file has increased in size because of this change.
3.7.0 IB11	22-MAR-07	***Updates in Version 3.7.0 IB11*** Fixed defect with MC_Read_Message_To_Tag that would cause subsequent problems with MC_Continue_Read_Message or MC_Continue_Read_Message_To_Stream. When receiving Implicit Little Endian data, and a PDU was received where the last data in the PDU was the tag length of the stop tag passed to MC_Read_Message_To_Tag, the function would not save the tag and length for subsequent calls, and would cause a parsing failure.
3.7.0 IB10	9-FEB-07	***Updates in Version 3.7.0 IB10*** Final updates for the MacOS X on Intel toolkit.
3.7.0 IB9	8-FEB-07	***Updates in Version 3.7.0 IB9*** Further updates for IRIX6.5. Resolved defect with MC_Send_Response_Message. When MergeCOM-3 was unable to

		automatically set the Message ID Being Responsed To tag, a semaphore was not being released properly and other subsequent calls to network functions would freeze and never return. This would only occur on platforms that support threading. Fixed memory leak with MC_Wait_For_Connection and MC_Wait_For_Connection_On_Port when they were called with a timeout other than -1 and the function returned a value other than MC_NORMAL_COMPLETION. Fixed memory leak with the T2 logging level. Fixed defect with MC_Duplicate_Message where it didn't work properly when making a copy of a compressed image. In this case no decompressors were registered, and the destination transfer syntax was the same as the source transfer syntax. The encapsuation of image was being corrupted.
3.7.0 IB8	19-JAN-07	***Updates in Version 3.7.0 IB8*** Updates when porting to IRIX 6.5 and HP-UX 11
3.7.0 IB7	15-JAN-07	***Updates in Version 3.7.0 IB7*** Updates when porting to Solaris 8 and updates for the Java Wrapper.
3.7.0 IB6	3-JAN-07	***Updates in Version 3.7.0 IB6*** Updates when porting to Borland C++ Builder 2006 and 64-bit Windows.
3.7.0 IB5	15-DEC-06	***Known Issue in Version 3.7.0*** On Linux and Solaris systems, when using multiple threads with the standard compressor or decompressor, there is a memory leak. The problem only occurs on these operating systems when using multiple threads. Specifically, there is a Pegasus routine that should be called to release resources associated with a thread before that thread terminates. The following is a prototype of the Pegasus function: `extern void PegasusLibThreadTerm(void);` This Pegasus function should be called directly from any Linux or Solaris application where the standard compressor is called from a thread which later exits. The call should be made from the thread before it exits. In a future release, a new MergeCOM-3 function will be introduced so applications do not have to call this Pegasus function directly. ***Enhancements in Version 3.7.0*** Added support for the following DICOM Supplements to the data dictionary: • 111, Segmentation Storage SOP Class • 112, Deformable Spatial Registration Storage SOP Class

		Made changes to the data dictionary to address the following change proposals:
		
		457, 526, 550, 553, 554, 572, 574, 576, 578, 579, 582, 586, 587, 589, 593, 599, 600, 610, 611, 614, 615, 616, 618, 627, 628, 642, 643, 649, 651, 652, 655
		
		Added support for IPv6 connections on operating systems that support IPv6. The `IP_TYPE` configuration option has been added to control how IPv6 is supported within MergeCOM-3. When this option is set to `IPV4`, MergeCOM-3 will attempt to only listen for IPv4 connections and will only use IPv4 addresses when connecting to a remote system. When `IPV6` is configured for this option, MergeCOM-3 will only attempt to use IPv6 for network communications. When `AVAILABLE` is configured, MergeCOM-3 will attempt to listen or connect with the IPv4 or IPv6, depending on what is available.
		
		Note that IPv6 IP addresses can be passed as a parameter to MC_Open_Association or configured in the mergecom.pro file. When a hostname is configured, MergeCOM-3 will attempt to resolve this hostname to an IP address according to the setting of the `IP_TYPE` configuration option.
		
		As part of support for IPv6, changes were made in the MC_Register_Network_Capture_Callbacks function. Specifically, all callback functions that included network addresses as their parameters have been changed to use `struct sockaddr` pointers as the addresses. Previously we used an unsigned long to represent an IP address and an unsigned short to represent a port.
		
		Made some small performance improvements in various parts of the library after profiling the library's network performance.
		
		Added new FREE_DATA callback for Callbacks registered with MC_Register_Callback_Function. When the new USE_FREE_DATA_CALLBACK configuration option is set to Yes, callback functions will be called with the FREE_DATA parameter when the message that is associated with the callback is freed by the library.
		
		The Query/Retrieve SCP (qr_scp.c) sample application has been updated. This server is now a functional Query/Retrieve SCP server (as opposed to the previous version which used fake data). The application can receive C-STORE-RQ messages, store them locally, and then respond to C-FIND-RQ and C-MOVE-RQ messages properly based on the images its received.
		
		Added support for DICOM Supplement 99, User Identity negotiation. Several new functions have been added to the library related to User Identity Negotiation:
		
		`MC_Open_Association_With_Identity`
		`MC_Accept_Association_With_Identity`
		`MC_Get_User_Identity_Info`
		`MC_Get_User_Identity_Length`
		
		In addition, the AssocParms structure returned by MC_Get_Association_Info has been extended to include user identity related information. See the Reference Manual for a complete description of these functions. Finally, the Storage SCU and SCP sample applications have been modified to show how to implement these functions.

		Updated the default maximum PDU size and the default TCP/IP send and receive buffer sizes in the mergecom.pro file. The maximum PDU size and TCP/IP Send and receive buffer sizes are now set to a multiple of the TCP/IP MSS (maximum segment size) of 1460 bytes in an effort to increase performance. See the frequently asked questions of the User's Manual or the discussion of the various MergeCOM-3 configuration options in the Reference Manual for further details. Added new configuration option, `UPDATE_GROUP_0028_ON_DUPLICATE`. When set to `Yes`, MergeCOM-3 will automatically update the group 0x0028 elements with a duplicated message or file. The Photometric Interpretation, Lossy Image Compression, Lossy Image Compression Ratio, and Lossy Image Compression Method tags will be updated if appropriate. Note that on lossy compression, a new SOP Instance UID may still need to be updated by the application. Added the duplicate.c sample application as part of the distribution on all platforms. This can now be used with RLE compression on all platforms. ***Defects Resolved in Version 3.7.0*** Resolved defect introduced in the 3.6.0 release where in some cases the MC_Send_Request_Message, MC_Send_Response_Message, and MC_Read_Message routines were incorrectly returning `MC_SYSTEM_ERROR` on threaded platforms when an association was aborted. These routines will now return `MC_ASSOCIATION_ABORTED`. Addressed a problem where MergeCOM-3 could not read DICOM Part 10 format files that had the incorrect value for Group 0x0002 length encoded in them. We can parse this type of file. Removed a misleading log line from the log file when MC_NewSyntaxList is called. Previous versions would display an error message that the syntax list being created by this routine could not be found. Fixed two small memory leaks with MC_Duplicate_Message that totaled around 100 bytes. The leak would occur when duplicating a message that had tags encoded after the pixel data tag. Fixed problem with MC_RLE_Compressor where it did not work with RGB images that were an odd image size (and contained a padding byte). These were 8 bit images that have an odd value for both rows and columns. Fixed small memory leak with the MC_Continue_Read_Message_To_Stream function. Fixed problem with the `COMPRESSION_RGB_TRANSFORM_FORMAT` configuration option. The option was not being read properly from the mergecom.pro file. Resolved issue with the Storage SCU sample application where it could not transfer more than 32000 images at one time due to how it tracked when a response message was received. Fixed data dictionary issue where the `DIR_REC_IMAGE` directory record did not allow setting of the Frame of Reference UID. Also added several other tags that could not be set that should have

		been allowed. Added fix where MC_List_Message/MC_List_File was not displaying FL/FD attributes properly. These values were displayed as longs, and thus did not display the precision of the numbers. Fixed problem with the Borland compiled Windows tool kit where standard compressors and decompressor was not exported from the DLL properly. Fixed problem with Borland compiled Windows tool kit where MC_List_Message/MC_List_File would not display output properly when using the DLL. Fixed problem with MC_Standard_Compressor where if we encountered an image where the pixel data attribute was too small, we would fail silently instead of logging an error. We now log an error when encountering this situation. Fixed problem where the attribute immediately preceding a private attribute when a message that was read into MergeCOM-3 could not be retrieved with a call to one of the MC_Get_Next_Value routines. One of the MC_Get_Value routines would have to be called first. Fixed problem where the MC_Validate_Message routine would not work properly when pixel data was contained within a registered callback function. Changed MC_Register_Compression_Callbacks to return `MC_INVALID_MESSAGE_ID` instead of `MC_SYSTEM_ERROR` when it is called with an invalid message ID.
3.6.1 IB1	5-JUL-06	***Known Issue in Version 3.6.1*** On Linux and Solaris systems, when using multiple threads with the standard compressor or decompressor, there is a memory leak. The problem only occurs on these operating systems when using multiple threads. Specifically, there is a Pegasus routine that should be called to release resources associated with a thread before that thread terminates. The following is a prototype of the Pegasus function: `extern void PegasusLibThreadTerm(void);` This Pegasus function should be called directly from any Linux or Solaris application where the standard compressor is called from a thread which later exits. The call should be made from the thread before it exits. In a future release, a new MergeCOM-3 function will be introduced so applications do not have to call this Pegasus function directly. ***Enhancements in Version 3.6.1*** Updates to introduce the Pegasus libraries for Lossy and Lossless JPEG (JPEG 2000 is not available at this time) on Fedora Core 3 for x86-64. Ported the library MacOS X on Intel. Universal binaries are now supplied for 32-bit PowerPC and Intel based MacOS X systems. Added support back in for dynamic libraries on MacOS X on Intel.

		Defects Resolved in Version 3.6.1 Resolved defect introduced with the changes in listing of messages/files with the 3.6.0 release. A crash could occur in the library when T2 level logging was enabled or when the MC_List_Message or MC_List_File routines were called. The crash could occur when the library was listing a multi-valued text attribute that had a null component. Resolved defect introduced in the 3.6.0 release where in some cases the MC_Send_Request_Message, MC_Send_Response_Message, and MC_Read_Message routines were incorrectly returning MC_SYSTEM_ERROR on threaded platforms when an association was aborted. These routines will now return MC_ASSOCIATION_ABORTED. Fixed defect in the data dictionary where the STANDARD_OPHTHALMIC_16_BIT and STANDARD_OPHTHALMIC_8_BIT images incorrectly did not have the Sop Common module included with them. Instead, one of the Enhanced MR modules was included with these image types. Resolved problem on 32-bit Linux operating systems where a crash was occurring with the Pegasus libraries when doing compression or decompression on an Intel EM64T based processor systems. Made some minor changes to the sample applications in how they report timings. Also made some minor changes with some callback functions to make them work better when we compile the library on Windows with __stdcall calling convention (for the new .NET wrapper) instead of _cdecl.
3.6.0 IB7	3-MAY-06	***Updates in Version 3.6.0 IB7*** Updates to the 3.6.0 code when porting the tool kit to Solaris 10 on Intel x86. Some minor updates were done to header files in anticipation of a .NET wrapper version of MergeCOM-3.
3.6.0 IB6	6-APR-06	***Updates in Version 3.6.0 IB6*** Updates to the 3.6.0 code when porting the tool kit to SunOS, IRIX 6.5, MacOS X, and AIX. Note, support for MacOS X dynamic libraries, which was reported as being complete in previous build of 3.6.0 was removed. Problems were encountered while testing. This support will be added in a future release. Resolved defect introduced in the 3.5.1 release where fixed width binary VR tags could not be retrieved with the MC_Get_Value_To_Function call. Tags with a VR of SL, SS, UL, US, AT, FL, and FD were affected.
3.6.0 IB5	3-APR-06	***Updates in Version 3.6.0 IB5*** Updates to the 3.6.0 code when porting the tool kit to LynxOS, 64-bit Linux (Fedora Core 3),

		Windows with Borland C++, Win64, HP10, and HP11
3.6.0 IB4	3-MAR-06	***Enhancements in Version 3.6.0*** Added support for the following DICOM Supplements to the data dictionary: • 83, Enhanced XA Image Storage SOP Class/ Enhanced XRF Image Storage SOP Class • 86, Digital Signatures in Structured Reports • 94, Diagnostic X-Ray Radiation Dose Reporting (Dose SR) • 100, Color Softcopy Presentation State Storage SOP Class • 101, HL7 Structured Document Object References • 102, Radiotherapy Extensions for Ion Therapy • 103, Real World Value Mapping SOP Class • 104, DICOM Encapsulation of PDF Documents • 105, JPEG 200 Part 2 Multi-component Transfer Syntaxes Made changes to the data dictionary to address the following change proposals: 484, 490, 497, 498, 501, 506, 507, 515, 530, 536, 537, 549, 551, 558, 561, 563, 564, 566, 569, 575 Added support for DICOM Supplement 90. As part of this support, added new configuration options, ACCEPT_STORAGE_SERVICE_CONTEXTS and ACCEPT_RELATED_GENERAL_SERVICES to the mergecom.pro file. Also made modifications to the mergecom.app file to support specifying related general SOP Classes in the proposed service list. Added changes to support the new JPEG2000 multi-component transfer syntax in supplement 105. Note that this transfer syntax is only supported, the standard compressor does not support compressing or decompressing these syntaxes. Added new routines, MC_RLE_Compressor and MC_RLE_Decompressor, which can be used with MC_Duplicate_Message to RLE compress or decompress an image. These routines are available on all platforms. The duplicate.c sample has also been modified to show how to support these. Added changes to improve the tool kit's memory usage when working with large DICOMDIR files. First, a new configuration option, DICOMDIR_STREAM_STORAGE has been added. When set

		to Yes, this option causes the tool kit to store individual directory records in "stream" format, to reduce their memory footprint. If tags in the directory record are accessed, the directory record is automatically translated into our internal data structures for updating. The MC_Dir_Stream_Directory_Record routine has been added to translate a directory record that is stored in our internal format to be stored in the "stream" format. This can be used when creating a DICOMDIR after a directory record has been created and it is unlikely that it will be referenced again. Added new MC_Reset_Message_Transfer_Syntax routine. This routine can be used to reset the transfer syntax associated with a message after it was received over the network. This routine is required if a message is received over the network in an uncompressed format, and then wants to be sent over the network on a different uncompressed transfer syntax. Previously this did not work properly and the message had to be sent with the same transfer syntax as it was received in. This routine allows the user to avoid storing the presentation contexts negotiated over an association. Added new routine, MC_Open_Empty_Item, which can be used to create an empty item. Changed the format of the log when T2 logging is enabled. Previously, messages were logged in the format generated by the MC_List_Message routine. A new format has been developed that only contains a single log line per tag in the message in order to increase performance. Also, some modifications were made in the MC_List_Message format for multi-valued tags. These tags are now listed with a \ character in between each value, instead of having a new line for each value. Made changes in how we validate multi-byte character set values. We now will no longer flag Korean values as invalid. Modified mc3conv to support conversion to and from compressed transfer syntaxes. This support is now added on platforms where we support the Pegasus libraries. (32 bit Linux, Solaris, and 32 bit Windows). RLE conversion is supported on all platforms. Improved the toolkit logging when MC_Open_Association fails because a remote system is not operational to better help diagnosis the problem in the field. Made changes to MC_Report_Memory so that memory that was directly malloc'd by the library (usually large buffers) are now tracked and their usage is reported by this call. Previous this memory was not counted as being in use. Added new configuration option, CREATE_OFFSET_TABLE. This option specifies if MC_Duplicate_Message will create an offset table when duplicating to a compressed transfer syntax. The default value is Yes. Previous versions of MergeCOM-3 always created an offset table. Added new configuration option, PEGASUS_OPCODE_PATH that specifies the directory in which the Pegasus opcode DLLs should be loaded. The opcode DLL refers to files like picn6220, etc, and not the dispatcher DLL picn20. If this option is empty, the SSM/DLL is loaded from the

		same directory as the dispatcher DLL. If not found there then the opcode SSM is loaded using the directory order Windows uses when loading SSMs/DLLs. The SSM/DLL is loaded from the current directory if '.' Is specified. Made changes on MacOS X so the library is now supplied as a dynamic library and a static library. ***Defects Resolved in Version 3.6.0*** When decompressing a message with MC_Duplicate_Message, the decompression callback function would not be passed the isLast flag when an offset table was not included. MergeCOM-3 will now set the isLast flag for single frame images without an offset, however, it still will not set this flag for multi-frame images. The user is expected to parse the incoming data to determine when a frame ends. Fixed problem with compression and decompression of images by the standard compressor that were the source format was encoded as YBR_FULL_422. Fixed defect where MC_Standard_Compressor would fail for JPEG Lossless compression (and potentially other compression methods) if we internally stored the last buffer of pixel data in a small buffer. Fixed several memory leaks in MC_Standard_Compressor and MC_Standard_Decompressor that occurred in failure conditions with either routine. Corrected data dictionary problem where tag 0040,0281 was not present in MPPS N-SET-RQ message. Corrected a data dictionary problem where the tag Referenced Patient Setup Number (300C,006A) was incorrectly marked as retired and removed from the diction.h file. Corrected data dictionary problem where the tags Study Read Date (0032,0034) and Study Read Time (0032,0035) were not included study level C-FIND request messages. Added additional defined terms for tag (0008,0064) which were not added properly. Fixed problem with tag (0040,A050) not being include in the SR content sequence of the Basic Text and Comprehensive SR IODs. In version 3.5.1, MC_Report_Memory incorrectly had its parameters changed to type int. These parameters have been changed back to type unsigned long through use of a new typedef. Resolved defect where MC_Message_To_SR was extremely slow when converting large structured reports. This routine should now be much faster. Fixed problem where calling MC_Abort_Association for an association in one thread while a second thread is sending a message over the network or reading a message from the network could cause a crash in some instances. The ability to call MC_Abort_Association from another thread

		was added in version 2.4.1. Fixed problem introduced in 3.5.1 where the tool kit used an excessive amount of memory when storing tags that were close to 64 bytes in length. This may have caused memory usage to increase when loading large DICOMDIRs. Memory usage should now be more in line with the actual size of the tags being stored. Fixed problem where retrieving a private tag via the MC_Get_Next_pValue or MC_Get_pValue routines could cause the last accessed pointer to be reset for other tags in the message.
3.5.1 IB8	17-NOV-05	***Updates in Version 3.5.1 IB8*** Fixed problem where MC_Duplicate_Message would not work properly on DICOM files unless they were originally DICOM messages and they had their transfer syntax set with MC_Set_Message_Transfer_Syntax.
3.5.1 IB7	16-NOV-05	***Updates in Version 3.5.1 IB7*** Fixed longstanding problem with our routines for reading in files and streaming in messages that could cause tags in a file or message to be ignored. Specifically, when a process would run out of memory or was close to running out of memory, there was a condition where our routines for streaming in messages and files would ignore a tag in a stream, but not return an error. In this case, if the file or stream was written back out again, it would be missing tags without the user realizing it. Note, however, that there were error messages logged in this situation that could be used to identify the condition occurred. The tool kit will now fail the read process when this error occurs. Fixed crashing problem with MC_Validate_Attribute and MC_Validate_Message. These routines would crashing when validating a standard DICOM tag contained in a message that was not a valid tag for the message. Fixed a problem where we could not decompress properly images whose decompressed size are an odd number of bytes in size. Specifically, these images would be decompressed to be 2 bytes too small, and there was a memory leak in that decompression resources were not released when these images were decompressed. Resolved defect introduced in 3.5.1 where private tags defined in the data dictionary could not be looked up to determine their tag name and VR. Resolved longstanding defect with MC_Read_Message_To_Tag, MC_Continue_Read_Message_To_Tag, and MC_Continue_Read_Message(). In previous versions, when the stop tag passed to MC_Read_Message_To_Tag or MC_Continue_Read_Message_To_Tag was the last tag in the message (typically the pixel data tag), subsequent calls to MC_Continue_Read_Message would fail and return MC_INVALID_MESSAGE_RECEIVED and the association would be aborted. Now, they will return MC_NORMAL_COMPLETION.

