



Merge DICOM Toolkit™ V. 5.8.0

PLATFORM NOTES

Merge Healthcare
900 Walnut Ridge Drive
Hartland, WI 53029
USA

© Copyright 2018 Merge Healthcare Incorporated, an IBM Company.

The content of this document is confidential information of Merge Healthcare Incorporated and its use and disclosure is subject to the terms of the agreement pursuant to which you obtained the software that accompanies the documentation.

Merge Healthcare® is a registered trademark of Merge Healthcare Inc.

The Merge Healthcare logo is a trademark of Merge Healthcare Inc.

All other names are trademarks or registered trademarks of their respective companies.

DICOM is a registered trademark of National Electrical Manufacturers Association (NEMA). *Merge DICOM Toolkit™* is a trademark of Merge Healthcare. The names of other products mentioned in this document may be the trademarks or registered trademarks of their respective companies.

The Merge DICOM Toolkit includes an evaluation version of Accusoft's AIMTools for Android and iOS SDKs. It is the client's responsibility to license a full version of the AIMTools for the Android or iOS SDK before deploying products built using the AIMTools. Please contact Accusoft via www.Accusoft.com or info@accusoft.com to obtain licensing for the full version of the AIMTools for the Android or iOS SDK. AIMTools is a trademark of Accusoft Corporation.

U.S. GOVERNMENT RESTRICTED RIGHTS:

This product is a “Commercial Item” offered with “Restricted Rights.” The Government’s rights to use, modify, reproduce, release, perform, display or disclose this documentation are subject to the restrictions set forth in Federal Acquisition Regulation (“FAR”) 12.211 and 12.212 for civilian agencies and in DFARS 227.7202-3 for military agencies. Contractor is Merge Healthcare.

The symbols glossary is provided electronically at <http://www.merge.com/Support/Resources.aspx>.



Manufacturer's Address

Merge Healthcare Incorporated
900 Walnut Ridge Drive
Hartland, WI 53029

For assistance, please contact Merge Healthcare Customer Support:

- In North America, call toll free 1-800-668-7990, then select option 2
- International, call Merge Healthcare (in Canada) +1-905-672-7990, then select option 2
- Email MDTsupport@merge.com or MC3Support@ca.ibm.com

Part	Date	Revision	Description
COM-3441	December 2018	1.0	Updated bi-annually

Contents

OVERVIEW.....	8
Document Roadmap	8
Document Conventions	9
Summary of Supported Platforms	9
32-BIT MICROSOFT WINDOWS USING MICROSOFT VISUAL C++ (008-91204).....	10
Supported Configurations.....	10
The Merge DICOM Toolkit Libraries	10
Miscellaneous Notes	11
Threading Support	11
Debug Version	11
Compression Support	12
Unicode Support	12
API Changes for Windows	12
Sample Applications	13
Installation	13
The Merge DICOM Toolkit Requirements	13
Post-Installation	14
Files	14
64-BIT MICROSOFT WINDOWS USING MICROSOFT VISUAL C++ (008-91208).....	20
Supported Configurations.....	20
The Merge DICOM Toolkit Libraries	21
Miscellaneous Notes	22
Debug Version	22
Sample Applications	22
API Changes for Windows	22
Threading Support	23
Compression Support	23
Unicode Support	23
Installation	23
The Merge DICOM Toolkit Requirements	24

Post-Installation	24
Files	25
32-BIT MICROSOFT WINDOWS USING BORLAND® C++ (008-91205)	30
Supported Configurations.....	30
The Merge DICOM Toolkit Libraries	31
Miscellaneous Notes	32
API Changes for Windows	32
Threading Support	33
Compression Support	33
Unicode Support	33
Installation	34
Merge DICOM Toolkit Requirements	34
Post-Installation	34
Files	35
32-BIT LINUX® ON INTEL® X86 (008-91126)	40
Supported Configurations.....	40
The Merge DICOM Toolkit Libraries	40
Miscellaneous Notes	42
Threading Support	42
Compression Support	42
Unicode Support	42
Files	43
64-BIT LINUX® ON INTEL® X86-64 (008-91132)	48
Supported Configurations.....	48
The Merge DICOM Toolkit Libraries	49
Miscellaneous Notes	50
Threading Support	50
Compression Support	51
Unicode Support	51
Files	52
32-BIT SOLARIS™ 10 ON INTEL® X86 (008-91117)	57

Supported Configurations.....	57
The Merge DICOM Toolkit Libraries	57
Miscellaneous Notes	58
Threading Support	58
Compression Support	59
Unicode Support	59
Files	60
64-BIT SOLARIS™ 10 ON INTEL® X64 (008-91116)	65
Supported Configurations.....	65
The Merge DICOM Toolkit Libraries	65
Miscellaneous Notes	66
Threading Support	66
Compression Support	66
Unicode Support	66
Files	67
32-BIT SOLARIS™ 8 ON SPARC® USING SUN™ COMPILER (008-91119) ...	71
Supported Configurations.....	71
The Merge DICOM Toolkit Libraries	72
Miscellaneous Notes	73
Threading Support	73
Compression Support	73
Unicode Support	73
Files	74
32-BIT SOLARIS™ 8 ON SPARC® USING GCC COMPILER (008-91130)	79
Supported Configurations.....	79
The Merge DICOM Toolkit Libraries	79
Miscellaneous Notes	81
Threading Support	81
Compression Support	81
Unicode Support	81
Files	82