		Fixed problem introduced in 3.5.1 where MC_Duplicate_Message would fail when attempting to duplicate a message that had a SQ VR attribute set to NULL. Fixed problem with MC_Duplicate_Message being used on a file object. If there were tags after the pixel data in the message and the destination format was an encapsulated transfer syntax, the destination file object created would be corrupted with the majority of the tags missing from the object. Modified the MC_Set_Message_Transfer_Syntax and MC_Get_Message_Transfer_Syntax routines to now work on DICOM file objects. These routines set and get the tag 0x00020010 as specified by each routine. Fixed problem with MC_Duplicate_Message where it couldn't duplicate messages when the Samples Per Pixel tag was missing in the message. We now check the value for Photometric Interpretation and assume a value for Samples Per Pixel based on this tag.
3.5.1 IB6	18-OCT-05	***Updates in Version 3.5.1 IB6*** Updates for porting to Solaris on SPARC.
3.5.1 IB5	7-OCT-05	***Enhancements in Version 3.5.1 IB-5*** Made several internal changes in the library to reduce the memory usage of the library by changing how data is stored. Part of the change was that memory to store validation information for tags is only allocated now when a message is created with a specific type. Also made changes in how some information is stored related to the compression and decompression of objects. Added additional description of Winsock errors into our log file. Made updates to support Solaris 10 on Intel x86. ***Defects Resolved in Version 3.5.1 IB-5*** Fixed a defect where we couldn't support pixel data that was greater than 2GB in size. Fixed memory leak with MC_Release_Callback_Function. A small leak existed each time this routine was called. Fixed longstanding problem with MC_Validate_Message & FORCE_OPEN_EMPTY_ITEM. We were not loading validation info for sequences properly and not validating sequences when the FORCE_OPEN_EMPTY_ITEM configuration option was set to Yes.
3.5.1 IB4	22-SEP-05	***Enhancements in Version 3.5.1 IB-4*** The MC_Wait_For_Association and MC_Wait_For_Secure_Association functions have a limitation in that they block incoming associations while they're processing an association request or an echo request. Specifically, if they are processing an association request, no other

associations can be received and processed at the same time. If a connection is opened, but no data is sent, the call will block until the ARTIM_TIMEOUT expires. This would cause a server to be unresponsive during this time. To address this problem, this release adds a number of new functions to allow a new thread or process to be created right after a connection is received, but before we attempt to read any data from the connection. These new functions are described in the following paragraphs.

The MC_Wait_For_Connection and MC_Wait_For_Connection_On_Port routines wait for an incoming connection. As soon as the connection is received, the functions exit and return the socket which has connected. The MC_Process_Association_Request and MC_Process_Secure_Association_Request methods can then be called from a new thread or forked process to handle the connection, thus preventing the main thread waiting for connections from blocking. On UNIX systems, the MC_Release_Parent_Connection function can be used to close the socket for the new connection in the parent process. The following are the prototypes for these new functions:

```
MC_STATUS MC_Wait_For_Connection(int            Timeout,
                                 MC_SOCKET*     Socket);

MC_STATUS MC_Wait_For_Connection_On_Port
                                (int            Timeout,
                                 int            Port,
                                 MC_SOCKET*     Socket);

MC_STATUS MC_Process_Association_Request
                                (MC_SOCKET      Asocket,
                                 const char*    AserviceListName,
                                 int*           ApplicationID,
                                 int*           AsessionID);

MC_STATUS MC_Process_Secure_Association_Request
                                (MC_SOCKET      Asocket,
                                 const char*    AserviceListName,
                                 int*           ApplicationID,
                                 int*           AsessionID,
                                 SecureSocketFunctions* SS_functions,
                                 void*          SS_application_context);

void MC_Release_Parent_Connection(MC_SOCKET     Asocket);
```

The parameters to these functions are similar to those utilized by the MC_Wait_For_Association and MC_Wait_For_Secure_Association functions. The Storage SCP sample application has also been modified to use these functions. Please see this sample application for an example implementation.

Made modifications in how we write data to the network to improve performance. These changes require some additional CPU time when sending data over the network, but it should improve the actual transmit times and eliminate some inconsistencies that were previously seen with network performance. Previously we would write PDU/PDV headers in small buffers, that would sometimes cause small TCP/IP packets to be generated and sent over the network. We now buffer

		these packets, and cause them to be written at once, which should improve the network performance. With this change, the COMBINE_DATA_WITH_HEADER configuration option no longer has any impact. Also, the DESIRED_LAST_PDU_SIZE configuration option is no longer utilized by the library. Made changes in how tags in our internal data dictionary are stored to improve performance. This change also caused the format of the mc3dict.c file generated by the gendict utility to change. Note also that all data dictionary access is through memory. The DICTIONARY_ACCESS configuration option is no longer utilized. These changes should help to slightly improve performance on systems that have large private data dictionaries. Created a new typedef, MC_SOCKET that is used in places where a socket is retrieved from or passed into the library. On Windows systems, this typedef is of type SOCKET, on all other platforms it is an int. Made changes to port the library to the Win64 platform. Added a new typedef, MC_size_t, that is now used by a number of toolkit API calls when a length is required, and we previously had used an integer to signify the length. On the Win64 platform, this typedef is the size_t type. On all other platforms this is set to an int. Note that in a future release, we likely we change this variable to be size_t on all platforms. Added usage of the setvbuf function into the sample applications where we were reading or writing files. This function should help with performance reading and writing files. Made changes to our internal data structures in how we maintain tags within a message. These changes should reduce the number of malloc's done by the library and slightly reduce the memory usage of the library. These changes will especially be seen when dealing with DICOM files or message with a large number of attributes in the header such as DICOMDIRs and some of the RT messages. Made changes in how pixel data is stored internally in the library. It is now stored in larger chunks that are managed by the library. Performance of the library when streaming in messages or DICOM files is improved with these changes and the behavior of the library will now be more consistent in how we internally store the pixel data. Made a large number of other general performance improvements and changes to the library. ***Defects Resolved in Version 3.5.1 IB-4*** Fixed a defect where dynamically created transfer syntaxes were not being freed properly. Added fix in how unsolicited C-CANCEL-RQ messages were dealt with. We no longer need the record of the original RQ message to process the Cancel message. Fixed problem in most of the sample applications where we were not flushing files that we wrote to

		disk, and we did not check the return value from the fclose() function call. This could have been causing files to not be written to disk properly, and us not detecting it.
3.5.1 IB3	13-JUL-05	***Enhancements in Version 3.5.1 IB-3*** Made further internal changes and optimizations. These changes should further reduce the number of mallocs the toolkit requires to store attributes in messages and thus improve the overall performance of the library. Added a change for writing DICOMDIRs that significantly increases performance when writing out large DICOMDIRs. ***Defects Resolved in Version 3.5.1 IB-3*** Resolved a memory leak in the IB2 version when setting multiple values in fixed width VR tags.

3.5.1 IB2	11-JUL-05	***Enhancements in Version 3.5.1***
		
		Changed the internal representation of values to reduce the number of mallocs the toolkit requires to store attributes in messages.
		
		This release of the MergeCOM-3 toolkit allows several of the message manipulation routines in the MergeCOM-3 library to be reentrant on platforms that we support threading. We currently support threading on Windows, Linux, Solaris, and HP-UX 11. The following functions are reentrant:
		
		`MC_Read_Message_To_Tag`
		`MC_Continue_Read_Message`
		`MC_Continue_Read_Message_To_Stream`
		`MC_Register_Callback_Function`
		`MC_Set_Value_...`
		`MC_Set_pValue_...`
		`MC_Get_Value_...`
		`MC_Get_pValue_...`
		
		Also, the MC_Get_First_Attribute and MC_Get_Next_Attribute routines can be called from a single thread at the same time as the above routines.
		
		This support for reentrancy allows multi-tasking when reading in images off the network in two scenarios:
		
		1. The MC_Read_Message_To_Tag routine could be used to read a message up to a tag before the pixel data. A background thread could be created to parse the header and insert data into a database while the primary thread calls MC_Continue_Read_Message to read the remainder of the message including the pixel data.
		
		2. A registered callback function could be in use via the MC_Register_Callback_Function routine. In this case, a background thread could be created when the registered callback is called to do background process of the header, or to add additional tags if needed into the header as the pixel data is being read from the network.

3.5.0 IB6	16-JUN-05	***Updates in Version 3.5.0 IB6*** This release resolves a number of issues related to Pegasus compression. This build only affects Windows, Solaris, and Linux versions of the library: • Resolved an issue introduced in 3.5.0 where JPEG2000 images could not be decompressed that didn't have an offset table in them. (Note that MergeCOM-3 adds an offset table to all compressed images.) The toolkit failed to flag to process the last chunk of data • Resolved an issue introduced in 3.5.0 where JPEG2000 images could not be decompressed if the entire image fit in one internal buffer within MergeCOM-3. The toolkit failed to flag to process the last chunk of data. This would originate with small images. • Resolved an issue where MergeCOM-3 could not compress properly pixel data with a photometric interpretation of YBR_FULL_422. A generic calculation of frame size based on sample per pixel was wrongly used for YBR_FULL_422 and caused data overflow. • Fixed problem where some images could not be compressed in JPEG Baseline and JPEG Extended 2/4 format. The latest Pegasus library introduced in 3.5.0 required data to be passed in larger chunks. The toolkit would fail compression on some images where the last chunk of data to compress was too small. • Toolkit will no longer attempt to compress in JPEG Baseline or JPEG Extended 2/4 format images that are 16 bits allocated, 8 bits stored. The Pegasus libraries will not properly decompress images back to 16 bits allocated, so this is disabled. Previous versions of the library were not compressing and Decompressing these image types properly. (Note that the other compression algorithms work fine with these image types.) • Fixed problem with compression of multi-frame images in JPEG Baseline and JPEG Extended 2/4 format introduced in 3.5.0. This problem occurred when the image data is stored internally to MergeCOM-3 in large data buffers. It did not occur when the data was stored internally in small chunks.
3.5.0 IB5	26-APR-05	***Updates in Version 3.5.0 IB5*** Further changes for porting to Borland C++ compiler. Resolved defect originally attempted to have been fixed in 3.4.0 with how the Windows toolkit released Winsock resources. There were problems with the how the library released Winsock and reinitialized it when the library was reset or released and reinitialized. On Windows, Solaris, and Linux, resolved defect with the Lossy JPEG decompressor where it would not return enough data in some cases when attempting to decompress images. Made changes to resolve problem on Linux with the library crashing in MC_Open_Association

		when a network cable was disconnected.
3.5.0 IB4	20-APR-05	***Updates in Version 3.5.0 IB4*** Updates for porting to MacOS X, HP-UX 10, LynxOS, AIX, OSF/1 (Digital UNIX) and IRIX.

3.5.0 IB3	30-MAR-05	***Enhancements in Version 3.5.0***
		Added support for the following DICOM Supplements to the data dictionary:
		• 60, Hanging Protocols
		• 75, Relevant Patient Information Query
		• 88, Media Creation Management
		• 91, Ophthalmic Photography SOP Classes
		Made changes to the data dictionary to address the following change proposals:
		• 155, 168, 194, 201, 252, 393, 433, 435, 436, 438, 440, 441, 443, 444, 447, 465, 467, 469, 476, 480, 482, 483, 486, 487, 489, 491
		Implemented support for CP-252. This allows the Unicode and GB18030 character sets to be supported with MergeCOM-3 without generating an MC_INVALID_CHARS_IN_VALUE warning return value when setting UTF-8 or GB18030 encoded strings. The toolkit checks the specific character set tag 0008,0005 if it is set to GB18030 for Chinese character sets, and ISO_IR 192 for Unicode. It will only allow GB18030 encoding when GB18030 has been set and it will only allow Unicode UTF-8 encoding when ISO_IR 192 is set.
		Added the following new performance related configuration options to the mergecom.pro file:
		`EXPORT_UNDEFINED_LENGTH_SQ`
		`EXPORT_UNDEFINED_LENGTH_SQ_IN_DICOMDIR`
		`EXPORT_GROUP_LENGTHS_TO_NETWORK`
		`EXPORT_GROUP_LENGTHS_TO_MEDIA`
		The EXPORT_UNDEFINED_LENGTH_SQ configuration options configures if undefined or defined length sequences are used when exporting to disk or transferring over the network. This option configures all exports except when writing DICOMDIRs to media. The EXPORT_UNDEFINED_LENGTH_SQ_IN_DICOMDIR configuration option sets if undefined length sequences are exported for DICOMDIRs. Setting either of these options to Yes increases the streaming performance of the library, especially when exporting messages that have a large number of sequences.
		The EXPORT_GROUP_LENGTHS_TO_NETWORK and EXPORT_GROUP_LENGTHS_TO_MEDIA configuration options specify if group length attributes are exported by the library to the network or media, respectively. Setting these options to No will cause group length attributes to not be in outgoing streams/files created by the library (if

		they are included in the source messages). This will also improve the export performance of the library. Made changes to improve the performance of the MC_Get_Next_Value_... routines for attributes with a large number of values. These changes specifically apply to text attributes or SQ attributes (and not to fixed width binary tags). Previously, the performance of the MC_Get_Next_Value_... routines would degrade as values were retrieved. Now, a fixed amount of time should be spent for each retrieval. Made some general performance improvements to improve streaming performance and general performance of the library. Specific improvements were made when a large number of messages and/or items are open at one time. Also, some small improvements were made in general streaming performance. Made an performance improvement in how private attributes are accessed and streamed in. This should help with streaming performance of messages with a large number of private attributes. Made modifications to the Storage SCU sample application to support DICOM asynchronous communications that were added in release 3.4.0. Note that the mergecom.app file must be modified for this functionality to be enabled. Added a new configuration option, IGNORE_JPEG_BAD_SUFFIX. This option will allow MC_Standard_Decompressor to deal with lossless JPEG images whose suffixes have been invalidly written according to the JPEG specification. See the mergecom.pro file for further details. Made changes in MC_Get_Value_To_String when retrieving tags of VR FD. Previously we were using %g in an sprintf() to do this conversion, which would only display a limited number of decimal points. We now use %.16g which will display 16 decimal points. Made modifications to mc3conv so that it can modify tags that have multiple values. The utility can be used to modify tags such as the following: `mc3conv in.dcm out.dcm -tag 0x00080008 ORIGINAL\PRIMARY\AXIAL` Improved the User's Manual documentation for use of registered callback functions. Also added clarification on support of asynchronous communications for SCP applications. Also added some more details on performance tuning in the Frequently Asked Questions section. Updated the library to include the latest version of the Pegasus compression and decompression libraries. Improved temporary file usage to now support storing tags that are almost 4GB in size. (Approximately 3.9GB is allowed.) The toolkit will now store tags in two temporary files that can grow to 4GB in size. Previously, only tags with a length up to 2GB could be supported. ***Defects resolved in Version 3.5.0***

		Fixed problem where the JPEG2000 Expand Pegasus license could not be configured with MC_Set_String_Config_Value function properly. Resolved problem with the DICOM SR API where an SR object could not be converted twice into an SR object. It was failing the second time to recognize the SR tree. Fixed problem that the toolkit could not process association requests where the Association-RQ PDU was larger than our locally configured Maximum PDU Size. Fixed problem where the Deflate & Inflate transfer syntaxes would not work properly when LARGE_DATA_STORE was set to FILE. Fixed problem where MC_FreeSyntaxList was not really freeing the memory associated with the syntax list, and the toolkit would not allow another syntax list to be created using the same syntax list name. Fixed crashing problem with MC_Set_Next_Encapsulated_Value_From_Function. This function would crash if it was called before MC_Set_Encapsulated_Value_From_Function for a tag. Fixed a problem where converting from an SR to message back and forth several times on a single object would not work properly. Fixed memory leak in MC_Duplicate_Message where a memory leak would occur when compressing an image that had more than 1 byte of padding at the end of its pixel data. Resolved defect introduced in the 3.4.0 release where a C-FIND-RSP or C-MOVE-RSP could not be sent after an SCP application received a C-CANCEL-RQ. The call to MC_Send_Response_Message() call would fail. Add new log line when encountering a situation where a tag is encoded in a stream as a VR of UN, and our data dictionary has a definition for the VR of the tag, however, the length of the tag is not a valid length according to the VR we have configured. Changed our validation of DA tags. Previous versions would return MC_INVALID_CHARS_IN_VALUE in some cases for values that were too long instead of MC_INVALID_VALUE_FOR_VR. This has been fixed and some additional validation of DA tags has been added in. Resolved problem where calling MC_List_Message would cause the MC_Get_Next_..() routines to return MC_NO_MORE_VALUES when called the first time on a tag. Now the first value of the tag being retrieved will be returned on the first call to a MC_Get_Next_..() routine for a tag. Resolved problem where if MC_Free_Message() or MC_Free_File() were called on an item, these routines would not properly check the reference count for the item, and instead would automatically free the item.

		Fixed problem with temporary file handling which caused some temporary file access errors when handling compressed images. Fixed problem where the ALLOW_INVALID_PRIVATE_ATTRIBUTES configuration option was not working properly. When this option is enabled, the tool kit was not allowing the streaming in of private attributes that did not have a corresponding private creator code. We now will stream these attributes in when this configuration option is enabled.
3.4.0 IB12	5-JAN-05	***Updates in Version 3.4.0 IB12*** Fixed a memory leak with MC_Duplicate_Message when duplicating between compressed transfer syntaxes. The leak would not be caused when duplicating from uncompressed to compressed or from compressed to uncompressed. Fixed a problem with MC_Validate_Message where the validation would fail if private attributes were defined in the data dictionary for the message being validated and the message was first created as an empty message and then validated.
3.4.0 IB11	21-DEC-04	***Updates in Version 3.4.0 IB11*** Fixed memory leak in MC_Duplicate_Message when duplicating multi-frame objects. As the offset table was updated with each new frame added, the previous buffer containing the offset table was being leaked. Fixed problem where MC_Set_Encapsulated_Value_From_Function would not allow data to be stored to temporary files. IE, LARGE_DATA_STORE was set to FILE, the toolkit would not use temporary files properly.
3.4.0 IB10	13-DEC-04	***Updates in Version 3.4.0 IB10*** Addressed porting issues for SunOS 4.1.4 operating system.
3.4.0 IB9	2-DEC-04	***Updates in Version 3.4.0 IB9*** Addressed porting issues for LynxOS operating system. Addressed compressing/decompressing problem for 8 bits stored, 16 allocated for JPEG_LOSSLESS_HIER_14.
3.4.0 IB8	8-NOV-04	***Updates in Version 3.4.0 IB8*** Addressed porting issues for VxWorks for x86 and made changes to eliminate a problem with the JAVA wrapper toolkit causing JAVA sockets to stop working when releasing or resetting the library.
3.4.0 IB7	15-OCT-04	***Updates in Version 3.4.0 IB7***

		Fixed crashing problem with how the new asynchronous management functionality was implemented. An internal buffer was allocated too small and could cause crashing when the length of the SOP Instance UID of a message being transferred was longer than 53 characters long.
3.4.0 IB6	14-OCT-04	***Updates in Version 3.4.0 IB6*** Fixed problem where new functions MC_Wait_For_Association_On_Port and MC_Wait_For_Secure_Association_On_Port were not being exported properly in the Windows DLL, so they could not be linked against when using mc3adv.dll. Updates for port to VxWorks on Intel x86.
3.4.0 IB5	7-OCT-04	***Updates in Version 3.4.0 IB5*** Updates for the JAVA Wrapper version of MergeCOM-3. Resolved problem with Linux systems where setting the LARGE_DATA_STORE option to FILE on Linux systems did not work properly when using multiple threads to access the DICOM message. MergeCOM-3 was incorrectly determining the names of the temporary files causing problems accessing these attributes.
3.4.0 IB4	4-OCT-04	***Updates in Version 3.4.0 IB4*** Updates for porting to Solaris.
3.4.0 IB3	17-SEP-04	***Defects resolved in Version 3.4.0*** Fixed a longstanding problem in MergeCOM-3 where the toolkit would get "Unknown PDU" errors in the MergeCOM-3 file. The problem arose when the PDU/PDV header was split into more than 1 TCP/IP packet. MergeCOM-3 would on read the bytes in the first TCP/IP packet, time out, and effectively throw out the first bytes, expecting the bytes in the next TCP/IP packet to be part a new PDU/PDV header. Reading of these bytes would cause the subsequent "Unknown PDU" error. This error would mainly occur when short timeouts were used when calling MC_Read_Message. Made some updates to STANDARD_CR image type, some tags were missing from the data dictionary definition. Added tag (0008,0005) to the MPPS N-SET message so that multi-byte characters can be set in these messages. Updated the UID of MR_SPECTROSCOPY in the data dictionary. The UID should have been 1.2.840.10008.5.1.4.1.1.4.2. Also resolved several other issues with how the ENHANCED_MR object was defined in the data dictionary. The mc3info and mc3icomb utilities were not working properly when over 400 messages were

		contained in the configuration files. This limit has been raised to 1200. Added changes to address a problem where DICOMDIRs could only support a limited number of directory records. The limit has now been expanded. Added fix for mc3icomb, where it would not include messages in the mergecom.srv properly. mc3icomb would not add a message it there were two empty messages for a given service. Added fix for mc3valid, where it opens a file twice into the same file pointer variable Compressor now supports 8 bits stored, 16 allocated for JPEG_LOSSLESS_HIER_14. Resolved problem where RT Treatment related tags where not included in the Q/R C-FIND-RQ & C-FIND-RSP messages. Fixed minor memory leak with MC_Library_Reset on Windows. The library was leaving Winsock initialized, which was causing the leak. Note that if an application that utilizes MergeCOM-3 also uses direct Winsock calls, it may end up with problems if it doesn't initialize Winsock itself. Fixed issue with the MergeCOM-3 validation functions. The CS VR did not have a check to make sure an attribute was in upper case, as required by DICOM. Previous versions did not do this check. Fixed issue with MC_Duplicate_Message and the standard compressor where the library would crash when attempting to decompress some JPEG Lossless images that were in an invalid JPEG format. ***Enhancements in Version 3.4.0:*** The MergeCOM-3 data dictionary was updated based on the following change proposals: • 159, 294, 306, 308, 309, 326, 328, 330, 337, 340, 343, 347, 350, 351, 353, 357, 358, 360, 361, 366, 367, 368, 369, 370, 379, 380, 381, 383, 386, 392, 395, 397, 400, 404, 409, 411, 420, 422, 424, 431, 432 The MergeCOM-3 data dictionary was updated based on the following supplements: 42,47,58,66,73,92,93. See the message.txt and mergecom.srv files for details on the names and content of the new services added. Added text based descriptions to Pegasus error codes that are displayed in the MergeCOM-3 log file. This should aid in diagnosis of field problems with the Pegasus libraries. Added support for the JPEG-LS transfer syntaxes (CP-174) and the MPEG2 transfer syntaxes (Sup 42). Note that this does not include support for actual compression in these transfer syntax. The syntaxes can now be set through all of the MergeCOM-3 configuration methods & can be read by default.