32-BIT MAC OS® X USING UNIVERSAL BINARIES (008-91301)	86
Supported Configurations.....	86
The Merge DICOM Toolkit Libraries	87
Miscellaneous Notes	88
Threading Support	88
Compression Support	88
Unicode Support	88
Files	89
64-BIT MAC OS® X ON INTEL® X64 (008-91303)	93
Supported Configurations.....	93
The Merge DICOM Toolkit Libraries	93
Miscellaneous Notes	94
Threading Support	94
Compression Support	94
Unicode Support	94
Files	95
32-BIT ANDROID ON ARMV7-A (008-91998)	99
Supported Configurations.....	99
The Merge DICOM Toolkit Libraries	100
Miscellaneous Notes	101
Threading Support	101
Compression Support	101
Unicode Support	101
Files	102
64-BIT ANDROID ON ARMV8-A (89-00161-00)	107
Supported Configurations.....	107
The Merge DICOM Toolkit Libraries	107
Miscellaneous Notes	108
Threading Support	108
Compression Support	108
Unicode Support	108

Files	109
64-BIT IOS ON ARMV8-A (008-91990)	113
Supported Configurations.....	113
The Merge DICOM Toolkit Libraries	114
Miscellaneous Notes	114
Threading Support	114
Compression Support	114
Files	114
32-BIT LINUX ON ARMV7-A (008-92001) – RETIRED.....	118
64-BIT LINUX ON ARMV8A (89-00163-00)	119
Supported Configurations.....	119
The Merge DICOM Toolkit Libraries	119
Miscellaneous Notes	120
Threading Support	120
Compression Support	120
Files	120

Overview

The Merge DICOM Toolkit™ provides DICOM functionality through Merge's Applications Programming Interface to various operating systems. While the Merge DICOM Toolkit provides the same consistent API, it takes advantage of the capabilities in each operating system. Refer to the appropriate chapter in this guide for your implementation.

Document Roadmap

The Merge DICOM Toolkit documentation is structured as shown in *Figure 1*.

Platform specific information required to use the toolkit on your target platform is specified in Platform Notes. This includes supported compilers, compiler options, link options, configuration, and run-time related issues.

Read me first!

The *User's Manual* is the foundation for all other documentation because it explains the concepts of DICOM and the toolkit. Before plunging into the *Reference Manual* or *Sample Application Guide* you should be comfortable with the material in the *User Manual*.

The *Reference Manual* contains detailed information on the toolkit. This includes the Application Programming Interface (API), tool kit configuration, the runtime object database, and status logging. The *Reference Manual* also includes guidelines for writing a DICOM conformance statement for an application.

The DICOM *Database Manual* is an optional extension that describes the organization of the Merge DICOM Toolkit Database and how to use it to extend standard services and define your own private services. Tools are supplied for converting the contents of the database into the binary runtime object database.

Sample Application

The *Sample Application Guide* describes approaches to developing specific classes of DICOM applications (Image Transfer, Query/Retrieve, Print, Media, Modality Worklist, etc.). The *Sample Application Guide* presents the pertinent information from Parts 3 or 4 of the DICOM Standard in a more readable way and in the context of the toolkit. The *Sample Application Guide* also details the DICOM messages that can be passed between applications on the network. In addition, the sample applications that are supplied for your platform are described within the *Sample Application Guide*.

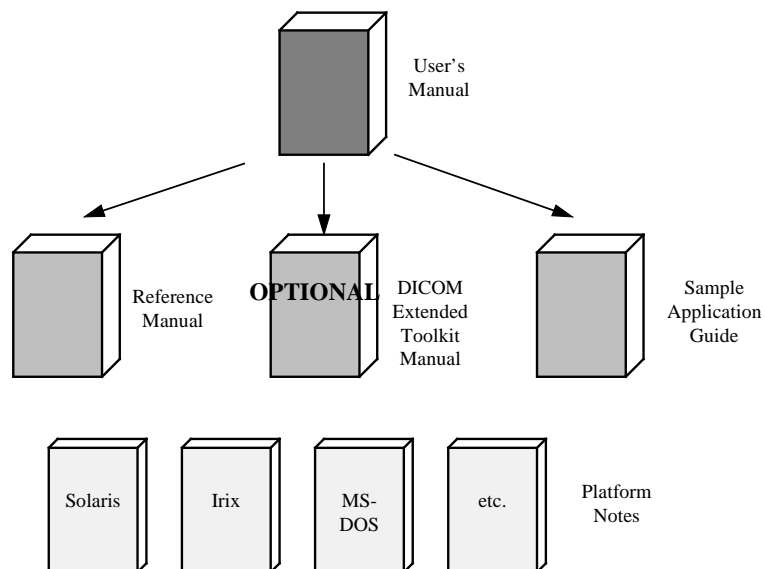


Figure 1: Merge DICOM Toolkit Documentation Roadmap

Document Conventions

This document follows a few formatting conventions:

Terms that are being defined are presented in **boldface**.

Sample Margin Note

Margin notes (in the left margin) are used to highlight important points or sections of the document.

Sample commands, sample output, source code, and function calls appear in `courier new` font.

Hexadecimal numbers are written with a trailing H. For example 16 decimal is equivalent to 10H hexadecimal.

Note: Notes are used to indicate information which may be helpful or of special interest to the reader.

Summary of Supported Platforms

- 32-Bit Microsoft Windows using Microsoft Visual C++ (008-91204)
- 64-Bit Microsoft Windows using Microsoft Visual C++ (008-91208)
- 32-Bit Microsoft Windows using Borland® C++ (008-91205)
- 32-Bit Linux® on Intel® x86 (008-91126)
- 64-Bit Fedora Core Linux® on Intel® x86-64 (008-91132)
- 32-Bit Solaris™ 10 on Intel® x86 (008-91117)
- 64-Bit Solaris™ 10 on Intel® x64 (008-91116)
- 32-Bit Solaris™ 8 on SPARC® using Sun™ Compiler (008-91119)
- 32-Bit Solaris™ 8 on SPARC® using GCC Compiler (008-91130)

- 32-Bit Mac OS® X Using Universal Binaries (008-91301)
- 64-Bit Mac OS® X on Intel® x64 (008-91303)
- 32-Bit Android on ARMv7-A (008-91998)
- 64-Bit Android on ARMv8-A (89-00161-00)
- 64-Bit iOS on ARMv7-A (008-91990)
- ~~32-Bit Linux on ARMv7-A (008-92001) – retired~~
- 64-Bit Linux on ARMv8-A (89-00163-00)

32-Bit Microsoft Windows using Microsoft Visual C++ (008-91204)

Supported Configurations

The following table describes the Merge DICOM Toolkit system requirements for 32-Bit Microsoft Windows using Microsoft Visual C++.

Category	Requirement
Hardware	32-bit Microsoft Windows supported Ethernet Network Interface Card.
Software (base)	32-bit Microsoft Windows
Software (development)	32-bit Microsoft Windows Microsoft Visual Studio 2005 SP1 Development System and Tools (or later).

The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for 32-Bit Microsoft Windows using Microsoft Visual C++ is provided in two forms: a **DLL** and a **static library**.

Static and dynamic link libraries

A static library is a collection of subroutines that are callable by your programs. To use them, simply link them with your program.

Dynamic link libraries, or DLL for short, are the cornerstone of Windows. A DLL is similar to a static library; it that it contains code that will be executed by many different modules. The difference, however, is that the code is not included in the executable file built by the linker or loader. Instead, the code is loaded at runtime when the resources are requested. The code is then mapped into the processes address space.

Two files make up this Merge DICOM Toolkit DLL: `MC3ADLL.LIB` and `MC3ADV.DLL`. The `MC3ADLL.LIB` file is called the Merge DICOM Toolkit **import library**. When a sample program is linked, the user must specify this file in the load line so that the loader will know how to resolve any calls made to Merge DICOM Toolkit DLL. The DLL file contains the actual “executable” code used by any programs that has linked with the import library.

The compile options in the following table should be used:

Compile Options	Description
/O2	(Optional) optimize for speed
/MT	Application is multithreaded and uses static C libraries. This option should be used when linking with the static library (<code>mc3adv.lib</code>).
/MD	Application is multithreaded and uses C libraries in the form of DLLs. This option should be used when linking with the DLL (<code>mc3adll.lib</code>).
/MDd	Application is multithreaded and uses debug C libraries in the form of DLLs. This option should be used when linking with the DLL (<code>mc3adlld.lib</code>).

The link options in the following table should be used:

Link Options	Description
/SUBSYSTEM:console	Needed for sample applications
/INCREMENTAL:no	Do not use incremental links
/MACHINE:I386	Specify Intel x86 machine type
/LARGEADDRESSAWARE	Specify the ability to support larger than a 2GB address space.

See the example `makefile` supplied with the toolkit for an example link line with further explanation of compiler options.

Note: Developer Studio projects are not included with the library.

Miscellaneous Notes

Threading Support

The Merge DICOM Toolkit for 32-bit Windows supports multi-threaded applications. Please see the *User's Manual* for details on the limitations for using Merge DICOM Toolkit with multiple threads.

Debug Version

The Merge DICOM Toolkit for 32-bit Windows provides an extra debug version of the library that shall be used with the debug versions of your application. This library should be used with /MDd in compiles. It enables diagnostic tools that depend on the Debug Runtime to work.

Compression Support

The Merge DICOM Toolkit for 32-bit Windows supports the PICTools (formerly known as Pegasus) libraries for compression/decompression from Accusoft (formerly Pegasus Imaging).

The Lossless and Lossy JPEG compressors can be utilized within your application without purchasing an additional license from Accusoft. However, the Lossy and Lossless JPEG Pegasus libraries are limited to compress and decompress at a maximum rate of 3 frames per second. This limit can be removed by purchasing a license from Accusoft (www.accusoft.com) and configuring that license in our `mergecom.pro` configuration file.

Note: To use the JPEG2000 compressor or decompressor in your applications, you must purchase a separate license from Accusoft.

Unicode Support

The Merge DICOM Toolkit for 32-bit Windows supports Unicode conversion of DICOM defined character sets with and without code extensions. Two optional shared object libraries, `icuuc49.dll` and `icudt49.dll`, are distributed with the toolkit and are used to perform Unicode character set conversion. Users that wish to use Unicode conversion functions must call `MC_Enable_Unicode_Conversion` to initialize the shared object libraries and ensure the dependency files listed in the table below are available at runtime. Existing users that have no plan to use the Unicode conversion functions don't need to deploy the two shared objects and their dependency files.

Dependency files of `icuuc49.dll` and `icudt49.dll` on 32-bit Windows platform:

Unicode conversion library	Dependency File	Description
icuuc49.dll icudt49.dll	msvcr80.dll	Microsoft runtime
	msvc80.dll	Microsoft runtime
	advapi32.dll	Microsoft runtime
	kernel32.dll	Microsoft runtime

API Changes for Windows

Three toolkit functions have different parameters for Windows. The following is a listing of the new parameters to these functions:

```
void MC_List_Message(
    int AmessageID,
    char* Afilename
);
```

```
void MC_List_Item(  
    int AitemID,  
    char* Afilename  
);  
void MC_List_File(  
    int AfileID,  
    char* Afilename  
);
```

The second parameter to each of these functions is defined as being a `FILE*` in the *Reference Manual*. Because `FILE*` variables cannot be passed to a DLL, these functions have been changed to pass a filename instead. With the static library, the output will be sent to `stdout` when the `Afilename` parameter is set to `NULL`. If it is set to `NULL` when using the DLL, an exception error will occur. When `Afilename` is set to a filename, the message, item, or file object is listed to this text file.

Sample Applications

Several sample programs are provided with Merge DICOM Toolkit. They are as follows: `stor_scp`, `stor_scu`, `qr_scp`, `qr_scu`, `qr_get_scp`, `qr_get_scu`, `prnt_scp`, `prnt_scu`, `work_scp`, `work_scu`, `ssl_scp`, `ssl_scu`, `comp`, `mpeg2dicom`, `duplicate` and `med_fsu`. All of these programs have been customized to work with Windows with the store server program having been given extra attention. The sample storage server (`stor_scp.c`) has been modified to create multiple threads to handle multiple simultaneous DICOM associations. A version of each sample application can be generated using both the static library and the Merge DICOM Toolkit DLL.