Added configuration option, MSG_FILE_ITEM_OBJ_TRACE to the mergecom.pro file. This option allows the tracking of the creation, referencing, and freeing of message, file, and item objects. This option can be used if the user suspects a memory leak in their application from not freeing one of these object types. The logging is done at the T1 trace level & must be enabled in the merge.ini file.

Added configuration option, COMPRESSION_RGB_TRANSFORM_FORMAT, which allows the user to select the output format when doing Lossy JPEG compression of RGB images. The value can be set to YBR_FULL or YBR_FULL_422 to specify what photometric interpretation MergeCOM-3 should compress into when compressing RGB images.

Made changes to MC_Wait_For_Association and MC_Wait_For_Secure_Association so that they can change the port they are listening on. If the MC_Set_Int_Config_Value function is used to change the TCPIP_LISTEN_PORT configuration option between calls to MC_Wait_For_Association or MC_Wait_For_Secure_Association, these routines will stop listening on their previous port, and start listening on the newly configured port.

The AssocInfo and ServiceInfo structures have been expanded to include more association negotiation information. The following is the current format of these structures:

```
typedef struct MC_Service_Info {
    char            ServiceName[50];        /* MergeCOM-3 Service Name */
    TRANSFER_SYNTAX SyntaxType;             /* Transfer syntax negotiated
                                               for the service */
    ROLE_TYPE       RoleNegotiated;         /* The role negotiated for the
                                               service */
       int                              PresentationContextID;
} ServiceInfo;


typedef struct MC_Assoc_Info {
    int     NumberOfProposedServices;        /* From service list */
    int     NumberOfAcceptableServices;      /* Acceptable to both sides */
    char    RemoteApplicationTitle[20];      /* 16-characters max */
    char    RemoteHostName[66];              /* Network node name
                                                64-characters max*/
    int     Tcp_socket;                      /* TCP Socket used for
                                                association */
    char    RemoteIPAddress[66];             /* Network IP Address */
    char    LocalApplicationTitle[20];       /* 16-characters max */
    char    RemoteImplementationClassUID[66];/* 64-characters max */
    char    RemoteImplementationVersion[20]; /* 16-characters max */
       unsigned long LocalMaximumPDUSize;
       unsigned long RemoteMaximumPDUSize;
       unsigned short MaxOperationsInvoked;     /* Negotiated Max operations
                                                 * invoked by the assoc requestor */
       unsigned short MaxOperationsPerformed;   /* Negotiated Max operations
                                                 * invoked by the assoc requestor */
} AssocInfo;
```

Made changes on Solaris to use open()/close() instead of fopen()/fclose(). Solaris only allows 256 files to be opened at one time with fopen() in one process. This change should more concurrency when using threads on Solaris.

		Made change so that the toolkit's log file is now kept open while the toolkit is initialized, instead of it being opened and closed for each log line written. Added support for negotiation of the DICOM Asynchronous Operations Window. MergeCOM-3 can now support DICOM Asynchronous communication, if enabled. The Max Operations Invoked and Performed can now be configured in service lists. See the mergecom.app file for further details. When enabled, a user can send multiple Request messages over an association before receiving a response message. The toolkit checks to make sure the Max Operations Invoked & Performed that were negotiated are not exceeded. Added a new return value, MC_MAX_OPERATIONS_EXCEEDED, to the function MC_Send_Request_Message. This value is returned when the Maximum Operations Performed would be exceeded over the association by sending the request message. Note that this error status does not require aborting of the association. The association is still active. Added improved support for extended negotiation when operating in a threaded environment. The toolkit now has the capability to set extended negotiation information for a given service list in the mergecom.app file, or via dynamic configuration functions. This allows setting different extended negotiation information when connecting to different AE titles. ***New functions added in Version 3.4.0*** Added new function. MC_Dir_Sort. This function can be used sort the directory records within a DICOMDIR so that they are accessed more efficiently. See the reference manual for more details on this function. Added new function, MC_Report_Memory, that reports how much memory MergeCOM-3 is currently using. Calls to this function can be done to determine if MC_Cleanup_Memory should be called. See the reference manual for more details on this function. Added new function, MC_Continue_Read_Message_To_Stream. This function allows a user to read from the network directly to a user callback function, bypassing MergeCOM-3 internal message parsing routines. It can be used to increase performance when reading in a large number of messages from the network. See the reference manual for more details on this function. Added new function, MC_Close_Listen_Port. This function allows the user to cause MergeCOM-3 to stop listening for DICOM connections on a specific port. See the reference manual for more details on this function. Added new function, MC_Get_Listen_Socket_For_Port. This function allows the user to get the listen socket for a specific port that MergeCOM-3 is listening on. See the reference manual for more details on this function. Added new functions, MC_Wait_For_Association_On_Port, and MC_Wait_For_Secure_Association_On_Port. These functions allow an application to listen on a specific port with a specific application ID for associations. Calls can be made to these functions

		from different threads. See the reference manual for more details on these functions. Added new function, MC_NewProposedServiceListAsync, for creating a service list that supports negotiating the DICOM Asynchronous Window. See the reference manual for further details. Also added new function, MC_Set_Negotiation_Info_For_Association, for setting association extended negotiation information when accepting an association. This call sets the negotiation information for one specific association. Added new functions, MC_NewServiceWithExtInfoFromName and MC_NewServiceWithExtInfoFromUID for creating dynamic service lists containing extended negotiation information for use when requesting an association.
3.3.0 IB31	7-JUL-04	***Updates in Version 3.3.0 IB31*** Porting to VxWorks/PPC440
3.3.0 IB30	18-MAY-04	***Updates in Version 3.3.0 IB30*** isFirst not passed to registered callback when offset table was previously passed. Now it is!
3.3.0 IB29	24-MAR-04	***Updates in Version 3.3.0 IB29*** Lossy decompression works correctly for 10 bit image. Updated for javawrapper - runtime exception occurred when reading in an image from the network that contained an offset table.
3.3.0 IB28	15-MAR-04	***Updates in Version 3.3.0 IB28*** Ported to Solaris 2.4 - 2.8
3.3.0 IB27	20-FEB-04	***Updates in Version 3.3.0 IB27*** New toolkit built for Solaris 8
3.3.0 IB26	12-FEB-04	***Defects resolved in Version 3.3.0 IB26*** MC_Get_pValue..., MC_Get_pValue_To_Function, MC_Get_Next_pValue..., MC_Get_pValue_To_Buffer, MC_Get_pValue_Count, MC_Get_pValue_Length were updated to not allow NULL to be passed as a pointer.
3.3.0 IB25	14-JAN-04	***Defects resolved in Version 3.3.0 IB25*** This update is for NESTED pixel data. Nested pixel data is any occurrence of 7FE0,0010) that is not in the root level of the message. It may be contained in the icon image, or a private sequence. All nested pixel data is treated the same, whether Icon Image, or within a private attribute. The

		format of the data contained in ANY nested 7FE0,0010 is dictated by the transfer syntax of the message. When the message is non-jpeg, then the nested pixel data must be of the same non-jpeg format. When the message is jpeg, then the length determines if the pixel data is compressed. If the length is undefined, then the nested pixel data is compressed in the same jpeg format as the rest of the message. If it is defined, then it is in explicit little endian.- VR of the destination message's pixel data is changed to OW in MC_Duplicate_Message when a source message is being decompressed into a destination message, and the bits allocated (0028,0100) is greater then 8.
3.3.0 IB24	29-DEC-03	***Defects resolved in Version 3.3.0 IB24*** MC_Set_pValue..., MC_Set_Next_pValue... functions now return MC_NULL_POINTER_PARM when the private code, or value is a null pointer.
3.3.0 IB23	23-DEC-03	***Defects resolved in Version 3.3.0 IB23*** The buffers returned from the MC_Open_File callback could introduce an unchecked boundary condition when the offset table was split between buffers.
3.3.0 IB22	1-DEC-03	***Defects resolved in Version 3.3.0 IB22*** Made DIR_REC_PRESENTATION a valid sub record of DIR_REC_SERIES and DIR_REC_TOPIC.
3.3.0 IB21	26-NOV-03	***Enhancements in version 3.3.0 IB21*** Added REMOVE_SQ as a configuration parameter to force MC_Set_Value_To_Empty and MC_Set_Value_To_Null to remove unreferenced sequence items. Previously, the item ID would still be valid after these calls.
3.3.0 IB20	19-NOV-03	***Defects resolved in Version 3.3.0 IB20*** MC_Duplicate_Message could incorrectly return MC_OFFSET_TABLE_TOO_SHORT if MC_Get_(Next_)Encapsulated_Value was called previously with the same message. Only occurred with certain combinations of frame size, and the buffer's size returned to MC_Open_File from the user callback. Testing showed this bug with small frame size (64x64) and buffer size >3k where multiple frames would be contained in a single returned buffer.
3.3.0 IB19	12-NOV-03	***Defects resolved in Version 3.3.0 IB19*** In certain instances, using COMBINE_HEADER_WITH_DATA could cause hang.
3.3.0 IB18	6-NOV-03	***Defects resolved in Version 3.3.0 IB18*** NETWORK_CAPTURE would default to enabled, and REWRITE_CAPTURE_FILES wouldn't set based on the value in mergecom.pro

3.3.0 IB17	3-NOV-03	***Defects resolved in Version 3.3.0 IB17*** Duplicating a NULL attribute via MC_Duplicate_Message caused an exception Odd length files/streams could be produced when DEFLATED_EXPLICIT_LITTLE_ENDIAN. ***Enhancements in Version 3.3.0 IB17*** Updated dictionary/message files
3.3.0 IB16	8-OCT-03	***Enhancements in v3.3.0 IB16*** Ported to HPUX11
3.3.0 IB15	30-SEP-03	***Defects resolved in Version 3.3.0 IB15*** MC_Standard_Compressor/Decompressor releases all memory every frame. Previously would reuse, but never free. Leak - When emptying the pixel data tag for an encapsulated value, would not free the OffsetTable Leak - When streaming an encapsulated value in, the OffsetTable pointer would be overwritten without being freed. STANDARD_NM and STANDARD_NM_RETIRED UIDs were swapped G_P_SCHEDULED_PROCEDURE_STEP had some attributes incorrectly assigned type 3.
3.3.0 IB14	26-AUG-03	***Defects resolved in Version 3.3.0 IB14*** Linux only - Toolkit now will build on Red Hat 9
3.3.0 IB13	14-AUG-03	***Enhancements in Version 3.3.0 IB13*** Windows only - A spin count can be set for critical sections for multiprocessor systems. This will typically improve threading performance. Set an environment variable, MC3_SPIN_COUNT to use this feature.
3.3.0 IB12	1-AUG-03	***Defects resolved in Version 3.3.0 IB12*** MSG_FILE_ITEM_OBJ_TRACE config parm added to allow the tracking of message/file/item object creation/referencing/freeing. T1 must be turned on in merge.ini The reference count on a sequence contained in a file opened with MC_Open_File would be 0, so it could be deleted with MC_Free_Item even though the file was referencing it.
3.3.0 IB11	29-JUL-03	***Defects resolved in Version 3.3.0 IB11***

		LARGE_DATA_STORE = FILE would stream out the wrong data length Attributes for General Purpose Scheduled Procedure Step, N_ACTION_RQ (0008,1195) and (0040,4001) incorrectly specified as type 3.
3.3.0 IB10	24-JUL-03	***Defects resolved in Version 3.3.0 IB10*** ServiceList names can now have up to 129 characters. A Decompressed image with 8 bits stored, 16 allocated could contain garbage data in it's non-significant 8 bits. MC_Read_Message_To_Tag, MC_Continue_Read_Message_To_Tag, and MC_Continue_Read_Message could fail for long message transfers when the remaining data is contained in more then one PDU. LARGE_DATA_STORE = FILE could cause MC_Duplicate_Message to return a message that wouldn't work with other toolkit APIs (MC_Write_File, MC_Message_To_Stream, etc)
3.3.0 IB9	13-JUL-03	***Defects resolved in Version 3.3.0 IB9*** Pegasus could crash on pre-Pentium III machines when using JPEG_2000 decompression With an image as described below, the toolkit would crash. It now returns an error. Certain lossless JPEG images written by, for example, some versions of Lead tools or Osiris, are invalid images according to the JPEG spec. They have a 16-zero-bit suffix following a -32768 prefix where the JPEG spec says the suffix is omitted following a -32768 prefix.
3.3.0 IB8	27-JUN-03	***Enhancements in Version 3.3.0 IB8*** TCPIP_DISABLE_NAGLE added to give the ability to disable the Nagle Algorithm COMBINE_DATA_WITH_HEADER combines the 12 byte header with data if combined size will be less then 256 bytes. Item reference counters added so an item referenced in 2 messages won't be removed if the first message is freed. MC_Get_Tags_Dict_Info added to get dictionary information for a tag ***Data Dictionary Changes in 3.3.0 IB8*** Updated to 2003 DICOM standard info.pfl had lines with only ">;" and could not be parsed

		DIR_REC_KEY_OBJECT_DOC added DIR_REC_PRESENTATION was missing group 4 attributes Typos in diction.h fixed and aliased (both new and old work), diction.pfl names corrected Corrected VRs for Start Cumulative Meterset Weight(300C,0008), End Cumulative Meterset Weight(300C,0009),Tolerance Table Number(300A,0042), Referenced Brachy Application Setup Number(300C,000C) Instance Availability Tag (0008,0056), High Dose Technique Type(300A,00C7) was missing Fixed VR of (0040,DB73)Referenced Content Item Identifier SC_MULTIFRAME_TRUE_COLOR definition was incomplete STANDARD_CT & STANDARD_CR were missing Exposure in uAs (0018,1153) ***Defects resolved in Version 3.3.0 IB8*** MC_NewProposedServiceList detects duplicate list (existing in mergecom.app) Dynamic service lists could fail in multi-threaded environment duplicate.c sample program could overwrite output files decompressor isLast flag is sent before a release (see user manual) MC_Set_Next_Value_From_String accepts empty strings (NULL) as a value DIR_REC_KEY_OBJECT_DOC now acceptable lower record of SERIES and TOPIC MC_Duplicate_Message handles case when LARGE_DATA_STORE = 'F' MC_SR_Delete_Child now deletes all children underneath it. Conditional Tags based on VALUE_TYPE (0040,A040) fixed Validation of optional values for (0008,0008) does not return an error Icon Image can be encapsulated or unencapsulated Role is sent only once per SOP UID.
3.2.1 IB9	13-MAY-03	***Issues in 3.2.1 IB9*** Ported to OSF/1 on Dec Alpha (64bit)

3.2.1 IB8	15-APR-03	***Version 3.2.1 IB8*** AT attributes could cause validation error if first value was 0x20. This as being mistaken for a leading space. Ported to Windows, Linux, and Solaris
3.2.1 IB7	11-APR-03	***Version 3.2.1 IB7*** Ported to AIX
3.2.1 IB6	11-APR-03	***Defects resolved in Version 3.2.1 IB6*** MC_FreeSyntaxList could attempt to free already freed memory.
3.2.1 IB5	1-APR-03	***Defects resolved in Version 3.2.1 IB5*** Double to DS was failing in some cases. MC_Set_Value would return INVALID_VALUE_FOR_VR
3.2.1 IB4	20-MAR-03	***Defects resolved in Version 3.2.1 IB4*** Duplicate was not working on messages/sequences with no attributes.
3.2.1 IB3	25-FEB-03	***Defects resolved in Version 3.2.1 IB3*** ICON_IMAGEs can now be compressed or uncompressed. The standard (de)compressor can also be used to get/set encapsulated values. A duplication will default to unencapsulated ICON_IMAGE unless DUPLICATE_ENCAPSULATED_ICON is set to yes in mergecom.pro. Also, an encapsulated stream will not change the ICON_IMAGE length to undefined (bugfix).
3.2.1	11-FEB-03	***Defects resolved in Version 3.2.1*** Deflated Explicit Little Endian format was incorrectly containing a zlib header and tail. Tags, offsets, and lengths within encapsulated data were incorrect when using MC_Set_(Next_)Encapsulated_Value_From_Function. When message was streamed, the incorrectly swapped values would be sent. (Big Endian machines only) PDU_MAXIMUM_LENGTH larger then negotiated Maximum PDU Length would cause the last 2 PDUs to be sent with Last bit set in message control header. T5 log messages were incorrectly being generated for unrequested validation errors. MC_Set_Value_From_Float(msgID,<tag w/VR=DS>,<Value with lead 0 like 0.1>) will not return MC_INVALID_VALUE_FOR_VR.
3.2.0 IB8	8-JAN-03	***Defects resolved in Version 3.2.0 IB8:*** Tags, offsets, and lengths within encapsulated data were incorrect when using

		MC_Set_(Next_)Encapsulated_Value_From_Function. When message was streamed, the incorrectly swapped values would be sent.
3.2.0 IB5	18-DEC-02	***Defects resolved in Version 3.2.0 IB5:*** PDU_MAXIMUM_LENGTH larger then negotiated Maximum PDU Length would cause the last 2 PDUs to be sent with Last bit set in message control header.
3.2.0 IB2	19-NOV-02	***Defects resolved in Version 3.2.0 IB2:*** T5 log messages were incorrectly being generated for unrequested validation errors.
3.2.0 IB1	15-NOV-02	***Defects resolved in Version 3.2.0 IB1:*** MC_Set_Value_From_Float(msgID,<tag w/VR=DS>,<Value with lead 0 like 0.1>) will not return MC_INVALID_VALUE_FOR_VR.
3.2.0	4-NOV-02	***Defects resolved in Version 3.2.0:*** Double conversion to DS lost some significant digits Logging of proposed transfer syntaxs was not working correctly due to padding at the end of transfer syntax string Memory leak could occur when streaming a message over an existing message. Memory leak could occur when recalling MC_Validate_Attribute Memory overwrite when IMPLEMENTATION_VERSION was too long in mergecom.pro MC_Dir_Entity_Count returning incorrect value when MC_Dir_Add_Entity used to create entity MC_Get(_Next)_Value_To_Float failing when internal strtod fails. Streaming a multivalued, non OB/OW/OF/SQ, attribute could exceed the size allocated to Explicit VR (2-byte length). Now returns MC_CALLBACK_CANNOT_COMPLY. Memory leak in Unix network capture facility NULL sequences would cause MC_Duplicate_Message to fail Attributes with VR = UN, and undefined length caused assertion when streaming MC_Get/Set(_Next)_Encapsulated_Value..., and MC_Register_Compression_Callbacks not checking a file's transfer syntax correctly. Type 3 attributes with VM>1 with a single NULL value were failing validation. The handling of

		this situation is now controlled with NULL_TYPE3_VALIDATION Calling MC_Get_Next_Encapsulated_Value... after it returned MC_NORMAL_COMPLETION would cause unpredictable behavior. Now returns MC_NO_MORE_VALUES. MC_TIMEOUT wasn't being handled properly through the secure association calls. mrgcom3.msg could potentially not be closed in an error condition. User could set values for an attribute with VM = 1-n until internal object's value count overflowed. NOTE: Once you get anywhere near this point, you probably will have to send the message IMPLICIT_LITTLE_ENDIAN because it's length won't fit in 2 bytes (EXPLICIT_LITTLE_ENDIAN). MC_Byte_Swap_OBOW wouldn't handle OB correctly. Now returns MC_NORMAL_COMPLETION without doing anything. MC_ServiceList not freeing ServiceList correctly. Compressor failure when more the one frame presented from user callback. Decompressor failure when entire file returned from user callback. ***Data Dictionary Changes in Version 3.2.0:*** Instance Availability Tag Directory Record Type should be required (1 or 1C) for RT_STRUCTURE_SET G_P_WORKLIST_MANAGMENT_META incorrect in mergecom.srv Attribute 0040,DB73 is US instead of UL per the standard "300A, 00C7" -- Type of high-dose treatment technique not in dictionary, but in 2001 standard SC_MULTIFRAME_TRUE_COLOR missing everything. ***Enhancements in Release 3.2.0:*** Dictionary updated for Enhanced MR (Supp. 49) Reverse DNS lookup eliminated when ACCEPT_ANY_HOSTNAME = Yes mc3sr.h folded into mc3media.h Improved logging of association rejections (hostname/ip)

		MC_Get_Meta_ServiceName added to get the META SOP of the last in coming request message.
		
		Deflated Explicit VR Little Endian added as transfer syntax. Deflating and inflating is automatically handled by the toolkit.
		
		JPEG 2000, JPEG 2000 Lossless Only added as transfer syntax. This transfer syntax is not currently handled by the standard compressor/decompressor.
		
		Configurable Deflate parameters added
		`DEFLATE_ALLOW_FLUSH`
		`FLATE_GROW_OUTPUT_BUFFER_SIZE`
		`DEFLATE_COMPRESSION_LEVEL)`
		
		Standard Compressor/Decompressor (Windows, Solaris 2.7 and newer, Linux):
		
		Latest version of the Pegasus libraries (Windows, Solaris, Linux)
		
		Ability to set Luminance and Chrominance factor of MC_Standard_Compressor
		`(COMPRESSION_LUM_FACTOR, COMPRESSION_CHROM_FACTOR)`
		JPEGs can be output non-fragmented using COMPRESSION_ALLOW_FRAGS configuration parameter
		
		Compression sample application duplicate.c

3.1.0	25-MAR-02	***Defects resolved in Version 3.1.0:*** MC_NewSyntaxList now accepts an array of TRANSFER_SYNTAX instead of less intuitive ints. When reading media files and then sending them out as messages, the toolkit erroneously attempted to send the message using the transfer syntax used in the media file rather than the one negotiated. Temporary files were incorrectly created when LARGE_DATA_STORE=FILE was used. MC_Duplicate_Message incorrectly duplicated pixel data when stored in callbacks. Resolved write past end of buffer when logging, and when retrieving config info from mergecom.srv file work_scp had memory leaks due to not freeing messages mc3file updated UIDs and Service Commands. Under heavy multi-threaded use the library could sometimes deadlock on a thread semaphore ***Enhancements in Release 3.1.0:***

		MC_Register_Enhanced_MemoryLog_Function added. It registers a callback function that will be notified as each log message is written. Gives more information then the standard MC_Register_MemoryLog_Function call.
3.0.1	21-FEB-02	***Defects resolved in Version 3.0.1:*** Red/blue color swapping in color jpeg images by the standard compressor. Improper handling of some valid images size by the standard compressor. File handle problem following repeated release and re-initialization of library. Access violation during toolkit library internal memory management.

3.0.0	14-JAN-02	***Enhancements in Release 3.0.0:***
		This release of the toolkit introduces enhanced support for compression and decompression of DICOM images.
		A Pegasus compressor/decompressor module is included with the following toolkits:
		• 008-91204 - Win32 platform using Microsoft C compiler
		• 008-91119 - Sun Solaris 2.7 and above
		• 008-91126 - Red Hat Linux for x86 platform
		The entry points to the provided Pegasus module are the new functions MC_Standard_Compressor, and MC_Standard_Decompressor which must be registered to a message using the new function MC_Register_Compression_Callbacks.
		All platforms may develop their own compressor/decompressor callbacks and register them using the new MC_Register_Compression_Callbacks function. The User Manual provides information about developing compressor/decompressor callbacks.
		Using the registered callbacks, data is also encapsulated for you in the message using these new functions:
		`MC_Close_Encapsulated_Value`
		`MC_Get_Encapsulated_Value_To_Function`
		`MC_Get_Next_Encapsulated_Value_To_Function`
		`MC_Set_Encapsulated_Value_From_Function`
		`MC_Set_Next_Encapsulated_Value_From_Function`
		The Set/Get functions set/return the first frame of an encapsulated message, and an internal pointer moves to each subsequent frame when the Set_Next/Get_Next functions are called. MC_Close_Encapsulated_Value is used to terminate an encapsulated message.

		A typical use of these functions would be:
		
		1) Register the Standard Compressor/Decompressor (or your own compressor) with a message.
		
		2) MC_Set_Encapsulated_Value_From_Function for the first frame.
		
		3) MC_Set_Next_Encapsulated_Value_From_Function for any subsequent frames.
		
		4) MC_Close_Encapsulated_Value
		
		A message may now be read up to a tag, or continued to a tag, allowing for a message to be read in up to it's pixel data, and then decide what to do with it before going and getting it. A message must be fully read before getting the next one, regardless.
		
		`MC_Read_Message_To_Tag`
		`MC_Continue_Read_Message`
		`MC_Continue_Read_Message_To_Tag`
		
		MC_Duplicate_Message has been added to allow for creating an exact duplicate of the original. If either message is changed after a duplicate, UIDs should be changed also. The destination transfer syntax may be different from the source's.
		
		This release of the toolkit introduces support for Part 15 of the DICOM standard: Security Profiles. The toolkit now allows you to write a special security handler that is responsible for transmitting and receiving network packets to/from the toolkit. The handler can thus provide any appropriate security mechanism.
		
		Two new calls (MC_Open_Secure_Association and MC_Wait_For_Secure_Association) are provided. The calls allow you to provide a security handler by providing a set of callback functions that the toolkit uses to provide and receive network data. It is the responsibility of the callback to transmit and receive network data on sockets provided by the toolkit. A sample security handler is provided that implements OpenSSL.
		
		This release of the toolkit provides an option to capture network packets sent and received by the toolkit in a file that can be read by the MergeDPM analysis product. This feature avoids the need to use a separate network capture program for use with the toolkit. It also allows visibility of network information when secure sockets are being used.
		
		A standard network capture facility is provided and can be turned on and off through options contained in the mergecom.pro file. Options also allow you to customize the behavior of the network capture.
		
		A new function (MC_Register_Network_Capture_Callbacks) allows you to register a set of functions that will be used instead of the standard network capture function, although it is unlikely you will need to override the standard handler.
		
		The data dictionary and message files were updated for consistency with the 2001 Edition of the

		DICOM Standard. The following supplements were included: • 40 DVD using UDF Media • 41 Security Enhancements Two • 48 Intravascular Ultrasound IOD and Application Profile • 50 Mammography Cad • 51 Media Security • 52 General Purpose Worklist • 53 DICOM Content Mapping Resource • 57 Expanded Secondary Capture • 59 Key Object Selection SOP Class
2.5.3	12-SEP-01	
2.5.2	3-APR-01	

2.5.1	13-MAR-01	***Enhancements in Release 2.5.1:***
		MC_Send_Request_For_Service was added to the API to allow the specification of a service to facilitate the selection of presentation contexts by the toolkit. The previous toolkit design required the use of two separate associations for Basic Grayscale Print Management Meta SOP and Basic Color Print Management Meta SOP. Using this new call, grayscale and color print can be sent on the same association.
		The toolkit is now capable of dynamic generation of service lists used in making an association. Previously, the service lists needed to be pre-defined in the mergecom.app file. The following calls were added to the API; please refer to the Reference Manual for details on usage.
		`MC_NewSyntaxList`
		`MC_FreeSyntaxList`
		`MC_NewServiceFromUID`
		`MC_NewServicetFromName`
		`MC_FreeService`
		`MC_NewProposedServiceList`
		`MC_FreeProposedServiceList`
		Data Dictionary Updates in 2.5.1
		Updated data dictionary to correct some attribute names and insert some recently added attributes for consistency with the 2000 edition of the DICOM standard. Note: Aliases for the previous names were added to diction.h to ensure backward compatibility.

		Added Basic Annotation Box SOP Class to the message file and mergecom.srv.
		
		The SOP class was inadvertently removed from the version 2.5.0 definitions.
		
		Added the following sequences and attributes to the definition for Basic Film Box N-CREATE-RQ and N-CREATE-RSP messages: (2010,0510) (2010,0520) (2020,0050)
		
		Added the following attributes to Performed Procedure Step N-CREATE-RQ, N-CREATE-RSP, N-SET-RQ, N-SET-RSP messages: (0040,0280) (0040, 0306) (0040,8302)
		
		Added to following attributes to the definitions for the Basic Grayscale Image Box N-SET-RQ and N-SET-RSP messages: (2010,0120) (2010,0130) (2010,0150)
		
		Added all valid 2-character enumerated values to the definition for Patient Orientation (0020,0020) attribute.
		
		Added attributes to the definitions for the following IOD modules for consistency with the 2000 edition of the DICOM standard: Radiation Dose Module, X-ray Acquisition Dose Module, X-ray Generation Module, Mammography Image Module, SOP Common Module, RT Image Module, ROI Contour Module.
		
		Added definitions for the following print status codes to mergecom.h: "Warning: Attribute Value Out of Range" 0x0116, "Warning: Attribute List Error" 0x0107.
		
		Bug Fixes in Release 2.5.1:
		
		Corrected a memory fault error on the Window 95/98/NT platform when A-ASSOCIATE-RQ PDUs with presentation context sub-items that have an ID of 0 were received. Although 0 is an invalid value, SCUs in the field have been observed using it. The toolkit now aborts this association attempt.
		
		Updated the reference manual with some corrections and clarifications.
		
		Descriptions of the new API features were also added.
		
		Added support for port numbers greater than 32767 on 16-bit platforms.
		
		Corrected handling of negative values by MC_Set_Value_From_String functions when the target VR is FL. Affected are:
		
		`MC_Set_Value_From_String`
		`MC_Set_Next_Value_From_String`
		`MC_Set_pValue_From_String`
		`MC_Set_Next_pValue_From_String`
		
		Corrected MC_Message_To_SR to report MC_INVALID_MESSAGE_ID when a bad message ID is passed.

		Added enhancements to the SR functions to ensure they are tread-safe. Enhanced MC_SR_Delete_Child so that it can't be used to inadvertently delete the root objects; added correction so it properly deletes the first child in the list. Corrected the validation of attributes with US, SS and UL value representations and a value of 32. (Example: the toolkit reported a validation error of "Value not an enumerated value" when Bits Allocated had a value of 32.) Corrected MC_Validate_Attribute to properly load validation information for messages created by MC_Open_Empty_Message. (A previous work-around required the use of MC_Validate_Message to correctly load the validation information.)

2.5.0	15-SEP-00	***Issues when upgrading to Release 2.5.0***
		
		The Structured Reporting Suppliment (suppliment 23) implemented in release 2.4.1 of the toolkit was not in final text form at the time of the toolkit release. The version 2.5.0 of the toolkit now incorporates suppliment 23 in its final text form.
		
		MC_Get_First_Acceptable_Service and MC_Get_Next_Acceptable_Service were documented incorrectly in the Reference Manual. The Reference Manual has been changed to reflect a bug fix made in release 2.4.1 (see below - "Long service names handled incorrectly")
		
		Enhancements in Release 2.5.0:
		
		Added structured reporting APIs. See below.
		
		New functions added in Release 2.5.0:
		
		The following functions were added to the toolkit to allow users to manipulate SR content sequence macros in a more straightforeward manner:
		
		`MC_Message_To_SR`
		`MC_SR_To_Message`
		`MC_SR_Get_Root`
		`MC_SR_Add_Child`
		`MC_SR_Add_Root`
		`MC_SR_Get_First_Child`
		`MC_SR_Get_Next_Child`
		`MC_SR_Delete_Child`
		`MC_SR_Get_Location`
		
		Bug Fixes in Release 2.5.0:
		
		The Windows toolkit has been fixed so that when attempting to run two SCPs on the same listen port an error occurs. UNIX platforms (and now the Windows platform, after the fix) log errors similar to the following:
		
		`(13820) 09-14 15:35:54.67 MC3(TL_Start_Transport_Listener) E: Failure binding listen port`
		`(13820) 09-14 15:35:54.77 MC3(TL_Start_Transport_Listener) E: Port 104 already in use`

		When using MC_Open_Association with a given service list parameter, the toolkit used to cache this service list such that subsequent calls to MC_Open_Association with the service list as NULL would not retrieve the default service list. This problem has been fixed. Fixed problem where the linkage between a particular message and the message's specific character set was being lost. Basically, when setting the (0008,0005) tag a second time, any previous setting of the specific character set attribute would be lost. The functionality to allow environment variables specified in Windows format added in release 2.4.1 of the toolkit was flawed. This has now been fixed so that entering environmental variables as %TEMP% instead of %$(TEMP) will function. Fixed a bug in the work_scp.c where setting the value of the referenced image sequence to empty within a MPPS response message would fail. This was because the code attempted to set the value of regerenced image sequence to empty in the root message when this value should have been contained within a sequence.

2.4.2	9-MAY-00	***Issues when upgrading to Release 2.4.2:***
		
		Addition of DESIRED_LAST_PDU_SIZE configuration value: In order to provide inter-operability with other DICOM implementations, the last PDU's length has been made configurable.˜ This will allow the user to configure the toolkit to operate with other DICOM implementations that are intolerant of either zero or two byte PDU's that are sent as the last PDU.
		
		The Maximum size allowed is 16 and the default value is 8. This configuration value can be added to the mergecom.pro file in the MESSAGE_PARMS section if incompatibilities arise.
		
		Enhancements in Release 2.4.2:
		
		Added support for the following DICOM Supplement to the data dictionary:
		
		• Supp 30: Waveform Interchange
		
		The DICOM standard has changed the names of some data elements.˜ The new names have been added to the diction.h file distributed with the tool kit. To accommodate these name changes without having a negative impact on existing code, the replaced names are included at the end of the diction.h file as "alias" names. Except for the availability of the old names in diction.h, all other references to these data elements will use the new names.˜ For example, MC_List_Message will show the new names for attributes.
		
		The alias names added to the end of diction.h for this release are:
		`Old Name New Name`
		`---------- ----------`
		`MC_ATT_IMAGE_DATE MC_ATT_CONTENT_DATE`
		`MC_ATT_IMAGE_TIME MC_ATT_CONTENT_TIME`
		`MC_ATT_IMAGE_NUMBER MC_ATT_INSTANCE_NUMBER`
		`MC_ATT_NUMBER_OF_PATIENT_RELATED_IMAGES MC_ATT_NUMBER_OF_PATIENT_RELATED_INSTANCES`

		`MC_ATT_NUMBER_OF_STUDY_RELATED_IMAGES MC_ATT_NUMBER_OF_STUDY_RELATED_INSTANCES`
		`MC_ATT_NUMBER_OF_SERIES_RELATED_IMAGES MC_ATT_NUMBER_OF_SERIES_RELATED_INSTANCES`
		
		New functions added in Release 2.4.2:
		
		The following functions were implemented in order to provide statistics about DICOMDIR objects:
		
		`MC_Dir_Root_Count`
		`MC_Dir_Entity_Count`
		`MC_Dir_Item_Count`
		
		The functionality and usage of these API's is documented in the Reference Manual (Refer.pdf) included with this release of the toolkit.

2.4.1	7-FEB-00	***Issues when upgrading to Release 2.4.1:*** Trailing zero-length PDUs: Release 2.4.0 introduced sending over the network a trailing PDU with a PDV length of zero. While this is supported by the DICOM standard some implementations were intolerant of the zero-length PDVs. Release 2.4.1 avoids sending these zero-length PDVs to accommodate such implementations. Calling MC_Abort_Association from another thread: It is now possible to call MC_Abort_Association while in the process of sending a message in another thread. However, this affects some other functions. The following functions are affected: MC_Send_Request_Message, MC_Send_Response_Message, MC_Read_Message, MC_Close_Association, and MC_Abort_Association. Each of these functions now return MC_SYSTEM_ERROR if they are called at the same time from multiple threads. If MC_Abort_Association is called when MC_Close_Association is being called in another thread, MC_Abort_Association will return with MC_NORMAL_COMPLETION, and will let MC_Close_Association to continue executing until it has completed. If MC_Abort_Association is called when another thread is in MC_Read_Message, it will return immediately. When the MC_Read_Message call has (almost) completed execution, i.e. if there is a timeout, or it has completed receiving a message, it will then abort the association, and return with the MC_ASSOCIATION_ABORTED return value. If MC_Abort_Association is called when MC_Send_Request_Message or MC_Send_Response_Message is in use, it will return with MC_NORMAL_COMPLETION. When the other thread has completed sending the PDU that is currently being sent, it will send an Abort PDU, and subsequently drop the connection. Depending on the maximum PDU size negotiated, it may take some time for this to occur. When this occurs MC_Send_Request_Message or MC_Send_Response_Message will return with MC_ASSOCIATION_ABORTED. There will also be a log saying another thread has requested that the association be aborted. This was done such that the abort is properly formatted in DICOM, and could be recognized by the side receiving

		the message. ***Enhancements in Release 2.4.1:*** Updated the Reference Manual, User's Manual, and Sample Applications Guide to reflect changes since release 2.4.0. Added support for the following DICOM Supplements to our data dictionary: • Supp 23: Structured Reporting Object • Supp 33: Softcopy Presentation State The mc3file utility now generates image files for all services supported by the current released mergecom.srv file. MC_Set_Value_Representation now allows changing attributes with a VR of US to either OW or SS, and it now allows changing attributes with a VR of SS to either OW or US. This change was made to support some of the latest DICOM dictionary changes which allow some attributes to have different value representations (VRs). MC_Validate_File, MC_Validate_Attribute and MC_Validate_Message now validate messages for the new services added (see above). ***New utility program in Release 2.4.1:*** For customers using the Extended MergeCOM-3 Advanced Tool Kit to customize the data dictionary and info files, two new utilities have been added to simplify the process of adding and maintaining data dictionary changes. The utilities allow you to maintain your custom changes in separate text files and then combine them with the released data dictionary and info files automatically. The utilities (mc3dcomb and mc3icomb) are described fully in the updated Message Database Manual distributed with Extended Tool Kits. ***New functions added in Release 2.4.1:*** A new function (MC_Parse_Association) was added to Unix tool kits. This function is used when the inetd daemon is used to wait for association connections on the DICOM port. The inetd daemon accepts incoming connections and then forks and exec's the DICOM SCP application. MC_Parse_Association is called by the application to parse the association request received. In such applications this function replaces the use of the MC_Wait_For_Association call. Check the Reference manual for a complete discussion of the new function. ***New configuration options added in Release 2.4.1:*** Two new boolean configuration parameters have been added to the [MESSAGE_PARMS] section of the mergecom.pro file to accommodate applications which use locale settings demanding a

		comma, instead of a period, for the floating point radix character.
		
		`RETURN_COMMA_IN_DS_FL_FD_STRINGS = Yes|No`
		
		If this parameter is set to Yes, the tool kit will return a comma as the radix character in the value when MC_Get_Value_To_String is called for an attribute with a VR of DS, FL or FD. If No (the default), a period will always be returned as the radix character. Note that DS values will always be properly encoded with a period in DICOM message objects.
		
		ALLOW_COMMA_IN_DS_FL_FD_STRINGS = Yes|No
		
		If this parameter is set to Yes, a comma or a period will be allowed in the value passed to MC_Set_Value_From_String for attributes with a VR of DS or FL or FD. If No (the default), only a period will be acceptable as the radix character. Note that the tool kit will always insure that DS attributes use a period radix character when streaming to the network or to a file, regardless of current locale settings.
		
		Use of percent sign to specify environment variables: In the past configuration parameters which specified file path names could embed $(name) in the parameter value to request that an environment variable value be substituted for $(name). For example, if MC3CONFIGDIR would be set to /usr/merge/mergecom3, then the following could be used in the merge.ini file:
		
		`MERGECOM_3_PROFILE = $(MC3CONFIGDIR)/mergecom.pro`
		`MERGECOM_3_SERVICES = $(MC3CONFIGDIR)/mergecom.srv`
		`MERGECOM_3_APPLICATIONS = $(MC3CONFIGDIR)/mergecom.app`
		
		Now, in addition to the $(name) protocol, the environment variable name may be specified using percent signs. For example, the following may be used:
		
		`MERGECOM_3_PROFILE = %MC3CONFIGDIR%/mergecom.pro`
		`MERGECOM_3_SERVICES = %MC3CONFIGDIR%/mergecom.srv`
		`MERGECOM_3_APPLICATIONS = %MC3CONFIGDIR%/mergecom.app`
		
		Either of these methods would be equivalent to the following:
		
		`MERGECOM_3_PROFILE = /usr/merge/mergecom3/mergecom.pro`
		`MERGECOM_3_SERVICES = /usr/merge/mergecom3/mergecom.srv`
		`MERGECOM_3_APPLICATIONS = /usr/merge/mergecom3/mergecom.app`
		
		Bug Fixes in Release 2.4.1:
		
		A bug was introduced in Release 2.4.0 which caused MC_Free_File or MC_Free_Message to take an extremely long time to release the resources used by message objects which contained very large DICOMDIRs or a great many embedded sequence of items. This has been fixed.
		
		The data dictionary now includes some missing tags.