Installation

The following notes provide some details regarding installation requirements and procedures for the Merge DICOM Toolkit running on Windows operating systems.

The Merge DICOM Toolkit Requirements

The distribution media contains an archive zip file. In order to use the libraries on Intel based computers, the following hardware and software requirements must be met. Please note that the user must meet the requirements for a Windows operating system installation before the requirements for Merge DICOM Toolkit installation. We also assume you have installed the Microsoft Visual C++ compiler.

Hardware

- An Intel 32-bit x86 based microprocessor.
- One (or more) hard disks, with approximately 30 megabytes of free disk space for the installation. For execution, 40 megabytes should be sufficient. Image storage requires extra storage space.
- A CD-ROM drive.
- An Ethernet network adapter card which is supported by the Microsoft Windows operating system.

Software

- Microsoft® Windows operating system configured to use the TCP/IP transport services.
- Microsoft Visual Studio 2005 SP1 or higher

Post-Installation

Perform post-installation configuration

This section describes the necessary post-installation procedures that must be completed before Merge DICOM Toolkit is ready to be used.

Environment Variables

Merge DICOM Toolkit requires that an environment variable called MERGE_INI be set to point to the current initialization file (shipped as `merge.ini`). If this environment variable is not set, Merge DICOM Toolkit will look in the execution directory of your application to find this configuration file.

Directory and File Paths

When specifying the directory or file path for a windows program, it is necessary to declare the explicit directory or file path. The examples below are from the `merge.ini` file and the `mergecom.pro` file.

merge.ini

```
[MergeCOM3]
# MergeCOM-3 system profile parameters
MERGECOM_3_PROFILE = C:\mc3alib\mergecom.pro
# MergeCOM-3 service and message definitions
MERGECOM_3_SERVICES = C:\mc3db\mergecom.srv
# MergeCOM-3 application configurations
MERGECOM_3_APPLICATIONS = C:\mc3alib\mergecom.app
# Message log parameters
LOG_FILE = C:\mc3alib\merge.log # Name of log file
```

mergecom.pro

```
#####
#           MergeCOM-3 MESSAGE CONFIGURATION SECTION
#####
[MESSAGE_PARMS]
LARGE_DATA_STORE      = MEM # |FILE |VMEM Default = MEM
LARGE_DATA_SIZE       = 200
OBOW_BUFFER_SIZE     = 4096
DICTIONARY_FILE       = C:\mc3db\mrgcom3.dct
TEMP_FILE_DIRECTORY  = C:\temp
MSG_INFO_FILE        = C:\mc3db\mrgcom3.msg
```

Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me.txt	Information on this release of the toolkit.
	0_w32	Information concerning how this distribution was created.
mc3apps	comp.c	Sample compression/decompression application.
	comp.vcproj	VS 2005 project file.
	ct.img	Example CT image file. (This file is generated by mc3file and can be regenerated by the user, if need be.)
	duplicate.c	Sample for using MC_Standard_Compressor/MC_Standard_Decompressor via MC_Duplicate_Message.
	duplicate.vcproj	VS 2005 project file.
	general_util.c	General utilities for all sample programs.
	general_util.h	General utilities for all sample programs.
	makefile	Makefile for example programs.
	mc3apps.sln	VS 2005 solution file for sample applications.
	med_fsu.c	Media File Set Updater Application.
	med_fsu.vcproj	VS 2005 project file.
	merge.ini	Merge DICOM Toolkit Initialization Configuration File. (Used by all sample applications.)
	mergecom.app	Merge DICOM Toolkit Application Profile Configuration File.
	mergecom.pro	Merge DICOM Toolkit System Profile Configuration File.
	mergecom.srv	Merge DICOM Toolkit Service Profile Configuration File.
	mpeg2dicom.c	Sample for packing/unpacking MPEG2 streams into/from DICOM files.
	mpeg2dicom.vcproj	VS 2005 project file.

Directory	File	Description
	prnt_scp.c	Sample Print SCP Application.
	prnt_scp.vcproj	VS 2005 project file.
	prnt_scu.c	Sample Print SCU Application.
	prnt_scu.vcproj	VS 2005 project file.
	prnt_svc.h	Sample Print Application header file.
	qr.h	Sample Query/Retrieve Application Include file.
	qr_get_scp.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_get_scp.vcproj	VS 2005 project file.
	qr_get_scu.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_get_scu.vcproj	VS 2005 project file.
	qr_scp.c	Sample Query/Retrieve SCP Application.
	qr_scp.vcproj	VS 2005 project file.
	qr_scu.c	Sample Query/Retrieve SCU Application.
	qr_scu.vcproj	VS 2005 project file.
	qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
	SampleApps.vsp s	Visual Studio project property file.
	sreport.c	Sample Structured Report Application
	sreport.vcproj	VS 2005 project file
	ssl_samp.c	SSL callbacks for SSL SCU and SSL SCP.
	ssl_samp.h	Header file for SSL callbacks for SSL SCU and SSL SCP.
	ssl_scp.c	Sample SCP application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.

Directory	File	Description
	ssl_scp.h	Header file for sample SSL SCP application.
	ssl_scp.vcproj	VS 2005 project file.
	ssl_scu.c	Sample SCU application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	ssl_scu.vcproj	VS 2005 project file.
	stor_scp.c	Sample Storage SCP Application.
	stor_scp.vcproj	VS 2005 project file.
	stor_scu.c	Sample Storage SCU Application.
	stor_scu.vcproj	VS 2005 project file.
	work.dat	Database flat file, used by the Modality Worklist SCP application.
	workdata.c	Sample database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	workdata.h	Sample header file for database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	work_scp.c	Sample Modality Worklist and Modality Performed Procedure Step SCP Application.
	work_scp.vcproj	VS 2005 project file.
	work_scu.c	Sample Modality Worklist and Modality Performed Procedure Step SCU Application.
	work_scu.vcproj	VS 2005 project file.