		Change to the work_scp sample application: The work_scp sample application in the past automatically created a SOP Instance UID should the SCU not send one in an N_CREATE request message. This behavior was not correct since it is required that the SCU fill in this field in the request. The sample now fails the N_CREATE if the SCU does not send this information . Documentation updates: The Message Database Manual now warns that it is necessary to add services to mergecom.srv before the meta services listed, when extending the data dictionary. The Reference manual now correctly notes that Windows tool kits must use different function arguments for MC_List_Message, MC_List_Item and MC_List_File. Handling missing sequence delimiter tag: If the sequence delimiter tag is missing at the end of an encapsulated formatted pixel data tag, the library failed with an error starting with release 2.3.2. Now, if the end of the file is encountered without the final sequence delimiter, the library only issues a warning. Sample apps incorrectly checked for MC_UNKNOWN_HOST_CONNECTED: Several of the sample programs incorrectly checked for an MC_UNKNOWN_HOST_CONNECTED status code. Since no such code will ever be returned from the library, this check has been removed from the sample applications. work_scu sample failed to check status: This sample application now checks for good status after all library calls. Improper checking of isLast flag from registered callback: In version 2.4.0, when a registered callback is used to provide data there was a case where the isLast flag set by the callback is ignored. This happened when isLast is set to true, but no data is actually being supplied via the callback. Now, the callback is repeatedly called until the isLast flag is set. Invalid validation error: When validating a US tag with enumerated values of 0x0000 and 0x0001 the tool kit failed the validation. This error occurred with attributes that have a value multiplicity of > 1 (ie 1-N). This is now fixed. (This bug was introduced in the 2.3.0 release.) Long service names handled incorrectly: If a service name exceeded 34 characters it caused problems when using the MC_Get_Next_Acceptable_Service call. Service names are now limited to 50 characters and, if a longer name is encountered, a warning message is issued and the name is truncated. Lost specific character set value: If the specific character set attribute (0008,0005) was set more than one time for a given message, the tool kit lost track of the specific character set value and thus failed when validating attributes. This has been fixed. mc3echo failure: The mc3echo program failed to connect in situations where no DNS server exists. This seems to have been introduced in Release 2.4.0 and has been resolved. mc3file generated invalid UIDs: The mc3file utility sometimes generated invalid UIDs (with

		leading zeros in a field). This has been fixed. work_scp sample application problem: When the work_scp.c sample program receives a C-CANCEL-RQ message, it properly returned a C-FIND-RSP message with a cancel status. However, it then sent a second C-FIND-RSP with success when it shouldn't be doing so. The code has been fixed. MC_Free_Message call in samples: In some samples applications the MC_Free_Message call was made too soon. This has been changed. Validation error for STANDARD_XRAY_ANGIO service: The tool kit did not validate the condition for 0x00181520 and 0x00181521 correctly. The tag's presence should be dependent on the positioner motion type (0x00181500) being DYNAMIC. This has been fixed. Possible buffer overflows from Association Response: The tool kit did not handle non-DICOM compliant lengths of the Implementation Class UID or the Implementation Class Name, causing buffer overflows that could crash the tool kit. This has been fixed. Possible buffer overflow when sending data: The tool kit did not handle large values for the LARGE_DATA_STORE parameter correctly. This could cause system crashes. This has been fixed.
2.4.0	21-JUN-99	***Issues when upgrading to Release 2.4.0:*** A new non-fatal error has been added to MC_Wait_For_Association and MC_Open_Association. The return value is MC_NEGOTIATION_ABORTED. SCU and SCP applications should be modified to check for this return value, and continue if it occurs. We changed the ServiceInfo structure returned by MC_Get_First_Acceptable_Service and MC_Get_Next_Acceptable_Service. Previously, there were two members to this structure, scu_role and scp_role that were flags telling if each role were negotiated. These members have been eliminated and replaced by an enumerated value, RoleNegotiated, with the values of "SCU_ROLE, "SCP_ROLE", and "BOTH_ROLES". This will make it easier to determine the roles negotiated. See the Enhancements for 2.4.0 section below for further details. ***Enhancements in Release 2.4.0:*** Updated the Reference Manual, User's Manual, and Sample Applications Guide to reflect changes since release 2.3.0. Made extensive modifications to the storage SCU and SCP sample applications. These modifications include: • the SCU can now read in DICOM Part 10 formatted files and send them over the network • the SCP can write received messages in DICOM Part 10 format and specify the transfer

		syntax in which received images are written • the SCU will now try and identify the transfer syntax of the messages being read in before opening the message and use the appropriate function calls to read it in • a verbose mode has been added to both applications that prints detailed association negotiation information among other options • the SCU can specify on the command line the remote hostname and port number being connected to instead of specifying in the mergecom.app file • the SCP can specify the local listen port and local AE title on the command line Added support for the Modality Performed Procedure Step (MPPS) Service Class to our current worklist management sample applications. These modifications include: • The work_scp.c sample application now requires the file workdata.c to be linked with it. workdata.c contains a software "database" to maintain worklist management and MPPS data. • The work_scp's datafile (work.dat) has been modified and is not compatible with the past versions. The new datafile now contains comments on how to properly format the data contained within the datafile. • The work_scp.c sample application now has two logging levels (-l and -u). These logs will show how the worklist matching is being performed, and also show the contents of key values contained within the work_scp's "database" during MPPS operations. • The work_scu.c sample application has been modified to retrieve a worklist, allow the user to select an item in the worklist, and then create an MPPS instance to the same server. • Currently, the notification and get MPPS service classes are not supported. Note that the documentation in our Sample Applications Guide has not been completed for these applications. The code, however, is well documented. Added support for the following DICOM Supplements: • Supp 15: Visible Light Image Object • Supp 36: Codes and Controlled Terminology • Supp 38: New Print Image Overlay Box • Supp 39: Stored Print Media Storage

		The Visible light supplement included adding support for four new storage services:
		`STANDARD_VL_ENDOSCOPIC: VL Endoscopic Image Storage`
		`STANDARD_VL_MICROSCOPIC: VL Microscopic Image Storage`
		`STANDARD_VL_PHOTOGRAPHIC: VL Photographic Image Storage`
		`STANDARD_VL_SLIDE_MICROSCOPIC: VL Slide-Coordinates Microscopic Image Storage`
		The codes and controlled terminology supplement required modifications to the definitions of multiple code sequence already defined in DICOM. These have been updated accordingly.
		
		The new print Image Overlay box supplement added a new normalized service to the tool kit:
		`BASIC_PRINT_IMAGE_OVERLAY_BOX: Basic Print Image Overlay Box SOP Class`
		The Stored Print Media Storage supplement defined a new directory record for stored print objects.
		
		Updated the med_fsu.c sample application to support the stored print and radiation therapy directory records as defined in DICOM supplements 39 and 29 respectively.
		
		Changed the tool kit's automatic association rejection informational log message into a warning log message.
		
		Modified the MC_Free_File and MC_Free_Message functions such that they both are capable of free'ing both message and file objects. Previously the MC_Free_File function only would free file objects and the MC_Free_Message function would only free message objects. This was the cause of memory leaks in some user's applications.
		
		Made performance improvements to our byte swapping code. These improvements should increase the tool kit's network transfer performance when byte swapping needs to be done.
		
		Added MC_NEGOTIATION_ABORTED return value to MC_Wait_For_Association and MC_Open_Association. This return value handles any case of a connection being aborted during association negotiation. Previously, MC_SYSTEM_ERROR was returned when this type of error occurred.
		
		Changed the ServiceInfo structure returned by MC_Get_First_Acceptable_Service and MC_Get_Next_Acceptable_Service to make it easier to determine the role negotiated for a service. The following are the definitions of these structures:
		
		`/* ========== *`
		`* SCU and SCP Role information *`
		`* ========== */`
		`typedef enum {`
		`SCU_ROLE,`
		`SCP_ROLE,`
		`BOTH_ROLES`
		`} ROLE_TYPE;`
		
		`/* ========== *`

```
 *                      Service Information Structure                       *
 * ========================================================================== *
 *  Calls to MC_Get_First_Acceptable_Service and                              *
 *   MC_Get_Next_Acceptable_Service must pass the address of a variable of    *
 *   this type.  The structure will be filled in by that function.           */
typedef struct MC_Service_Info {
    char            ServiceName[34];        /* MergeCOM-3 Service Name */
    TRANSFER_SYNTAX SyntaxType;             /* Transfer syntax negotiated
                                               for the service */
    ROLE_TYPE       RoleNegotiated;         /* The role negotiated for the
                                               service */
} ServiceInfo;
```

New functions added in Release 2.4.0:

Added new function, MC_Get_Version_String, to retrieve the release number of the tool kit at run time. The prototype for this function is the following:

```
MC_STATUS MC_Get_Version_String(int AbufferSize, char* Abuffer);
```

Added new functions for retrieving our configuration options at run-time. These functions are prototyped as follows:

```
MC_STATUS MC_Get_String_Config_Value( StringParm     Aparm,
                                      int            AbufferSize,
                                      char*          Abuffer);
MC_STATUS MC_Get_Bool_Config_Value(   BoolParm       Aparm,
                                      int*           Avalue);
MC_STATUS MC_Get_Long_Config_Value(   LongParm       Aparm,
                                      long int*      Avalue);
MC_STATUS MC_Get_Int_Config_Value(    IntParm        Aparm,
                                      int*           Avalue);
MC_STATUS MC_Get_Log_Destination(     LogParm        Aparm,
                                      int*           Avalue);
```

The functions operate similar to the MC_Set_xxx_Config_Value functions except they return the configuration options.

Changed the mc3info utility used with private data dictionaries so that it will detect problems with the info.pfl file that may cause run-time errors.

New configuration options added in Release 2.4.0:

Added new configuration option, ALLOW_INVALID_PRIVATE_ATTRIBUTES. When set to Yes, this option will force the tool kit to read in any improperly formatted private elements and write them out again if need be. When set to No, these types of attributes will be ignored. Improperly formatted private elements included the cases where a private creator code has not been included for a tag, or the private creator code itself is formatted improperly.

Bug Fixes in Release 2.4.0:

Fixed problem with reading in messages when using private message information files. Specifically, the problem occurred if a message or file object was created with MC_Open_Message or MC_Create_File and private attributes were defined for the message or file. In this case, the

		tool kit would ignore private attributes being read in that matched the attributes defined for the message. Fixed problem where the MC_Open_File function would go into an infinite loop when reading multiple very large DICOMDIRs. The error would occur when 32,000 directory records (that could be contained in multiple DICOMDIRs) were opened at once. Fixed problem where sequences of items were not properly inheriting the specific character set attribute from parent messages. (Part PS3.5, section 7.5.3 of DICOM describes this.) Because of this problem, we could not assign extended character set values to items for the tag (0008,0005) Specific Character Set was not included in the item. Fixed problem with the MC_Get_Next_pValue_To_ShortInt and MC_Get_Next_pValue_To_UShortInt functions not working properly when retrieving tags with 2-byte VRs. Fixed problems with byte swapping of OL VRs. (Support for the OL VR was added in release 2.3.3). Byte swapping was not always done properly when reading in OL VR tags. Fixed problem with the MC_Library_Reset function where when it was used in an SCP application, MC_Wait_For_Association would no longer function because it could not bind the listen port. Fixed problem with the tool kit's data dictionary where the lossy image compression tag was not included in the definitions of most image objects. Fixed problem with MC_Open_File_Upto_Tag not returning file offsets properly. Previously, when using this function with a stop tag that was contained within the file being read, the offset returned would be for before the stop tag instead of after the stop tag. Fixed minor DICOM problem with how presentation contexts are rejected by SCP applications. When a presentation context was rejected because the transfer syntaxes requested were not supported, a generic reason was given. This was changed to say the real reason the presentation context was rejected.
2.3.3	03-MAR-99	***Enhancements in Release 2.3.3*** Reduced the tool kit's memory management code overhead. This improvement mainly will affect applications that are dealing with very large DICOMDIRs or have other messages in memory with significantly sized header data. Added support for the new VRs: UT and OL. These additions require that the version of the mrgcom3.dct file that ships with this tool kit must be used with this release. (Using older versions of this file with result in application errors. Added the ability to log complete messages being sent or received over the network to our log file.

		Turning on the T2 logging level in the merge.ini file will enable this logging to occur. Note that in normal operation this logging level (and all other trace logging levels) should be turned off because they degrade performance considerably. Modified the PDU logging T9 level so that more information about P-DATA PDUs being sent is logged. Changed MC_List_Message so that the hex value of unprintable characters are now displayed instead of replacing the unprintable character with a '_' character. Added support for the Radiotherapy Treatment Record and Printer Configuration supplements to the tool kit's data dictionary. ***New functions added in Release 2.3.3*** Added function MC_Register_pCallback_Function. This function works identical to the MC_Register_Callback_Function call, except that it can be used for private OB or OW tags. ***Bug Fixes in Release 2.3.3*** Fixed problems where tags read into the tool kit (with the MC_Read_Message, MC_Stream_To_Message, or MC_Open_File_... functions) that were encoded as UN, but had a valid VR in our data dictionary were not converted into the valid VR from UN. Fixed problem where we could not decode private sequence attributes that were encoded as undefined length in an implicit VR transfer syntax. Fixed problem where the tool kit could not parse multi-frame images encoded in an encapsulated transfer syntax in some instances. The problem would occur when the item tag used to delimit frames of pixel data would be split across separate buffers passed to the tool kit's streaming functions. Fixed several typos in the data dictionary. The enumerated value for ISO_IR 144 in tag (0008,0005) was improperly entered. Also, the VR for the tag Time Slot Time (0054,0073) had the vr of US listed when it should have been DS. Fixed problem with pragma's listed in our header files for the Macintosh. Fixed problem with Macintosh tool kit where entering an IP address for in the mergecom.app file would not function properly. Fixed problem with the Macintosh tool kit where the MC_Set_MergeINI function was not working properly, and environment variables were not working properly on the Mac for filenames in the merge.ini and mergecom.pro files. Added new return value to the MC_Open_File...() functions. These functions will now return MC_INVALID_FILE when they read an object in that does not have DICM encoded as the

		DICOM prefix. Also, fixed a problem where these functions would go into an infinite loop if they were trying to decode a message that had an attribute's length encoded as 1 byte. (This error typically would occur when trying to read a file that was not a valid DICOM dataset.)
2.3.2	21-DEC-98	***Bug Fixes in Release 2.3.2*** Fixed library threading problem. The problem resulted in some instances when the MC_Open_Association, MC_Close_Association, or MC_Abort_Association functions were being called at the same time. Calling a combination of these functions simultaneously from multiple threads in some cases would cause corruption of the tool kits internal data structures which in turn cause an assertion failure by the library. The assertion failure resulted in a segmentation fault in the library. The problem affects Windows 95/NT and Solaris tool kit customers. Fixed problem with when MC_Open_File is used in conjunction with a registered callback function. In the case where the entire data is passed to MC_Open_File in a single call, the function will not return an error if the registered callback reports one. This happens when the registered callback is being called with the PROVIDING_DATA option. Fixed a potential problem when the tool kit aborts associations where a socket connection may be left open indefinitely. Previously, if we attempted to close the socket connection when there is still data in the TCP/IP subsystem, the actually socket would not be closed until all data is sent. If the DICOM system on the other end is in a state where it is no longer servicing its TCP/IP connections, the socket connection would be left open indefinitely. The tool kit now resets the connection in this case, so that it is properly cleaned up. Fixed problem in the PrintJob_N_GET function in our prnt_scp.c sample application. We were incorrectly retrieving the affected SOP class UID instead of the requested SOP Class UID from messages. Fixed problem with MC_List_Message and MC_List_File where attributes that had a length of exactly 50 characters only had the first 49 characters displayed by these functions. Fixed problems with the MC_Set_Negotiation_Info and MC_Get_Negotiation_Info functions. For SCP applications that did not support extended negotiation information, if there was extended information in the association request, the tool kit would automatically place in the association response the extended negotiation information that was included in the association request. For SCU applications, If extended negotiation information was included in the association request, but not in the response, the MC_Get_Negotiation_Info function would return the association request information instead of saying nothing was available. These functions now return the appropriate information when applicable. ***New configuration options added in Release 2.3.2*** Added a new configuration option, REMOVE_PADDING_CHARS to the mergecom.pro file. This configuration option removes padding characters (leading or trailing spaces for text based VRs) from attributes that are passed through the tool kit. When this option is set to Yes, attributes

		encoded with the MC_Set_Value...() functions will have these space character removed before they are stored in the tool kit. Also, when messages are read with functions such as MC_Read_Message, MC_Stream_To_Message, and MC_Open_File leading and trailing spaces are removed from attributes before they are stored within the tool kit. (Note that for attributes with a VR of LT or ST, only trailing spaces are removed.)
2.3.1	04-NOV-98	***Enhancements in Release 2.3.1*** Modified the MC_Send_Request_Message(), MC_Write_File(), and MC_Message_To_Stream() functions to improve performance when the pixel data does not have to be byte swapped. We have eliminated an extra memcpy of all of the pixel data in this case which will improve performance for these functions. Added 'const' to the declaration of string variables used in our API that are not modified by the tool kit. Modified MC_Open_Association so that it now returns MC_UNKNOWN_HOST_NAME instead of MC_SYSTEM_ERROR when a host it trying to connect to cannot have its hostname resolved into an IP address. ***Bug Fixes in Release 2.3.1*** Added pragma to our header files for the Borland C++ tool kit so that enumerated values are not treated as bytes. Fixed problems when MC_Set_String_Config_Value() was used to set the log file name. Previously, this option would not work if the log file had already been created. When this function is now called at run time, a new log file will be created based on the name passed to the function. Fixed problem where the tool kit could not handle hostnames passed to the MC_Open_Association() function that were greater than 30 characters. Fixed problem with the MC_Write_File() and MC_Write_File_By_Callback functions where in some cases group lengths were not being calculated properly in conjunction with UN VR attributes. Fixed problem with release 2.3.0 where the role negotiation for acceptor applications did not work intuitively. Acceptor applications were configured for the role they were expecting from the requester, instead of the role that they were taking. Fixed problem with the MS-DOS with PharLap extender tool kit where we could not handle IP address with '0' elements in them such as 128.1.0.1. Fixed problem with Borland tool kit where some of the new functions introduced with release 2.3.0 where not exported from the DLL.

		Removed some T3 level logging added with release 2.3.0. Fixed memory/stack overwrite problem with MC_Open_File() that caused some stack related problems under MacOS. Fixed problem where the MC_Read_Message, MC_Stream_To_Message, and MC_Open_File functions. We could not parse tags whose VRs are encoded as 2 bytes in explicit VR transfer syntaxes, and whose lengths were greater than 32K. Fixed problem with the definitions for the N-EVENT-REPORT-RQ and N-EVENT-REPORT-RSP messages for the Detached management services where incorrect. Fixed problem with the MC_Read_Message, MC_Stream_To_Message and MC_Open_File functions when the UNKNOWN_VR_CODE configuration value was set to OB. In this case, all attributes encountered with the real VR of OB where being changed to UKNOWN_VR.
2.3.0	21-SEP-98	***Issues when upgrading to release 2.3.0*** The following are several issues that need to be dealt with in upgrading to release 2.3.0 of MergeCOM-3. A new parameter has been added to the MC_Reject_Association() function. This parameter is now the reason for the rejection. We have defined several values for this parameter in the mergecom.h include file. Calls to this function must be modified to include this new parameter. The tool kit now ships with a single message information file instead of the multiple files that were used in previous releases of the tool kit. We have also modified the name of the configuration option in the mergecom.pro file that previously pointed to the directory where these files were contained. The format of the mrgcom3.dct file has changed. The file version shipped with this release must be used with the version 2.3.0 tool kits. Versions of this file from older tool kit releases will not work under 2.3.0. ***Enhancements in Release 2.3.0*** Eliminated the use of individual message information files that were contained in the mc3msg directory. These files are now contained in a single file called mrgcom3.msg in the mc3msg directory. These changes required the changing of a configuration option in the mergecom.pro file. The new option is: MSG_INFO_FILE = ../mc3msg/mrgcom3.msg This replaces the previous option, MSG_INFO_DIRECTORY. The tool kit is not backwards compatible with the old file format.

Also changed the format of the mrgcom3.dct file so that it is platform independent. The new mrgcom3.msg file is also platform independent. These files can now be exchanged between platforms. This is most useful for embedded systems so the same dictionary file can be used on both the cross compiler platform and the target platform.

Added support for DICOM Role Negotiation. This functionality lets SCP applications initiate connections. In particular it is used in the storage commitment and detached management service classes for SCPs to send N-EVENT-REPORT-RQ messages to an SCU after an association has been closed. Role negotiation has been implemented in MergeCOM-3 advanced by extending the use of service lists in the mergecom.app file. A role can now be specified for each service in a service list. See the mergecom.app file for details.

Made improvements in how attributes of VR SS, US, AT, SL, UL, FL, and FD are stored internally within the tool kit. The amount of memory used to store tags of these VRs where the value of multiplicity is high has been greatly reduced. These are mainly used in the Nuclear Medicine IOD to store look up tables. In conjunction with these improvements, the MC_Set_Value_From_Function and MC_Get_Value_From_Function calls can now be used to set and retrieve attributes with these value representations.

Added a new parameter to the MC_Reject_Association() call. This parameter allows an application to set the reason for rejecting the association. The reasons available are a subset of the potential DICOM reasons for rejecting an association because MergeCOM-3 automatically handles some of the reasons. The declaration for this function is now:

MC_STATUS EXP_FUNC MC_Reject_Association (int AssociationID,
REJECT_REASON Areason);

The REJECT_REASON enumerated value is defined in mergecom.h and has the following values:

```
/* ========================================================================== *
 * Association Rejection Reasons Enumerated Type *
 * ========================================================================== *
 * Calls to MC_Reject_Association must use this enumerated type to           *
 * specify the reason why the association is being rejected. Note that       *
 * not all of the potential rejection reasons are listed because they are    *
 * automatically handled by the tool kit. */
typedef enum {
    PERMANENT_NO_REASON_GIVEN,
    TRANSIENT_NO_REASON_GIVEN,
    PERMANENT_CALLING_AE_TITLE_NOT_RECOGNIZED,
    TRANSIENT_TEMPORARY_CONGESTION,
    TRANSIENT_LOCAL_LIMIT_EXCEEDED
} REJECT_REASON;
```

Made multiple improvements to the logging mechanism. These include adding a configuration option to adjust the number of columns in the log file. We now log the thread ID for a log under Solaris. Also added simple mechanism so that when different processes log to the same file, the logs in most cases no longer overwrite each other. Finally, added new function MC_Set_Log_Prefix() that will cause a prefix of up to 8 characters to be set before a log statement. This can then be adjusted by a user's application so that it can be determined which portion of code

generates a specific log statement.