Directory	File	Description
mc3bin	genconf.exe	Configuration source file generation utility.
	gendict.exe	Dictionary source file generation utility.
	mc3comp.exe	Compare the values within two DICOM message or file objects.
	mc3conv.exe	Convert a DICOM message or file object into a new transfer syntax.
	mc3dcomb.exe	Runtime Dictionary Combine utility.
	mc3dict.exe	Runtime DICOM Data Dictionary utility.
	mc3echo.exe	DICOM Echo Test Utility.
	mc3file.exe	Generate a DICOM message object.
	mc3icomb.exe	Runtime Info Combine utility.
	mc3info.exe	Runtime Message Database generation utility.
	mc3list.exe	List a DICOM message object.
	mc3valid.exe	Validate a DICOM message object.
mc3doc	Database.pdf	Merge DICOM Toolkit DICOM Database Manual.
	Platform.pdf	This document.
	Refer.pdf	Merge DICOM Toolkit Reference Manual.
	Sample.pdf	Merge DICOM Toolkit Sample Applications Guide.
	User.pdf	Merge DICOM Toolkit User's Manual.
mc3inc	diction.h	DICOM Data Dictionary macros.
	mc3items.h	Library include file for use with mc3adv.lib.
	mc3media.h	Library include file for use with mc3adv.lib.
	mc3msg.h	Library include file for use with mc3adv.lib.
	mc3services.h	Library include file for use with mc3adv.lib.
	mc3status.h	Library include file for use with mc3adv.lib.

Directory	File	Description
	mergecom.h	Library include file for use with mc3adv.lib.
	pcd.h	Pegasus include file.
	pic.h	Pegasus include file.
	stdtypes.h	Pegasus include file.
mc3lib	icuuc49.dll	Unicode conversion library.
	icudt49.dll	Unicode conversion data.
	jansson.dll	Library for encoding, decoding and manipulating JSON data.
	libxml2.dll	XML parsing library.
	mc3adll.lib	Merge DICOM Toolkit export library for mc3adv.dll.
	mc3adlld.lib	Merge DICOM Toolkit export library for mc3advd.dll.
	mc3adv.dll	Merge DICOM Toolkit dynamic link library.
	mc3adv.lib	Merge DICOM Toolkit software object code library.
	mc3advd.pdb	Debug database for Merge DICOM Toolkit dynamic link library.
	mc3advd.dll	Merge DICOM Toolkit dynamic link library for debug builds.
	mc3advd.pdb	Debug database for Merge DICOM Toolkit dynamic link library.
	picn20.dll	Pegasus dispatcher DLL.
	picn6120.dll	Pegasus library for JPEG-LS decompression.
	picn6220.dll	Pegasus library for JPEG Lossless compression.
	picn6320.dll	Pegasus library for JPEG Lossless decompression.
	picn6420.dll	Pegasus library for JPEG Lossy compression.

Directory	File	Description
	picn6520.dll	Pegasus library for JPEG Lossy decompression.
	picn6820.dll	Pegasus library for JPEG 2000 compression.
	picn6920.dll	Pegasus library for JPEG 2000 decompression.
	picnm.lib	Pegasus export library for Pegasus DLLs.
mc3msg	default.pfl	Default configuration file used by mc3file.
	diction.pfl	Runtime DICOM Data Dictionary profile.
	message.txt	DICOM message formats for INFO purposes.
	info.pfl	Runtime Message Database profile.
	mc3dcomb.pfl	Sample configuration file for use with mc3dcomb utility.
	mc3icomb.cfg	Sample configuration file for use with mc3icomb utility.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

64-Bit Microsoft Windows using Microsoft Visual C++ (008-91208)

Supported Configurations

The following table describes the Merge DICOM Toolkit system requirements for 64-Bit Microsoft Windows using Microsoft Visual C++.

Category	Requirement
Hardware	Microsoft Windows 2003/XP supported Ethernet Network Interface Card Intel EM64T or Advanced Micro Devices AMD64 Processor
Software (base)	A 64-bit Microsoft Windows operating system
Software (development)	Platform SDK or Microsoft Visual Studio 2005

The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for 64-Bit Microsoft Windows using Microsoft Visual C++ is provided in two forms: a **DLL** and a **static library**.

Static and dynamic link libraries

A static library is a collection of subroutines that are callable by your programs. To use them, simply link them with your program.

Dynamic link libraries, or DLL for short, are the cornerstone of Windows. A DLL is similar to a static library; it that it contains code that will be executed by many different modules. The difference, however, is that the code is not included in the executable file built by the linker or loader. Instead, the code is loaded at runtime when the resources are requested. The code is then mapped into the processes address space.

Two files make up this Merge DICOM Toolkit DLL: `MC3ADLL64.LIB` and `MC3ADV64.DLL`. The `MC3ADLL64.LIB` file is called the Merge DICOM Toolkit **import library**. When a sample program is linked, the user must specify this file in the load line so that the loader will know how to resolve any calls made to Merge DICOM Toolkit DLL. The DLL file contains the actual “executable” code used by any programs that has linked with the import library.

The following compile options should be used:

Compile Options	Description
<code>/O2</code>	(Optional) optimize for speed.
<code>/MT</code>	Application is multithreaded and uses static C libraries. This option should be used when linking with the static library (<code>mc3adv64.lib</code>).
<code>/MD</code>	Application is multithreaded and uses C libraries in the form of DLLs. This option should be used when linking with the DLL (<code>mc3adll64.lib</code>).
<code>/MDd</code>	Application is multithreaded and uses debug C libraries in the form of DLLs. This option should be used when linking with the DLL (<code>mc3adll64d.lib</code>).

The following link options should be used:

Link Options	Description
<code>/SUBSYSTEM:console</code>	Needed for sample applications.
<code>/INCREMENTAL:no</code>	Do not use incremental links.
<code>/MACHINE:AMD64</code>	Specify Intel machine type.

See the example `makefile` supplied with the toolkit for an example link line with further explanation of compiler options.

Note: Developer Studio projects are not included with the library.

Miscellaneous Notes

Debug Version

The Merge DICOM Toolkit for 64-bit Windows provides an extra debug version of the library that shall be used with the debug versions of your application. This library should be used with `/MDd` in compiles. It enables diagnostic tools that depend on the Debug Runtime to work.

Sample Applications

Several sample programs are provided with Merge DICOM Toolkit. They are as follows: `stor_scp`, `stor_scu`, `qr_scp`, `qr_scu`, `qr_get_scp`, `qr_get_scu`, `prnt_scp`, `prnt_scu`, `work_scp`, `work_scu`, `ssl_scp`, `ssl_scu`, `comp`, `mjpeg2dicom`, `duplicate` and `med_fsu`. All of these programs have been customized to work with Windows with the store server program having been given extra attention. The sample storage server (`stor_scp.c`) has been modified to create multiple threads to handle multiple simultaneous DICOM associations. A version of each sample application can be generated using both the static library and the Merge DICOM Toolkit DLL.

API Changes for Windows

Three toolkit functions have different parameters for Windows. The following is a listing of the new parameters to these functions:

```
void MC_List_Message(  
    int AmessageID,  
    char* Afilename  
);  
void MC_List_Item(  
    int AitemID,  
    char* Afilename  
);  
void MC_List_File(  
    int AfileID,  
    char* Afilename  
);
```

The second parameter to each of these functions is defined as being a `FILE*` in the *Reference Manual*. Because `FILE*` variables cannot be passed to a DLL, we have changed these functions to pass a filename instead. With the static library, the output will be sent to `stdout` when the `Afilename` parameter is set to `NULL`. If it is set to `NULL` when using the DLL, an exception error will occur. When `Afilename` is set to a filename, the message, item, or file object is listed to this text file.

Threading Support

The Merge DICOM Toolkit for 64-bit Windows supports multi-threaded applications. Please see the *User's Manual* for details on the limitations for using Merge DICOM Toolkit with multiple threads.

Compression Support

The Merge DICOM Toolkit for 64-bit Windows supports the PICTools (formerly known as Pegasus) libraries for compression/decompression from Accusoft.

The Lossless and Lossy JPEG compressors can be utilized within your application without purchasing an additional license from Accusoft. However, the Lossy and Lossless JPEG Pegasus libraries are limited to compress and decompress at a maximum rate of 3 frames per second. This limit can be removed by purchasing a license from Accusoft (www.accusoft.com) and configuring that license in our mergecom.pro configuration file.

Note: To use the JPEG2000 compressor or decompressor in your applications, you must purchase a separate license from Accusoft.

Unicode Support

The Merge DICOM Toolkit for 64-bit Windows supports Unicode conversion of DICOM defined character sets with and without code extensions. Two optional shared object libraries, icuuc49.dll and icudt49.dll, are distributed with the toolkit and are used to perform Unicode character set conversion. Users that wish to use Unicode conversion functions must call `MC_Enable_Unicode_Conversion` to initialize the shared object libraries and ensure the dependency files listed in the table below are available at runtime. Existing users that have no plan to use the Unicode conversion functions don't need to deploy the two shared objects and their dependency files.

Dependency files of icuuc49.dll and icudt49.dll on 64-bit Windows platform:

Unicode conversion library	Dependency File	Description
icuuc49.dll icudt49.dll	msvcr80.dll	Microsoft runtime
	msvcp80.dll	Microsoft runtime
	advapi32.dll	Microsoft runtime
	kernel32.dll	Microsoft runtime

Installation

The following notes provide some details regarding installation requirements and procedures for the Merge DICOM Toolkit running on Windows operating systems.

The Merge DICOM Toolkit Requirements

The distribution media contains an archive zip file. In order to use the libraries on Intel based computers, the following hardware and software requirements must be met. Please note that the user must meet the requirements for a Windows operating system installation before the requirements for Merge DICOM Toolkit installation. We also assume you have installed the Microsoft Platform SDK.

Hardware

- Intel 64-bit PC Pentium microprocessor or Micro Devices AMD64 Processor
- One (or more) hard disks, with approximately 30 megabytes of free disk space for the installation. For execution, 40 megabytes should be sufficient. Image storage requires extra storage space.
- A CD-ROM drive.
- An Ethernet network adapter card *supported by Windows 2003/XP*.

Software

- Microsoft® Windows operating system configured to use the TCP/IP transport services.
- Microsoft Platform SDK.

Post-Installation

Perform post-installation configuration

This section describes the necessary post-installation procedures that must be completed before Merge DICOM Toolkit is ready to be used.

Environment Variables

Merge DICOM Toolkit requires that an environment variable called MERGE_INI be set to point to the current initialization file (shipped as `merge.ini`). If this environment variable is not set, Merge DICOM Toolkit will look in the execution directory of your application to find this configuration file.