Expanded the AssocInfo structure returned by the MC_Get_Association_Info() function. This structure now includes the remote implementation class UID and version name from association negotiation. The structure is declared as follows:

```
/* ========================================================================== *
 * Association Information Structure *
 * ========================================================================== *
 * Calls to MC_Get_Association_Info must pass the address of a variable *
 * of this type. The structure will be filled in by that function. */
typedef struct MC_Assoc_Info {
    int  NumberOfProposedServices;   /* From service list */
    int  NumberOfAcceptableServices; /* Acceptable to both sides */
    char RemoteApplicationTitle[20]; /* 16-characters max */
    char RemoteHostName[40];         /* Network node name */
    int  Tcp_socket;                 /* TCP Socket used for
    char RemoteIPAddress[40];        /* Network IP Address */
    char LocalApplicationTitle[20];  /* 16-characters max */
    char RemoteImplementationClassUID[66];/* 64-characters max */
    char RemoteImplementationVersion[20]; /* 16-characters max */
} AssocInfo;
```

Added support for the letter ballot version of supplement 32, Digital X-ray. The 6 new storage service class objects have been added with the following service names:

```
STANDARD_DX_PRESENT: Digital X-Ray Image Storage - For Presentation
STANDARD_DX_PROCESS: Digital X-Ray Image Storage - For Processing
STANDARD_IO_PRESENT: Digital Intra-oral X-ray Image Storage - For Presentation
STANDARD_IO_PROCESS: Digital Intra-oral X-ray Image Storage - For Processing
STANDARD_MG_PRESENT: Digital Mammography Image Storage - For Presentation
STANDARD_MG_PROCESS: Digital Mammography Image Storage - For Processing
```

Implemented changes to allow the Worklist applications to: Perform wildcard matches, perform more enhanced date searches - that follow the DICOM standard, default the date of queries to today's date, and allow the user to define today's date as a field in the database's data file.

Added a pragma to the tool kit's header files for use with the Macintosh tool kit. This pragma allows the user to not have to set the option for 68k struct alignment in the CodeWarrior development environment.

Added new utility, mc3comp to the tool kit. This utility is the DICOM equivalent to the UNIX 'diff' application. It will list the differences between two DICOM objects. The objects can be encoded in either the media or "stream" format. The usage of the utility is the following:

```
Usage: mc3comp [-t1 -t2 ] [-e efile] [-o -m1 -m2] file1 file2
    -t1 = Optional specify transfer syntax of 'file1' message, where
        = 'il' for implicit little endian (default)
          'el' for explicit little endian
          'eb' for explicit big endian
    -t2 = Optional specify transfer syntax of 'file2' message, where
        = 'il' for implicit little endian (default)
          'el' for explicit little endian
          'eb' for explicit big endian
    -e  = Optional exception file of all tags to ignore in comparison
    -o  = Compare OB/OW (e.g. binary pixel) data
    -m1 = Compare 'file1' in DICOM-3 file format.
```

```
    -m2 = Compare 'file2' in DICOM-3 file format.
    file1 = DICOM SOP Instance (message) file
    file2 = another DICOM SOP Instance (message) file
    -h = get help - print this usage description
Example: mc3comp -t1 eb -m2 ct1.img ct2.img
```

Added a new utility, mc3conv. This utility can be us to convert the transfer syntax of an object or it can be used to convert from DICOM media format to DICOM "stream" format. The syntax for using the utility is the following:

```
mc3conv input_file output_file [-t ] [-m] [-tag <"newvalue"]
    input_file = DICOM SOP Instance (message) file
    output_file = Output DICOM SOP Instance (message) file
    -t   = Specify transfer syntax for 'output_file', where
         = 'il' for implicit little endian (default)
           'el' for explicit little endian
           'eb' for explicit big endian
    -m   = Specify format of 'output_file' to be DICOM-3 format.
    -tag = Change value for this tag in 'output_file', where
         = the tag that is to be changed in hex 0x...
         = the value for the tag in quotes ""
    -h   = get help - print this usage description
Example: mc3conv ct1.img file.001 -t eb -m
```

Modified the tool kit utilities (mc3list, mc3valid, and mc3comp) so that they now attempt to determine the transfer syntax an object is encoded in before reading it in. In most cases, the -m and -t options no longer have to be used to specify if an object is encoded in the media format or a transfer syntax other than implicit little endian.

Made general improvements to how pixel data is handled when reading in a DICOM file or reading over the network. The pixel data is now more efficiently passed to callback functions registered with MC_Register_Callback_Function() and stored internally.

Changed logging level from warning to T5 when we encounter an attribute with invalid characters when streaming in a message or file.

Modified the maximum number of characters for an attribute's name from 50 to 64 characters.

New functions added in Release 2.3.0

Added two new functions defined in the mc3msg.h include file that are related to logging:

```
void EXP_FUNC MC_Set_Log_Prefix(char *Aprefix);
void EXP_FUNC MC_Register_MemoryLog_Function(
    void (*AmemLogFunction)
          (MsgLogType Arg1,
           char* Arg2));
```

The MC_Set_Log_Prefix() function allows the user to set a prefix that is printed before the beginning of log states in the merge.log file. This can be used to determine the part of code that generates a given log statement. Note that in threaded environments this function sets the value for an entire process and not a particular thread.

Added function MC_Register_MemoryLog_Function. A pointer to a function is passed to this function. When a log occurs within the MergeCOM-3 library, this registered function is called and

passed the log statement. Note that the destinations for a log level must be set to "Memory" within the merge.ini file for this to work.

Added new function MC_Get_UID_From_MergeCOM_Service() that will find the DICOM SOP Class UID that is associated with a MergeCOM-3 Service name. The prototype for this function is the following:

```
MC_STATUS MC_Get_UID_From_MergeCOM_Service( char* AserviceName,
    char* Auid,
    int AbufferSize );
```

Added two new functions to convert between the MergeCOM-3 enumerated values used to represent a transfer syntax and the DICOM UID used to represent those transfer syntaxes. The prototypes of the functions are as follows:

```
MC_STATUS MC_Get_Transfer_Syntax_From_Enum( TRANSFER_SYNTAX Asyntax,
    char* Auid,
    int AbufferSize );
MC_STATUS MC_Get_Enum_From_Transfer_Syntax( char* Auid,
    TRANSFER_SYNTAX* Asyntax );
```

Added new function, MC_Set_Message_Callbacks to the tool kit. This function is used to associate a callback function registered with the MC_Register_Callback_Function() to be associated with a tag in a message. This would then allow the user to use the MC_Get_Value_Length() or MC_Get_Value_To_Function and MC_Set_Value_From_Function calls to manipulate tags whose values are contained in callback functions. Previously, this association was not done until a message was sent or received over the network, or read or written to media.

New configuration options added in Release 2.3.0

Added the ability to support up to 2 private transfer syntaxes within the tool kit. The private transfer syntaxes are now configured through the following options in the mergecom.pro file:

```
[ASSOC_PARMS]
PRIVATE_SYNTAX_1_SYNTAX =             # UID for transfer Syntax 1
PRIVATE_SYNTAX_1_LITTLE_ENDIAN = Yes # If Yes, syntax 1 is little endian,
                                      # if No, syntax is big endian
PRIVATE_SYNTAX_1_EXPLICIT_VR = Yes    # If Yes, syntax 1 is explicit VR
PRIVATE_SYNTAX_1_ENCAPSULATED = No    # If Yes, transfer syntax is encapsulated
                                      # and encapsulated rules must be followed
                                      # for pixel data (7fe0,0010).
PRIVATE_SYNTAX_2_SYNTAX =             # UID for transfer Syntax 2
PRIVATE_SYNTAX_2_LITTLE_ENDIAN = Yes # If Yes, syntax 2 is little endian,
                                      # if No, syntax is big endian
PRIVATE_SYNTAX_2_EXPLICIT_VR = Yes    # If Yes, syntax 2 is explicit VR
PRIVATE_SYNTAX_2_ENCAPSULATED = No    # If Yes, transfer syntax is encapsulated
                                      # and encapsulated rules must be followed
                                      # for pixel data (7fe0,0010).
```

These options allow the user to set the endian of a private transfer syntax, whether the syntax is explicit or implicit VR, and whether the pixel data within the transfer syntax follows the encapsulated pixel data encoding method used in the DICOM JPEG and RLE transfer syntaxes. There are also options for these transfer syntaxes in the TRANSFER_SYNTAX enumerated value used for various tool kit functions.

		Added four new configuration options to the mergecom.pro file to control exporting of private attributes and attributes with a VR of UN:
		`[MEDIA_PARMS]`
		`  EXPORT_UN_VR_TO_MEDIA = Yes`
		`  EXPORT_PRIVATE_ATTRIBUTES_TO_MEDIA = Yes`
		`[MESSAGE_PARMS]`
		`  EXPORT_UN_VR_TO_NETWORK = Yes`
		`  EXPORT_PRIVATE_ATTRIBUTES_TO_NETWORK = Yes`
		The EXPORT_UN_VR_TO_MEDIA and EXPORT_PRIVATE_ATTRIBUTES_TO_MEDIA options eliminate the exporting of attributes with unknown VR (UN) or all private attributes when using the MC_Write_File() and MC_Write_File_By_Callback() functions.
		
		The EXPORT_UN_VR_TO_NETWORK and EXPORT_PRIVATE_ATTRIBUTES_TO_NETWORK options eliminate the exporting of attributes with unknown VR (UN) or all private attributes when using the MC_Send_Request_Message() and MC_Send_Response_Message() functions.
		
		Added LOG_FILE_LINE_LENGTH configuration value to the merge.ini file. This configuration value dictates the number of columns in the tool kit's log file. It can be set to any value between 16 and 254.
		
		Bug Fixes in Release 2.3.0
		
		Fixed problem with MC_Get_Value_To_String where if the function returned MC_BUFFER_TOO_SMALL, it would place a NULL character one position in memory after the length of the string passed to the function.
		
		Added option to log a warning message when a configuration option is found in the tool kit's configuration files that is not used by the tool kit. We found several options in the configuration files that were not in use. These have been removed from the shipping tool kit's configuration files and may produce errors if your configuration files are not updated..
		
		Fixed problem with the Macintosh tool kit where the CONNECT_TIMEOUT configuration option was not used by the tool kit. This option now works properly when the MC_Open_Association() function is called.
		
		Fixed problem with the Macintosh tool kit where we would not properly handle when multiple connections were initiated at the same time. Error messages would be logged that the accept() function failed.
		
		Fixed problem with the worklist SCP sample application where it would not send a C-FIND-RSP message with a status of CANCELED when it received a C-CANCEL-RQ message.
		
		Under Windows 95, 98, and NT, changed the include file for Winsock to winsock2.h instead of winsock.h.

		Fixed problem with inadvertent error messages being generated when the ACCEPT_ANY_PRESENTATION_CONTEXT configuration option was set to "Yes". The tool kit would log that a presentation context was incorrectly encoded when sending messages over Meta SOP classes such as DICOM print. Fixed problem with the AUTO_ECHO_SUPPORT configuration option not working properly. Fixed problem where the tool kit's listen port was not properly released when an application exits. This would previously cause an application that was starting later to not be able to bind the listen port. This was mainly a problem under Solaris. Fixed problem where we would not look up in the data dictionary attributes within private groups that were in the curve or overlay groups. Changed behavior of the tool kit when we encounter private creator codes with the same name within a single private group. Previously the private attributes associated with the later creator codes would be moved into the first creator code. Now these attributes are just ignored. Modified the function MC_Add_Standard_Attribute() so that it can now be used to add group length attributes for private groups. Previous these could only be dealt with through the MC_Add_Nonstandard_Attribute function call. Fixed problem with MC_Set_Value_From_Double where it would not encode attributes of VR IS properly. It would actually produce a decimal string instead of an integer string. Fixed problems with the defined terms for tag (0008,0005) in our data dictionary. We did not include the defined terms for the Japanese extended character sets in this tag. Fixed problem in the query/retrieve SCU where it did not handle if the retrieve AE title returned in a C-FIND-RSP message was set to NULL. This is meant to indicate the object is off line. Also fixed a problem where the local AE title for this application could not be set. Fixed problem with MC_Write_File where if any of the group 2 elements in the file object being written were deleted with MC_Delete_Attribute, MC_Write_File() would fail with MC_SYSTEM_ERROR. Fixed problem with the DIR_REC_CURVE directory record where the tags Curve Number and LUT Number were listed as the wrong value type which would cause this directory record to be validated improperly. Fixed problem with implementation of DICOM print store where the message return status for the constant: N_ACTION_WARNING_IMAGE_BOX_LUT_NOT_SUPP was listed as 0xB606. It has been changed the the correct value of 0xB608. Fixed problem with the name of the tag Film Box Content Sequence. The name was misspelled

		causing the constant for this tag in diction.h to be misspelled. Fixed problem with the BLOCK_SEQUENCE item contained in the RT Plan IOD. The tag Block Transmission (300a,0102) was not included in this sequence and should have been. Fixed problem with the MC_Set_pValue_From_...() and MC_Get_pValue_To_...() functions. These functions would modify the private creator code variables passed to them if these variables had a space at the end of the string. They will no longer do this modification. Fixed problem that occurred when large buffers were passed to the tool kit with the MC_Set_Value_From_Function(). When the LARGE_DATA_STORE configuration option was set to FILE, we would allocate large buffers to manage the data when we shouldn't have. Fixed problem with Borland C++ based Windows NT tool kit where multithreading did not work correctly. The multithreading required semaphores were not included when this tool kit was built. Fixed problem with callback functions registered with MC_Register_Callback_Function(). These function's return values were not checked when they were called with the option PROVIDING_DATA_LENGTH.
2.2.5	18-MAY-98	***Enhancements in Release 2.2.5*** Added new function, MC_Set_Message_Callbacks to the tool kit. This function is used to associate a callback function registered with the MC_Register_Callback_Function() to be associated with a tag in a message. This would then allow the user to use the MC_Get_Value_Length() or MC_Get_Value_To_Function and MC_Set_Value_From_Function calls to manipulate tags whose values are contained in callback functions. Previously, this association was not done until a message was sent or received over the network, or read or written to media. Added support for DICOM Role Negotiation. This functionality lets SCP applications initiate connections. In particular it is used in the storage commitment and detached management service classes for SCPs to send N-EVENT-REPORT-RQ messages to an SCU after an association has been closed. Role negotiation has been implemented in MergeCOM-3 advanced by extending the use of service lists in the mergecom.app file. A role can now be specified for each service in a service list. See the mergecom.app file for details. ***Bug Fixes in Release 2.2.5*** Fixed problem with MC_Write_File where if any of the group 2 elements in the file object being written were deleted with MC_Delete_Attribute, MC_Write_File() would fail with MC_SYSTEM_ERROR. Fixed problem with the DIR_REC_CURVE directory record where the tags Curve Number and LUT Number were listed as the wrong value type which would cause this directory record to be validated improperly. Fixed problem with implementation of DICOM print store where the message return status for the

		constant: N_ACTION_WARNING_IMAGE_BOX_LUT_NOT_SUPP was listed as 0xB606. It has been changed the correct value of 0xB608. Fixed problem with the name of the tag Film Box Content Sequence. The name was misspelled causing the constant for this tag in diction.h to be misspelled. Fixed problem with the BLOCK_SEQUENCE item contained in the RT Plan IOD. The tag Block Transmission (300a,0102) was not included in this sequence and should have been. Fixed problem with the MC_Set_pValue_From_...() and MC_Get_pValue_To_...() functions. These functions would modify the private creator code variables passed to them if these variables had a space at the end of the string. They will no longer do this modification. Fixed problem that occurred when large buffers were passed to the tool kit with the MC_Set_Value_From_Function(). When the LARGE_DATA_STORE configuration option was set to FILE, we would allocate large buffers to manage the data when we shouldn't have. Fixed problem with Borland C++ based Windows NT tool kit where multithreading did not work correctly. The multithreading required semaphores were not included when this tool kit was built. Fixed problem with callback functions registered with MC_Register_Callback_Function(). These function's return values were not checked when they were called with the option PROVIDING_DATA_LENGTH.
Release 2.2.4	14-APR-98	***Enhancements in Release 2.2.4*** Modified the way callbacks registered with the MC_Register_Callback_Function() function work. Now, when functions such as MC_Get_Value_To_Function() and MC_Set_Value_From_Function() are called, if a callback is registered it will be ask for or supplied with the pixel data. Added support for the tag (0040,0020) to the tool kit. This tag was added in change proposal 72 and is used in the Modality Worklist service class. Added support to the data dictionary for Supplement 24, Stored Print Related SOP Classes and Supplement 22, Presentation Look Up Table (LUT). Made some general improvements in the med_fsu sample application. Also changed so that it better reflected the conformance statement in the sample applications guide. Added the return value MC_CONNECTION_FAILED to MC_Open_Association function. This value is returned instead of MC_SYSTEM_ERROR when the tool kit cannot connect to a remote system. It will allow for better diagnosis of this type of situation. ***New configuration options added in Release 2.2.4*** Added CALLBACK_MIN_DATA_SIZE configuration option to the MESSAGE_PARMS section

		of the mergecom.pro file. This option sets the minimum size of a DICOM element for which a callback function registered with the MC_Register_Callback_Function() is registered. This option can be used in several fashions. It may be used to manage storing pixel data for only very large images, or it can be used to set a value where ICON pixel data may be ignored. Added GROUP_2_TRANSFER_SYNTAX configuration option to the MEDIA_PARMS section of the mergecom.pro file. Note well that changing this value from the default, "EXPLICIT_LITTLE_ENDIAN" will cause the tool kit to not conform to the DICOM standard. This option was implemented so that the tool kit can deal with an early draft of the DICOM media format where the group 2 elements were encoded in the implicit little endian transfer syntax. In the final draft of the DICOM standard, the group 2 elements are encoded in the explicit little endian transfer syntax. ***Bug Fixes in Release 2.2.4*** Fixed problem with support for extended character sets. We previously did not handle when the '\' character was included in an extended character string. Fixed problem under Windows NT/95, where if an IP address was entered as the HOST_NAME in the mergecom.app file, and the IP address could not be resolved to a hostname on the local system, the tool kit would cause an exception fault. Fixed problem where the tool kit could not decode a sequence where the sequence is defined as fixed length and the items within it were encoded as undefined length. The DICOMDIR on a recent HeartLab demonstration CD were encoded in this format. Fixed problem with MC_Open_File_Bypass_OBOW() where the byte offset to pixel data attributes contained within a sequence were not given properly to a registered callback function. Fixed the names of multiple tags that were entered incorrectly. This will in turn cause the constants with the diction.h file to change. Note that 2 tags have been changed that will most likely cause problems. The new names for these tags are MC_ATT_TRANSFER_SYNTAX_UID and MC_ATT_BASIC_GRAYSCALE_IMAGE_SEQUENCE. These new names will have to be used in media and print applications, respectively. Fixed problem during association negotiation where the incorrect reason was given when an abstract syntax was supported, but the transfer syntaxes proposed were not supported. Fixed problem with MC_Write_File() where when a very large DICOMDIR file was written, it would cause a segmentation fault to occur when the file was being free'd with MC_Free_File(). Fixed problem where the configuration option, ACCEPT_MULTIPLE_PRES_CONTEXT, was misspelled in the mergecom.pro configuration file. Fixed several coding errors in the print SCP sample application. These included how COL format was parsed in the image display format, how the number of rows and columns were extracted from

		the image display format, and the fact that the pixel aspect ratio element was required to be present by the application when it shouldn't have been. Fixed problem under OpenVMS where logical names could not be used for the pathname configuration options in the mergecom.pro file.
2.2.3	6-FEB-98	***Enhancements in Release 2.2.3*** Added new function MC_Set_MergeINI() which will allow setting the path and name of the merge.ini file before initialization of the library. When this function is used, the library will not look at the MERGE_INI environment variable. Added the Modality Performed Procedure Step supplement to the tool kit. Information on this service class can be found in the message.txt file. Added configuration option WORK_BUFFER_SIZE to the mergecom.pro file. It is the sizeof working buffer used when streaming in or out messages/files. Setting this option to a size larger than 28K may cause memory fragmentation problems with the library and is not recommended. Changed MC_Set_Value_From_String() so that when the string "" is passed to the function, the value being set is set to NULL. Added configuration value HARD_CLOSE_TCPIP_CONNECTION to the mergecom.pro file. When this configuration option is set to NO, a FIN packet is sent when an TCP/IP connection is close. If it is set to Yes, a RST packet is sent. ***Bug Fixes in Release 2.2.3*** Fixed problem with MC_Add_Standard_Attribute() under Solaris where the library would go into an infinite loop when a curve or overlay tag was added to a message. This only occurs when threads are being used. Fixed problem with DICOMDIRs that would cause the library to cause a segmentation fault in MC_Free_File() on occasion when very large DICOMDIRs were being manipulated (> 5MB). Fixed problem with the library where A-RELEASE-RSP PDUs were not logged properly when PDU logging was turned on. Also fixed problem in the library where the ARTIM timer was not used by SCP applications when an A-RELEASE-RSP PDU was sent. Instead, a 1 second timeout was always used. Fixed PDU sizing problem in conjunction with registered callback functions used for pixel data. (MC_Register_Callback()). The tool kit would send PDUs in the size chunk that was being supplied to the tool kit by the user's callback instead of sizing them according to the value negotiated for maximum PDU size. Fixed problem in the modality worklist SCP sample application where C-CANCEL-RQ messages were not received properly if they were read after the last C-FIND-RSP message was being sent.