Directory and File Paths

When specifying the directory or file path for a windows program, it is necessary to declare the explicit directory or file path. The examples below are from the `merge.ini` file and the `mergecom.pro` file.

merge.ini

```
[MergeCOM3]
# MergeCOM-3 system profile parameters
MERGECOM_3_PROFILE = C:\mc3alib\mergecom.pro

# MergeCOM-3 service and message definitions
MERGECOM_3_SERVICES = C:\mc3db\mergecom.srv
# MergeCOM-3 application configurations
MERGECOM_3_APPLICATIONS = C:\mc3alib\mergecom.app
# Message log parameters.
LOG_FILE = C:\mc3alib\merge.log # Name of log file
```


mergecom.pro

```

#=====
#           MergeCOM-3 MESSAGE CONFIGURATION SECTION
#=====
[MESSAGE_PARMS]
    LARGE_DATA_STORE      = MEM # |FILE |VMEM Default =
MEM
    LARGE_DATA_SIZE       = 200
    OBOW_BUFFER_SIZE      = 4096
    DICTIONARY_ACCESS     = MEM # | FILE Default = MEM
    DICTIONARY_FILE       = C:\mc3db\mrgcom3.dct
    TEMP_FILE_DIRECTORY   = C:\temp
    MSG_INFO_FILE        = C:\mc3db\mrgcom3.msg

```

Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me.txt	Information on this release of the toolkit.
	0_w64	Information concerning how this distribution was created.
mc3apps	comp.c	Sample compression/decompression application.
	comp.vcproj	VS 2005 project file.
	ct.img	Example CT image file. (This file is generated by mc3file and can be regenerated by the user, if need be.)
	duplicate.c	Sample for using MC_Standard_Compressor/MC_Standard_Decompressor via MC_Duplicate_Message.
	duplicate.vcproj	VS 2005 project file.
	general_util.c	General utilities for all sample programs.
	general_util.h	General utilities for all sample programs.
	makefile	Makefile for example programs.
	mc3apps.sln	VS 2005 solution file for sample applications.
	med_fsu.c	Media File Set Updater Application.

Directory	File	Description
	med_fsu.vcproj	VS 2005 project file.
	merge.ini	Merge DICOM Toolkit Initialization Configuration File. (Used by all sample applications.)
	mergecom.app	Merge DICOM Toolkit Application Profile Configuration File.
	mergecom.pro	Merge DICOM Toolkit System Profile Configuration File.
	mergecom.srv	Merge DICOM Toolkit Service Profile Configuration File.
	mpeg2dicom.c	Sample for packing/unpacking MPEG2 streams into/from DICOM files.
	mpeg2dicom.vcproj	VS 2005 project file.
	prnt_scp.c	Sample Print SCP Application.
	prnt_scp.vcproj	VS 2005 project file.
	prnt_scu.c	Sample Print SCU Application.
	prnt_scu.vcproj	VS 2005 project file.
	prnt_svc.h	Sample Print Application header file.
	qr.h	Sample Query/Retrieve Application Include file.
	qr_get_scp.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_get_scp.vcproj	VS 2005 project file.
	qr_get_scu.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_get_scu.vcproj	VS 2005 project file.
	qr_scp.c	Sample Query/Retrieve SCP Application.
	qr_scp.vcproj	VS 2005 project file.
	qr_scu.c	Sample Query/Retrieve SCU Application.
	qr_scu.vcproj	VS 2005 project file.

Directory	File	Description
	qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
	SampleApps.vsp ops	Visual Studio project property file.
	sreport.c	Sample Structured Report Application
	sreport.vcproj	VS 2005 project file
	ssl_samp.c	SSL callbacks for SSL SCU and SSL SCP.
	ssl_samp.h	Header file for SSL callbacks for SSL SCU and SSL SCP.
	ssl_scp.c	Sample SCP application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	ssl_scp.h	Header file for sample SSL SCP application.
	ssl_scp.vcproj	VS 2005 project file.
	ssl_scu.c	Sample SCU application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	ssl_scu.vcproj	VS 2005 project file.
	stor_scp.c	Sample Storage SCP Application.
	stor_scp.vcproj	VS 2005 project file.
	stor_scu.c	Sample Storage SCU Application.
	stor_scu.vcproj	VS 2005 project file.
	work.dat	Database flat file, used by the Modality Worklist SCP application.
	workdata.c	Sample database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	workdata.h	Sample header file for database functions used by Modality Worklist and Performed Procedure Step SCP Application.

Directory	File	Description
	work_scp.c	Sample Modality Worklist and Modality Performed Procedure Step SCP Application.
	work_scp.vcproj	VS 2005 project file
	work_scu.c	Sample Modality Worklist and Modality Performed Procedure Step SCU Application.
	work_scu.vcproj	VS 2005 project file
mc3bin	genconf.exe	Configuration source file generation utility.
	gendict.exe	Dictionary source file generation utility.
	mc3comp.exe	Compare the values within two DICOM message or file objects.
	mc3conv.exe	Convert a DICOM message or file object into a new transfer syntax.
	mc3dcomb.exe	Runtime Dictionary Combine utility.
	mc3dict.exe	Runtime DICOM Data Dictionary utility.
	mc3echo.exe	DICOM Echo Test Utility.
	mc3file.exe	Generate a DICOM message object.
	mc3icomb.exe	Runtime Info Combine utility.
	mc3info.exe	Runtime Message Database generation utility.
	mc3list.exe	List a DICOM message object.
	mc3valid.exe	Validate a DICOM message object.
mc3doc	Database.pdf	Merge DICOM Toolkit DICOM Database Manual.
	Platform.pdf	This document.
	Refer.pdf	Merge DICOM Toolkit Reference Manual.
	Sample.pdf	Merge DICOM Toolkit Sample Applications Guide.
	User.pdf	Merge DICOM Toolkit User's Manual.
mc3inc	diction.h	DICOM Data Dictionary macros.

Directory	File	Description
	mc3items.h	Library include file for use with mc3adv.lib.
	mc3media.h	Library include file for use with mc3adv.lib.
	mc3msg.h	Library include file for use with mc3adv.lib.
	mc3services.h	Library include file for use with mc3adv.lib.
	mcstatus.h	Library include file for use with mc3adv.lib.
	mergecom.h	Library include file for use with mc3adv.lib.
	pcd.h	Pegasus include file.
	pic.h	Pegasus include file.
	stdtypes.h	Pegasus include file.
mc3lib	icuuc49.dll	Unicode conversion library.
	icudt49.dll	Unicode conversion data.
	jansson.dll	Library for encoding, decoding and manipulating JSON data.
	libxml2.dll	XML parsing library.
	mc3adll64.lib	Merge DICOM Toolkit export library for mc3adv.dll.
	mc3adll64d.lib	Merge DICOM Toolkit export library for mc3advd.dll.
	mc3adv64.dll	Merge DICOM Toolkit dynamic link library.
	mc3adv64.lib	Merge DICOM Toolkit software object code library.
	mc3adv64d.pdb	Debug database for Merge DICOM Toolkit dynamic link library.
	mc3adv64d.dll	Merge DICOM Toolkit dynamic link library for debug builds.
	mc3advd.pdb	Debug database for Merge DICOM Toolkit dynamic link library.
	picx20.dll	Pegasus dispatcher DLL.