		These messages are now ignored. Fixed problem under OS/9 and OS/9000 where an incorrect end of line character was being used in the log file.
2.2.2	13-NOV-97	***Enhancements in Release 2.2.2*** Added support to the MC_Set_Value_From_String() call for multi-byte character sets. This function will work for tags with the VRs PN, SH, LO, ST, and LT. Note that for the time being we had to turn off the validation of the length of values being assigned with this function. No validation is currently done to ensure that these values have been encoded properly according to what is set in the Specific Character Set (0008,0005) attribute. Added two new configuration options to the mergecom.pro file that allow the user to set the TCP/IP send and receive buffer sizes. These configuration options are TCPIP_RECEIVE_BUFFER_SIZE and TCPIP_SEND_BUFFER_SIZE. Setting higher values for these constants improve the TCP/IP performance of most systems. Note that high values for these constants can have a negative impact on performance on networks with heavy traffic. These values should be set slightly larger than the PDU_MAXIMUM_LENGTH configuration value. Added new configuration option ALLOW_INVALID_CREATOR_CODES. Previously, when messages were streamed in with MC_Read_Message or MC_Stream_To_Message, if a private creator code was contained in the message that had illegal characters, the entire block of private attributes associated with the creator were ignored. Now, when this configuration option is set to Yes, these private attributes will be read into the library. ***Bug Fixes in Release 2.2.2*** Fixed problem with the MC_Get_Value_To_LongInt function retrieving values from tags with VR DS. When the decimal string value was encoded with scientific notation, we would not properly translate it into an integer. Fixed problem with the MC_Open_File_Bypass_OBOW() call. This function was improperly bypassing both the pixel data in a file, and sequence attributes contained in the file objects when reading them in. Added Modality Worklist sample applications to the distribution. This sample application may still have to go through another review and may change before final release. Fixed problem with the MC_Get_Value_To_...() functions where they would return MC_EMPTY_VALUE instead of MC_INVALID_TAG when messages were opened with the MC_Open_Empty_Message() function. Fixed problem under OpenVMS where OpenVMS style pathnames placed in our configuration files would not work properly. Fixed performance problem seen when sending from Windows NT based systems to HP-UX based

		systems both using the Advanced Tool Kit. Performance was increased by a factor of 6 with the addition of the TCPIP_SEND_BUFFER_SIZE and TCP_RECEIVE_BUFFER_SIZE parameters and setting these values slightly larger than the PDU_MAXIMUM_LENGTH configuration value. Fixed problem with how the tool kit was rejecting associations when an invalid protocol version was used. The reasons for the rejection were not properly being set. Also fixed a problem where we were not checking the protocol version returned to an SCU from the SCP. Fixed problem introduced in version 2.2.1 where the command group elements were being added to empty messages (although they did not have values set). These re now added only when a message is validated with MC_Validate_Message(). Fixed problem introduced in version 2.2.1 where a series of error log messages were displayed on an SCU when an association was aborted in the middle of a send operation. Added the ICON Image Sequence to both C-FIND-RQ and C-FIND-RSP messages for all of the Query/Retrieve service classes.
2.2.1	22-SEP-97	***Enhancements in Release 2.2.1*** Changed the declaration of the the input string parameter in the MC_Set_Value_From_String function to const char* from char*. This was done such that the functions are compatible with the CString type used in Windows 95/NT programming. Cleaned up and revised the Query/Retrieve sample applications. Fixed several problems with fields not being set properly and problems when message validation was turned on. In particular, the interface to the Q/R SCU has been modified such that it more closely matches the interface that would be needed for a GUI application. Changed the way an association is rejected when a host connects where the hostname cannot be determined. Previously, the connection was dropped. Now, an informational message is logged, and the association is rejected. Improved performance of DICOMDIR related functions. Previously there was a degradation of performance for the MC_Dir_... functions as the size of a DICOMDIR increased. This degradation has been greatly reduced. Reduced the tool kit memory overhead to store message header information. This improvement is especially noticeable when dealing with very large DICOMDIRs. Added support for the final text version of the print queue management supplement. The service name is PRINT_QUEUE_MANAGEMENT. One problem was encountered during this implementation. The new tag "(2120,0070) Referenced Print Job Sequenece" is already a valid DICOM tag (2100,0500). We have used tag (2100,0500) within our implementation, and intend on asking DICOM Working Group 6 to do the same.

		Implemented the Positron Emission Tomography (PET) supplement. These include the standard PET (STANDARD_PET) and PET curve (STANDARD_PET_CURVE) message types. Added support for the final text version of the DICOM RT IODs. These include the RT Dose, RT Image, RT Structure Set, and RT Plan IODs. The names for these services are STANDARD_RT_DOSE, STANDARD_RT_IMAGE, STANDARD_RT_STRUCTURE_SET, and STANDARD_RT_PLAN respectively. Added new return value to the MC_Stream_To_Message(), MC_Stream_To_Message_With_Offset(), and MC_Open_File_..() functions called MC_OUT_OF_ORDER_TAG. This value is returned when tags in the data being parsed are not in ascending order. Implemented supplement 14 to the DICOM standard, "Unknown Value Representation". In conjunction with this addition, added the configuration option, "UNKNOWN_VR_CODE" to the "[MESSAGE_PARMS]" section of the mergecom.pro configuration file. This option allows for the changing of the unknown vr code from "UN" to any other value. This was implemented for the case of interoperability problems when an implementation does not yet support "UN", this value may then be changed to OB. Modified how the mergecom.app file is used so that IP address can be placed in the file instead of complete host names. This will not work under Windows 95 and the Macintosh. Windows NT 4.0 already allowed IP addresses to be entered. Modified the PDU_MAXIMUM_LENGTH configuration value so that it can be set to a minimum value of 4096. This was done because values less than this may cause association negotiation to fail. ***Bug Fixes in Release 2.2.1*** Fixed problems with how PDUs are sized when sending messages. Previously, very small PDUs (~12 bytes) were being sent by the tool kit. The PDU_MAXIMUM_LENGTH configuration value (found in mergecom.pro) now has an impact on how PDUs being sent by the tool kit are sized. (This value was previously only used during association negotiation to tell remote systems the maximum sized PDU that the tool kit can receive). In addition, if this value is set smaller than the maximum PDU length received by the tool kit during association negotiation, PDUs will be formed that are the size of the value set to PDU_MAXIMUM_LENGTH configuration option. This method was used in order to control the memory usage by the tool kit. Fixed problem with our implementation of Modality Worklist. We did not include the C-CANCEL-RQ message as a valid message when it should have been included. Fixed problem in release 2.2.0 where C_CANCEL_RQ message could not be properly sent in some situations. Fixed problem with how Curve and Overlay attributes that were not part of the base groups

		(0x5000,0x6000) were handled. When a message was created with MC_Open_Empty_Message(), these attributes could not be added to the message, the encoding functions would return MC_INVALID_TAG. Fixed problem when using registered callback functions with the MC_Send_Request_Message() call to supply pixel data. When the pixel data was contained in a sequence (such as in the print service class), the length of the sequence was not encoded properly. Fixed small memory leak with the MC_Register_Callback() function occurs when using it with callbacks for tags contained in sequences. Fixed small memory leak with the MC_Set_Value_To_NULL() function that occurs when setting tags that already have a value, or that have been emptied by a call to MC_Empty_Message() or MC_Empty_File(). Fixed problem with associations being rejected. If more than one association is rejected within a single call to MC_Wait_For_Association(), several warning messages are were being displayed. Fixed problem with MC_Set_Int_Config_Value(). When an illegal negative value was passed to this function, the value was still set, even though an error was returned. This value is no longer set. Removed the STANDARD_PH service. This was a very early implementation of the Visible light DICOM supplement. Fixed a problem with mc3file where is wasn't filling the modality field correctly for some message types. Fixed problem with MC_Validate_Message where the tool kit would cause a segmentation fault when validating a message that contained a NULL length sequence. Fixed problem with C-FIND-RQ message types for the 3 query/retrieve services where the retrieve response module was being included in request messages when it shouldn't have been. Fixed previous problems with the MC_Set_pValue_Representation() call. This function previously did not work properly with multi-valued attributes. It now works in conjunction with the support of unknown VR attributes and will properly decode any attributes (includes sequences of items). When a message was opened with MC_Open_Empty_Message() and then validated with the MC_Validate_Message() call, command level attributes (group 0) could not be added to the message and would return the value MC_INVALID_TAG from the MC_Set_Value_From_...() calls. The image date & time tags used in all objects that include the general image module were not listed with a conditional as they should have been according to the DICOM standard. Fixed a problem with how N-GET-RQ / RSP messages were handled by the print sample

		applications. These applications were not properly using the "attribute identifier list" tags to determine what tags should be requested by the SCU and returned by the SCP. Also did some minor clean up. Fixed a problem with the med_fsu sample application with how the group two elements were being added to the DICOMDIR file and image files. These tags were not being properly set. Also did some other minor cleanup to the application.

2.2.0	05-MAY-97	***Enhancements in Release 2.2.0***
		
		Added enhanced support for JPEG and other encapsulated transfer syntaxes. The main addition in this area is the use of "Transfer Syntax Lists" within the mergecom.app file. This format allows you to specify the DICOM transfer syntaxes supported for a given MergeCOM-3 Service. A detailed description of the format of these "Transfer Syntax Lists" is contained in the mergecom.app configuration file. Note: the new format is backwards compatible with the previous file format.
		
		The tool kit will also now allow for more than one presentation context per service (abstract syntax). It is now possible to negotiate two transfer syntaxes for one service, and send messages via either transfer syntax. Two new functions, MC_Set_Message_Transfer_Syntax() and MC_Get_Message_Transfer_Syntax() have been added to support this functionality. They are described in detail below.
		
		On the Macintosh, added a pragma to turn 'enums always int' on directly in our header files. This option no longer needs to be turned on within the CodeWarrior project.
		
		When T3 message logging is turned on to log association negotiation, the MergeCOM-3 service name is now listed for each abstract syntax negotiated. Also, the transfer syntax name is also listed with the transfer syntax UID.
		
		When T6 logging is turned on, the version of the tool kit being used is now logged.
		
		When T7 message logging is turned on to log the command group elements contained in messages sent over the network, the presentation context that the message was sent over is now logged.
		
		Changed the genconf/gendict utility programs so that the source files that they generate will only work with a specific version of the tool kit. In the future, any new versions of the tool kit will report an error at initialization that an incorrect version was used. However, if you use a pre-2.2.0 tool kit, an exception error/segmentation fault will occur at initialization time.
		
		Changed the AssocInfo structure populated by the MC_Get_Association_Info() function call. The remote IP address, and local application title being connect to were added to this structure. The following is the prototype for this structure:
		
		`typedef struct MC_Assoc_Info {`
		`  int  NumberOfProposedServices;  /* From service list */`
		`  int  NumberOfAcceptableServices; /* Acceptable to both sides */`
		`  char RemoteApplicationTitle[20]; /* 16-characters max */`
		`  char RemoteHostName[40];  /* Network node name */`

```
    int  Tcp_socket;                /* TCP Socket used for association */
    char RemoteIPAddress[40];       /* Network IP Address */
    char LocalApplicationTitle[20]; /* 16-characters max */
} AssocInfo;
```

The LocalApplicationTitle field was added for applications where the ACCEPT_ANY_APPLICATION_TITLE is turned on. This will allow the user to determine what AE a remote user is attempting to connect to.

New configuration options added in Release 2.2.0

Added a new "inactivity timeout" to the tool kit. This is used in conjunction with the MC_Read_Message() function call (and currently only affects this call). The timeout parameter contained in the MC_Read_Message() call now specifies the number of seconds to wait for the first data to come across the network. The INACTIVITY_TIMEOUT parameter contained in the mergecom.pro configuration file specifies to the amount of time to wait inbetween packets of data coming across the network (after the first as been received). When an inactivity timeout occurs, the tool kit will now return the value MC_INACTIVITY_TIMEOUT, and the network connection will have been dropped.

Added "ACCEPT_ANY_HOSTNAME" configuration value to the [ASSOC_PARMS] section of the mergecom.pro configuration file. When this option is turned on, the tool kit will no longer check if applications connecting to an SCP have their hostname listed in the system's host file.

Added ACCEPT_MULTIPLE_PRES_CONTEXTS configuration value to the [ASSOC_PARMS] section of the mergecom.pro configuration file. This option turns off the new feature that allows more than one presentation context to be negotiated per service.

Added ACCEPT_ANY_PRESENTATION_CONTEXT configuration value to the [ASSOC_PARMS] section of the mergecom.pro file. This function disables checking by the tool kit if the presentation context ID contained in P-DATA PDUs matches the presentation context negotiated for type of message contained in the PDU. Some early implementations of the standard did not properly fill in the presentation context. This option was added to address interoperability problems in this case.

Added FORCE_OPEN_EMPTY_ITEM configuration value to the [MESSAGE_PARMS] section of the mergecom.pro file. When this option is turned on, the MC_Open_Item() function will act similar to the MC_Open_Empty_Message() function. When turned on, this function will also force performance improvements when dealing with DICOMDIR files. This call has no effect on embedded platforms.

Added EMPTY_PRIVATE_CREATOR_CODES configuration value to the [MESSAGE_PARMS] section of the mergecom.pro file. When this option is turned off, private creator codes contained in messages are not emptied when the MC_Empty_Message() or MC_Empty_File() calls are made. Therefore; when populating a message the user is not required to add the private creator codes again.

New functions added in Release 2.2.0

Two new functions, MC_Get_Message_Transfer_Syntax() and MC_Set_Message_Transfer_Syntax have been added in compliment the improved support of encapsulated transfer syntaxes. These functions are only of use when multiple presentation contexts have been negotiated for the same MergeCOM service. The following are the prototypes for these functions:

```
MC_STATUS EXP_FUNC MC_Set_Message_Transfer_Syntax(
    int AmsgID,
    TRANSFER_SYNTAX Asyntax);
MC_STATUS EXP_FUNC MC_Get_Message_Transfer_Syntax(
    int AmsgID,
    TRANSFER_SYNTAX* Asyntax);
```

The MC_Get_Message_Transfer_Syntax() only works on messages that have been received over the network with MC_Read_Message(). It allows you to get the transfer syntax in which a message was received over the network. The MC_Set_Message_Transfer_Syntax() would be called on a message before it is sent over the network. It is used to set the presentation context over which you want to send a message. (For example, the case where an SCU has a standard transfer syntax in one presentation context, and a JPEG transfer syntax in another presentation context. The SCP accepts both presentation contexts, and the SCU can dictate when it sends a message if the image is compressed or uncompressed.) Note: these functions need not be used when sending response messages. The tool kit will send a response message over the presentation context that the last request message was sent over.

Added the function MC_Validate_Attribute(). This function works similar to MC_Validate_Message()/MC_Validate_File() except that it only validates a single attribute contained in a message. This function does not have the overhead of the MC_Validate_Message() function. It only loads validation info for the single tag being validated instead of for every tag in the message. The prototype for this function is:

```
extern MC_STATUS EXP_FUNC MC_Validate_Attribute (int AmsgID,
    unsigned long Atag,
    VAL_ERR** AerrInfo,
    VAL_LEVEL AerrLevel );
```

Added new function: MC_Write_File_By_Callback(). The following is the prototype for the function:

```
extern MC_STATUS EXP_FUNC MC_Write_File_By_Callback(int AapplicationID,
   int AfileID,
   int AnumBytes,
   void* AuserInfo,
   MC_STATUS (NOEXP_FUNC *AuserFunction)
      (char* CBfilename,
       void* CBuserInfo,
       int CBdataSize,
       void* CBdataBuffer,
       int CBisFirst,
       int CBisLast));
```

This function was introduced to fix a bug with MC_Write_File(). Because MC_Write_File was not passed the application ID, it was unable to use registered callback functions (with

MC_Register_Callback()) to supply pixel data when writing to media. When there are no callbacks registered for AapplicationID, this function will perform identical to MC_Write_File(), if a callback is registered, and the pixel data in the message already does not have a value assigned, this function will call the callback to retrieve the pixel data in the message.

Added the function MC_Stream_To_Message_With_Offset(). The following is the prototype for this function:

```
extern MC_STATUS EXP_FUNC MC_Stream_To_Message_With_Offset(
    int AmsgID,
    unsigned long AstartTag,
    unsigned long AstopTag,
    TRANSFER_SYNTAX Asyntax,
    unsigned long* AerrorTag,
    unsigned long* Aoffset,
    void* AuserInfo,
    MC_STATUS (NOEXP_FUNC *AuserFunction)
        (int CBmsgID,
         void* CBuserInfo,
         int CBfirstCall,
         int* CBdataLen,
         void** CBdataBuffer,
         int* CBisLast));
```

This function works identical to MC_Stream_To_Message(), except that a new parameter has been added to the function, Aoffset. This parameter returns the byte offset within the DICOM stream to the next tag in the message after AstopTag. If AstopTag is the last tag (or after the last tag) in the message, the length of the complete stream is returned.

Added MC_Byte_Swap_OBOW() function. The following is the prototype for the function:

extern MC_STATUS EXP_FUNC MC_Byte_Swap_OBOW (int AmsgID,
unsigned long Atag);

This function was developed to be used in conjunction with 8-bit pixel data. DICOM requires that in most cases, 8-bit pixel data be treated as OW (unless an explicit VR transfer syntax is used). As defined by DICOM standard change proposal number 14 (which has been approved by the DICOM standards committee), 8-bit pixel data on big endian machines is byte swapped. (The pixels are not in byte by byte ascending order, they are swapped.) This function was added to byte swap OB or OW pixel data to get it in the correct format before it is sent over the network. This function should only be needed on BIG ENDIAN machines.

Added MC_Library_Release() function. The prototype for this function is:

```
extern MC_STATUS EXP_FUNC MC_Library_Release (void);
```

This function releases all resource allocated by the tool kit. This function was added to be used in conjunction with tools that allow for checking of memory leaks within an application.

Added the function MC_Get_MergeCOM_Service. This funtion takes a SOP Class UID as an input parameter, and returns the MergeCOM-3 service name for this UID. The prototype for the function is the following:

```
extern MC_STATUS EXP_FUNC MC_Get_MergeCOM_Service(char* Auid,
    char* Aname,
    int Alength);
```

A call to this function has been added to the stor_scu.c and qr_scp.c sample applications to simplify them.

Added two new functions, MC_Get_Tag_Info() and MC_Get_pTag_Info(). These functions return a text string describing a given DICOM tag. The prototype for these functions is the following:

```
extern MC_STATUS EXP_FUNC MC_Get_Tag_Info (unsigned long Atag,
    char* Aname,
    int Alength);
extern MC_STATUS EXP_FUNC MC_Get_pTag_Info (char* AprivateCode,
    unsigned short Agroup,
    unsigned char Aelem,
    char* Aname,
    int Alength);
```

These functions would be of use for displaying names of tags in a log file or in a user interface, etc.

Bug Fixes in Release 2.2.0

Fixed several problems with the attributes contained in directory records within DICOMDIRs. The DIR_REC_IMAGE item was missing the Icon Image sequence (0088,0200) which has been added. Curve Number had a value type of 1 instead of 3 within DIR_REC_IMAGE. Image number had a value type of 3 instead of 1 in the DIR_REC_IMAGE item. Also, the DIR_REC_IMAGE and DIR_REC_SERIES items did not contain all of the image IE module attributes and series IE module attributes respectively. Note that all of the attributes added with this last problem were type 3 (not required).

Fixed problem introduced with Version 2.10 where an exception fault would occur in the MC_Wait_For_Association() call if an association is aborted during association negotiation.

Fixed problem with PDU logging where Abort PDUs were not being properly logged in some cases.

Fixed problem with the mc3file utility so that the -t to set the transfer syntax for the object generated works properly for DICOM media format files generated.

Fixed problem with validating C_MOVE_RQ messages that caused image/series/study keys to incorrectly be flagged as missing.

Fixed problem where sockets were potentially not being closed properly. In some cases, the first 3 sockets opened by the tool kit were not closed properly.

Fixed problem with sending encapsulated transfer syntax images. In some cases, a padding character was incorrectly being added to the pixel data.

Fixed problem with the Q/R SCP sample application where C-MOVE-RSP messages were not

		being sent after the move operation has completed. Fixed problem with all NULL text values that are encoded such as '\\\\\'. The MC_Message_To_Stream() function was not properly encoding these values.
2.12	21-JAN-97	Fixed problem with MC_Open_Association() call where when a connection failed, we were not freeing the socket properly. Fixed problem with the MC_Dir_Open_MRDR() function where the offset to the next directory record and the offset to the lower level directory entity were not being filled in, as they are supposed to be according to the DICOM standard. Fixed problem with the MC_List_Item()/MC_List_File()/MC_List_Message() functions not working with DLLs under Windows NT/95 and OS/2. These function calls were changed only on these platforms such that instead of the second parameter being of type FILE*, it is now a filename to place the listing in. (A value of NULL still sets the output to stdout.) Fixed problem with the listing output on the Macintosh for MC_List_Item/ MC_List_File()/MC_List_Message(). The output previously had the improper end of line terminator for the Macintosh, so that when these listings were opened in a Macintosh editor, the end of lines were not read properly.
2.11	13-DEC-96	Fixed data dictionary problem with Printer Name (2110,0030). We had hte VR as SH, when it should have been LO. Fixed build problem with release 2.10 for Windows NT/95 where the threading fixes implemented in this release were not included.
2.10	02-DEC-96	Modified the stor_scu sample application so that it checks the status return value in the C_STORE_RSP. This status value may tell that the C-STORE operation failed. We were not checking this value in the past, and should be checked by any storage service class SCU application. Added the N_CREATE_MEMORY_ALLOCATION_NOT_SUPPORTED return value for N-CREATE-RSP messages to the mergecom.h include file. We had omitted having this value in our include file. Added two new configuration parameters dealing with logging. The first, NUM_HISTORICAL_LOG_FILES is used in conjunction with the current LOG_FILE_BACKUP parameter. When LOG_FILE_BACKUP is turned on, NUM_HISTORICAL_LOG_FILES specifies the total number of log files to save. This parameter is placed in the MergeCOM3 section of the merge.ini configuration file. The second parameter, BLANK_FILL_LOG_FILE instructs the tool kit if it should blank fill the log file at initialization time. If this option is turned on, the log file will be expanded to the maximum size allowed at startup. If it is turned off, the log file will not be exanded and initialization time will be reduced. However, when logging occurs, the performance

		will be slower because the file must be expanded to include the new logs. A new configuration option, ELIMINATE_ITEM_REFERENCES, was added to the MESSAGE_PARMS section of the mergecom.pro file. This option is for performance tuning of the tool kit. When set to YES, if the MC_Free_Message, MC_Free_File, and MC_Free_Item functions encounter a item in the message they are freeing, they will search all other message, file, or item objects for references to the item it is freeing and remove these references. When set to NO, it does not perform this operation, thus improving performance. Previously, this was always set on and was a safeguard against incorrect references in objects. Added two new trace levels of logging that can be set in the merge.ini file. T7 will list the command group elements of all messages sent or received by the tool kit. T9 will log all PDUs that are sent or received by the tool kit. Note that for P-DATA PDUs, only the length of the PDU is logged, and not the actual complete PDU. Improved the performance of all message manipulation functions when multiple message/item/file objects are kept open at once. Fixed problems under preemptively threaded environments (OS/2, Windows NT/95, and Solaris 2.4). Intermittent failures were occurring when threaded SCP applications were heavily loaded. Media applications that are parsing file objects simultaneously in multiple threads could also be impacted. (Ie, MC_Open_File() is called from multiple threads at the same time.) Fixed problem with MC_Open_File where if a DICOMDIR had a circular reference, the tool kit would go into an infinite loop. (IE, a lower level directory record referenced an entity above it.) Added HARD_CLOSE_TCP_IP_CONNECTION configuration option to the mergecom.pro file. This option allows for the changing of how associations are closed at the TCP/IP level. When turned on (the default), a hard close is initiated sending a RST packet over TCP/IP. When turned on, a FIN packet is sent and we wait for a response over the connection. This may solve some problems with not catching associations closing. Fixed problem with the MC_Get_Negotiation_Info call where extended negotiation information was being returned for the wrong service. In character string attributes that require space padding to make them even length, the space is now stripped off by the tool kit when they are received. (IE, the value returned by MC_Get_Value_To_String will not have the padding space character.) This will eliminate some comparsion problems. Note that some value representations allow for leading spaces. We do not do anything about these values. We also leave attributes that are all spaces alone. Changed MC_Read_Message, MC_Open_File, and MC_Stream_To_Message so that if a message is being read that contains an invalid private creator code, instead of aborting the message, we will log a warning and continue on. Note that all attributes in the private group will also be ignored. The number of proposed services returned by the MC_Get_Association_Info function is not

		correct when called from an SCP application. For SCPs, the value is the number of services in its service list, and not the number proposed. If the echo service class is automatically supported, previously the number also would be 1 more than the number of services in the SCP's service list. Added change so that the error message "Unable to create log file" that is printed to stderr is only printed once to the screen instead of each time a log is attempted. This error was printed any time we are unable to create a log file. A problem with the prnt_scp sample application was fixed where if the print job service was not negotiated, the application would still try and send PRINT_JOB, N_EVENT_REPORT_RQ messages across the association. Fixed problem with the stor_scp and prnt_scu sample applications. The service name being returned by the MC_Read_Message() call was being used after the memory it pointed to was free'd by the MC_Free_Message() function call. This caused a problem with premptively threaded environments. Fixed problem with the MC_Get_Value_To_String()/MC_Get_Next_Value_To_String() functions getting NULL attributes. In the case of multi-valued attributes with one of the values NULL, if a either of these functions were called with a buffer of length 0 or 1 and the attribute they were trying to get was NULL, they would return MC_BUFFER_TOO_SMALL instead of MC_NULL_VALUE. If the VR of pixel data is set to OB, and an implicit transfer syntax is negotiated, the DICOM standard states the attribute should be of type OW. In the case of the Implicit Little Endian transfer syntax, 8 bit OW and OB data are encoded in the same fashion. If, however, an application is non-conformant, and has defined a UID for Implicit Big Endian transfer syntax, if the pixel data attribute's VR is set to OB, the data will be byte swapping incorrectly. This became a potential problem when the MC_Set_Value_Representation() was modified in release 2.07.
2.09	24-SEP-96	Note: This is a limited release for RMX-3 for Philips ONLY. Fixed DICOM related bug in the qr_scu sample application. The application was programmed to function properly with the qr_scp, but not with standard DICOM implementations. When a single C-MOVE-RQ was done for a series of images, the qr_scu would not wait until all of the C-STORE-RQ's are received that were requested to be moved. It would abort the association after the first C-STORE-RQ was received. Fixed problem introduced in Version 2.07 where the MC_List_Message() function caused a general protection fault when listing NULL length sequences. Eliminated problems on RMX-3 systems that were caused by having floating point instructions in the library.
2.08	22-AUG-96	Added support for the placement of environment variables within our configuration files in place

		of any options that contain a filename. Environment variable should occur in the following format: MERGECOM_3_PROFILE = $(ENVIRONMENT_VARIABLE)/mergecom.pro Only 1 environment variable is allowed, and the environment variable must be the first part of the pathname. In the case of MacOS tool kits which do not have environment variables, we have made two special cases to get the system's system folder and preferences folder. By placing $(PREFERENCES_FOLDER) or $(SYSTEM_FOLDER) in a configuration file, the respective pathnames will be placed in the configuration file. Cleaned up the prnt_scu and prnt_scp sample applications. A Print Service Class sample application guide is now shipping with the tool kit. Also added support for the print job service class to both the scu & scp. Added support for the storage commitment service class. Added some additional logging for diagnosing network problems. Added a new configuration option to check for the protocol verison on association negotiation. The option is "ACCEPT_ANY_PROTOCOL_VERSION" and is located in the [ASSOC_PARMS] section of the mergecom.pro file. The default value is FALSE. Fixed a problem with the Macintosh tool kit where the date in the log file was 1 day off. Added a configuration option that is a timeout for how long to wait when trying to connect to a remote host. This option impacts the MC_Open_Association() function call. On most systems, the timeout will only occur when trying to connect to a system that is not on the network, or is not operational. The "CONNECT_TIMEOUT" option is in the [DUL_PARMS] section of the mergecom.pro file. The default setting is 15 seconds. Made a fix to the embedded RMX3 tool kit so that IP addresses can be placed directly in the configuration files. Modifed the utilities so that the utility version matches with the tool kit version. Cleaned up some compilation warnings under Windows NT/95.

2.07	26-JUL-96	Fixed a problem introduced in Release 2.06 with mc3file. The utility would always generate 1 image fine, but when generating multiple images it would fail.
		
		Added the MC_Set_Next_Value_To_NULL() & MC_Set_Next_pValue_To_NULL() function calls to the tool kit. The API for these calls is:
		`MC_STATUS MC_Set_Next_Value_To_NULL(int MsgItemID,`
		`unsigned long tag);`
		`MC_STATUS MC_Set_Next_pValue_To_NULL(int MsgItemID,`
		`char* PrivateCode,`
		`unsigned short group,`
		`unsigned char ElementByte);`

		These two functions only work with text attributes that use '\' as a delimiter between multiple values. It can be used with attributes such as Image Type (0008,0008) where the 3rd value may want to be NULL in some situations. Note well that in cases where the last value in a multi-valued attribute is set to NULL, a space character may be added to make the attribute an even length (causing the value to no longer be "NULL"). Fixed problem with C-ECHO-RSP messages automatically sent by the tool kit. This message was always filling in the field "Responding Message ID" as having a value of 1 instead of the "Message ID" placed in the C-ECHO-RQ message. Fixed problem with mc3echo where when multiple echos were being sent, the message ID was not being incremented properly. Fixed problem in the Windows 95 / NT tool kit where the prnt_scu & prnt_scp applications were not compiling correctly. Fixed another small memory leak within the tool kit. This one only happens in applications that use registered callbacks with the MC_Register_Callback_Function(). This leak will happen in several types of applications. For every 2 times the MC_Send_Request_Message, MC_Read_Message(), MC_Write_File(), or MC_Open_File_..() functions are called, approximately 100 bytes are lost. It only occurs when these functions are called when a callback is registered with the MC_Register_Callback_Function(). Fixed problem where if MC_Stream_To_Message( ) was streaming in a message, and ran out of memory, the tool kit was not returning the MC_SYSTEM_ERROR return value. Made a fix where AE titles that had the same base name could be mistakenly be thought equal. IE, the AE titles MERGE_STORE and MERGE_STORE_SCP were registered with MC_Register_Application() in a server application. An application connecting to MERGE_STORE is returned from the MC_Wait_For_Association() call as connected to MERGE_STORE_SCP. Fixed incorrect logging statement in MC_Read_Message() encountered when VRs for an attribute are changed. Modified MC_Set_Value_Representation() and MC_Set_pValue_Representation() such that OB/OW/UNKNOWN_VR attributes can have their VR changed. Previously, only attributes with the VR UNKNOWN_VR could be changed. This change was made available to combat a problem encountered with encapsulated transfer syntaxes. This problem occurs when a image is read in (over a network or from a file) in an encapsulated transfer syntax. The pixel data within the file is converted from an undefined length into a defined length. When the message was then streamed out, the tool kit assumed the attribute was of VR OB, when in reality it was changed to OW with the conversion. In this case, the user must now use the function MC_Set_Value_Representation() when the pixel data is changed from compressed to uncompressed.

		Fixed a bug that caused the MC_Set_Negotiation_Info() & MC_Get_Negotiation_Info() functions to not work properly. Fixed problem with encapsulated transfer syntaxes where when they were parsed, all OB / OW attributes encountered were assumed to be compressed & undefined length. This especially was causing problems with private attributes or curve / overlay data. Fixed bug in the med_fsu sample application where the File Meta Information Version was not being set properly.
2.06	18-JUN-96	Fixed a small memory leak in the tool kit. After every eight associations, approximately 300 bytes were being leaked. Added a new return value to the MC_Wait_For_Assocation() function call. This function will now return MC_UNKNOWN_HOST_CONNECTED instead of MC_SYSTEM_ERROR when an unknown host attempts to connect. Be careful with how you handle this return value in your SCP applications! The sample SCP applications had to be changed to handle this return properly. This is not an error that should shut down an SCP (hence why we added the new return value). Added functionality to mc3file utility to support 8-bit images. Also, modified mc3file so it will generate valid XRAY, NM & US images. Added functionality to the mc3file, mc3valid, and mc3list utilities to support file objects. File objects can be created, validated and listed using the "-m" option. Fixed problem with mc3valid where it would not properly validate the STANDARD_US & STANDARD_US_MF objects. Also added support for the retired Ultrasound objets, STANDARD_US_RETIRED & STANDARD_US_MF_RETIRED. Made performance enhancements to the library. The performance of all streaming functions has been improved where byte swapping of pixel data was required. (IE, transfer syntaxes that are not in the machine's native endian.) Also, reduced the amount of memory malloc'd and freed in standard operation. Fixed problem with the VR of the attribute Lossy Image Compression (0028,2110). The VR of the attribute was changed from US to CS. This attribute is used in the XRAY & US image objects. Fixed several problems with the RT objects implemented previously. The RT Image and RT Plan IODs had the wrong SOP Class UID in the mergecom.srv file. Also upgraded the objects to reflect the version of the standard to be used at the AAPM demo (dated April 26, 1996). Fixed problem with MC_Message_To_File() & MC_File_To_Message() where it would fail with MC_TAG_ALREADY_EXISTS if either the group 2 or group 0 elements exist in the message/file before the call is made. Fixed problem with mc3file utility introduced with release 2.05 where the pixel data attribute was

		never added to the first message created by mc3file. Fixed an overflow problem encountered with message IDs. The problem occured when a message object is kept around, and then 31,000 new message objects were created before the first one was free'd. Fixed problem encountered with private attributes being streamed in when a private data dictionary is used. The tool kit was unable to look up the data dictionary info in some cases when multiple private blocks were contained in the same group. Fixed a memory leak in the qr_scu.c sample application. The C-FIND-SUCCESS response message for a C-FIND-RQ was not being free'd with MC_Free_Message(). Also fixed problem with qr_scp.c, where a buffer containing the service list name was not long enough. Memory was being overwritten and the application was not running correctly. Fixed problem with MC_Message_To_Stream() / MC_Write_File() / MC_Send_Request_Message(). The problem occured only when the same message/ file was streamed out two or more times with these functions. The condition also only occured in the case where the output transfer syntax was in a different endian from the platform's endian. The pixel data was being swapped internally, such that the second time a call was made, the pixel data was being swapped again into an invalid byte ordering. Modified the extended tool kit utility mc3dict. This utility was changed so that when an attribute's VM is set to N, it is treated as 1-N. This will change how the Nuclear Medicine object tests some of the offset vector attribute's VM in standard tool kits.

2.05	07-MAY-96	Made improvements to the mc3echo utility's command line interface.
		
		Fixed problem with logging where even if logging was turned off, log strings were being formatted. This should improve performance within the MC_Library_Initialization() call.
		
		Changed the behaving of streaming in of Private blocks. Previously, as private creator codes were being streamed in, they were assigned in ascending order within the group. Now they are streamed in and placed in the tags that they were assigned within the stream.
		
		Fixed encoding problem with the MC_Write_File() function. When streaming out in big endian transfer syntaxes, the sequence delimiter tags were always being encoded in little endian format.
		
		Added support for encapsulated transfer syntaxes. The following constants were added to the mergecom.pro file within the ASSOC_PARMS section:
		`# Encapsulated transfer syntax UIDs`
		`# RLE_SYNTAX = 1.2.840.10008.1.2.5`
		`# JPEG_BASELINE_SYNTAX = 1.2.840.10008.1.2.4.50`
		`# JPEG_EXTENDED_2_4_SYNTAX = 1.2.840.10008.1.2.4.51`
		`# JPEG_EXTENDED_3_5_SYNTAX = 1.2.840.10008.1.2.4.52`
		`# JPEG_SPEC_NON_HIER_3_5_SYNTAX = 1.2.840.10008.1.2.4.53`
		`# JPEG_SPEC_NON_HIER_6_7_SYNTAX = 1.2.840.10008.1.2.4.54`
		`# JPEG_FULL_PROG_NON_HIER_10_12_SYNTAX = 1.2.840.10008.1.2.4.55`

		`# JPEG_FULL_PROG_NON_HIER_11_13_SYNTAX = 1.2.840.10008.1.2.4.56`
		`# JPEG_LOSSLESS_NON_HIER_14_SYNTAX = 1.2.840.10008.1.2.4.57`
		`# JPEG_LOSSLESS_NON_HIER_15_SYNTAX = 1.2.840.10008.1.2.4.58`
		`# JPEG_EXTENDED_HIER_16_18_SYNTAX = 1.2.840.10008.1.2.4.59`
		`# JPEG_EXTENDED_HIER_17_19_SYNTAX = 1.2.840.10008.1.2.4.60`
		`# JPEG_SPEC_HIER_20_22_SYNTAX = 1.2.840.10008.1.2.4.61`
		`# JPEG_SPEC_HIER_21_23_SYNTAX = 1.2.840.10008.1.2.4.62`
		`# JPEG_FULL_PROG_HIER_24_26_SYNTAX = 1.2.840.10008.1.2.4.63`
		`# JPEG_FULL_PROG_HIER_25_27_SYNTAX = 1.2.840.10008.1.2.4.64`
		`# JPEG_LOSSLESS_HIER_28_SYNTAX = 1.2.840.10008.1.2.4.65`
		`# JPEG_LOSSLESS_HIER_29_SYNTAX = 1.2.840.10008.1.2.4.66`
		`# JPEG_LOSSLESS_HIER_14_SYNTAX = 1.2.840.10008.1.2.4.70`
		
		In the default distribution, these are commented out. As these are uncommented, the tool kit will automatically negotiate for the services. If any of these encapsulated transfer syntaxes are defined, the toolkit will select these syntaxes, if possible over a non-encapsulated syntax when performing association negotiation. Note that in the case of media applications, the transfer syntaxes have to be defined in this file in order for an message of that syntax to be read in. We intend on changing the way transfer syntaxes are negotiated in a future release so that the syntaxes can all be defined within the mergecom.pro file without having them be negotiated for..
		
		The TRANSFER_SYNTAX enumerated type within the mc3msg.h file has also been changed to reflect the support of these new transfer syntaxes. Note also the MC_Set_String_Config_Value() can be used to set these transfer syntaxes from within your application.
		
		The tool kit DOES NOT encode JPEG data for the user. The pixel data is handled as previously within the tool kit. The user is responsible for encoding the pixel data properly. When the MC_Set_Value_From_Function(), or the MC_Register_Callback() function is used to assign pixel data within a message, the user must encode the basic offset table, each of the pixel fragments within the pixel data attribute, and then pass it to the tool kit. Just placing the pixel data itself into the pixel data attribute (7fe0,0010) when an encapsulated transfer syntax is negotiated will cause your implementation to break. In the same manner, when retrieving pixel data from a message within the tool kit via the MC_Get_Value_To_Function() or MC_Register_Callback() functions, the user must decode the basic offset table and pixel data fragments in order to access the actual pixel data. Part 5 of the DICOM standard describes how to encode/decode encapsulated pixel data properly.

2.04	10-APR-96	Fixed two bugs in MC_Get_Value_To_Buffer( ) which is used to retrieve values from attributes with unknown VR's (e.g. private attributes in implicit VR DICOM messages). If the value was NULL (had zero length) the length was not properly returned as 0 but left floating at some uninitialized value. Also, if the return buffer size was too small the proper return code (MC_BUFFER_TOO_SMALL) was returned but the actual length of the value was not returned. Both of these bugs have been fixed. Another subtle defect was fixed where MC_Stream_To_Message could core dump if a range of attributes (not the entire Stream) was being read in in an Explicit VR Transfer Syntax. Added the function MC_Get_File_Length(). This function is the equivalent of MC_Get_Stream_Length(), but for file objects. The prototype for the function is as follows:

		`extern MC_STATUS MC_Get_File_Length(int AfileID, unsigned long* AfileLength);` The function uses the transfer syntax UID attribute of the file to determine which transfer syntax to calculate the length for. Also, be careful because the MC_Write_File() function can add trailing padding to the file when it is written to media. This could cause the length to change. Added the 3 new Radiation Therapy objects. These are the RT Plan, RT Structure, and RT Image objects. This implementation is based on the Supplement 11, dated 13 March 1996. Some additional changes were made based on comments from the RT Ad-hoc Working Group Added a subset of the Endoscopy object for use with a Endo demo. (This only includes the modules already in the standard and no Endoscopy specific modules.) The service name is STANDARD_PH. Changed the MC_Set_Value_From_String() function so it does not complain when an attribute of type SH, CS, DS, IS, or LO is encoded with all spaces (and no other characters). Fixed bug with validation of LT value representation attributes. The bug caused the toolkit to complain with the value MC_INVALID_CHARS_IN_VALUE when the control characters LF, FF, and CR were in the attribute. This affected when the attribute was being set with the MC_Set_Value_From_String function, or when the message was validated with one of the MC_Validate_... functions.
2.03	28-FEB-96	Fixed a problem with group length attributes for private groups. When streaming in a file over the network or from media, private group length attributes encoutered within the stream were placed in the message with a random tag instead of the actual tag. Changed the MC_Open_File that when an error is encountered streaming in a DICOMDIR, the call calculates the group length offsets and creates the internal DICOMDIR structures. This previously wasn't the case with DICOMDIRs when an error was encountered. Previously, the MC_Dir_... API calls would not work on this type of DICOMDIR. Made further performance enhancements in the management of message/file objects.
2.02	17-FEB-96	Further performance enhancements were made in the management of message/file objects; particularly in the area of freeing large message/file objects.
2.01	29-DEC-95	Fixed 2 bugs concerning message validation. Attributes with enumerated values and with a range of values of multiplicity (ie, VM's equal to 3-4), would not validate properly. Also, when validated, sequence attributes of value type 2 or 3 that were set to NULL would sometimes give errors for missing required attributes contained within the item when they shouldn't have.
2.00	26-DEC-95	Initial version of Release 2.X of the tool kit including DICOM media functionality.