Directory	File	Description
	picx6120.ssm	Pegasus library for JPEG-LS decompression.
	picx6220.ssm	Pegasus library for JPEG Lossless compression.
	picx6320.ssm	Pegasus library for JPEG Lossless decompression.
	picx6420.ssm	Pegasus library for JPEG Lossy compression.
	picx6520.ssm	Pegasus library for JPEG Lossy decompression.
	picx6820.ssm	Pegasus library for JPEG 2000 compression.
	picx6920.ssm	Pegasus library for JPEG 2000 decompression.
	picxm.lib	Pegasus export library for Pegasus DLLs.
mc3msg	default.pfl	Default configuration file used by mc3file.
	diction.pfl	Runtime DICOM Data Dictionary profile.
	message.txt	DICOM message formats for INFO purposes.
	info.pfl	Runtime Message Database profile.
	mc3dcomb.pfl	Sample configuration file for use with mc3dcomb utility.
	mc3icomb.cfg	Sample configuration file for use with mc3icomb utility.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

32-Bit Microsoft Windows using Borland® C++ (008-91205)

Supported Configurations

The following table describes the system requirements for the Merge DICOM Toolkit.

Category	Requirement
Hardware	32-bit Microsoft Windows Ethernet Network Interface Card.
Software (base)	32-bit Microsoft Windows.
Software (development)	32-bit Microsoft Windows. Borland C++ Builder 2006 or later.

The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for 32-Bit Microsoft Windows using Borland® C++ is provided in two forms: a **DLL** and a **static library**.

Static and dynamic link libraries

A static library is a collection of subroutines that are callable by your programs. To use them, simply link them with your program.

Dynamic link libraries, or DLL for short, are the cornerstone Windows. A DLL is similar to a static library; in that it contains code that will be executed by many different modules. The difference, however, is that the code is not included in the executable file built by the linker or loader. Instead, the code is loaded at runtime when the resources are requested. The code is then mapped into the processes address space.

Two files make up this Merge DICOM Toolkit DLL: `MC3ADLL.LIB` and `MC3ADV.DLL`. The `MC3ADLL.LIB` file is called the Merge DICOM Toolkit import library. When a sample program is linked, the user must specify this file in the load line so that the loader will know how to resolve any calls made to Merge DICOM Toolkit DLL. The DLL file contains the actual “executable” code used by any programs that have linked with the import library.

The following compile options should be used:

Compile Options	Description
-O2	(Optional) optimize for speed.
-WC	Generate a Windows “console” program.
-WM	Application is multithreaded.

The following link options should be used:

Link Options	Description
-Tpe	Output file is an “.EXE”.
-ap	Generate a protected mode executable that runs in “console” mode.

-x	Do not use a “map” file.
----	--------------------------

See the example `makefile` supplied with the toolkit for an example compile and link line. Some additional compile options were used with the sample applications. The following describes their usage.

- `-w-aus` — Identifier not used. Occasionally, in the sample applications, the status of a function call is returned to a variable in an “if” statement. This is usually done to aid debugging. The Borland compiler will note that this variable is never used. This flag will mask these types of warnings
- `-w-par` — Parameter not used. Occasionally, in the sample applications, parameters are passed into a function, only to be passed to another function. Borland’s compiler will note that the original parameter was not used in the original function, when it was in fact passed to another sub-function. This flag will mask these types of warnings.

Another thing to note when looking at the sample application `makefile` is that it is necessary to rely on Borland’s “response” files. A response file is an ordinary text file that contains either linker arguments, or compiler arguments.

A response file is needed because there is a Windows limitation to the length of all command line arguments passed to a program. With the number of compiler defines and flags that are used, this length would be exceeded. Therefore, “make” will temporarily create the response file, pass its filename to the linker or compiler as an argument, and then delete it. The temporary response files are named with an “*.rsp” or “*.dsp” file extension.

Note: Integrated Development Environment (IDE) projects are not included with the library.

Miscellaneous Notes

API Changes for Windows

Three toolkit functions have different parameters for Windows. The following is a listing of the new parameters to these functions:

```
void MC_List_Message(
    int AmessageID,
    char* Afilename
);
void MC_List_Item(
    int AitemID,
    char* Afilename
);
void MC_List_File(
    int AfileID,
    char* Afilename
);
```

The second parameter to each of these functions is defined as being a `FILE*` in the reference manual. Because `FILE*` variables cannot be passed to a DLL, we

have changed these functions to pass a filename instead. With the static library, the output will be sent to `stdout` when the `Afilename` parameter is set to `NULL`. If it is set to `NULL` when using the DLL, an exception error will occur. When `Afilename` is set to a filename, the message, item, or file object is listed to this text file.

Threading Support

The Merge DICOM Toolkit for Windows supports multi-threaded applications. Please see the *User's Manual* for details on the limitations for using Merge DICOM Toolkit with multiple threads.

Compression Support

The Merge DICOM Toolkit for 32-bit Windows supports the PICTools (formerly known as Pegasus libraries for compression/decompression from Accusoft.

The Lossless and Lossy JPEG compressors can be utilized within your application without purchasing an additional license from Accusoft. However, the Lossy and Lossless JPEG Pegasus libraries are limited to compress and decompress at a maximum rate of 3 frames per second. This limit can be removed by purchasing a license from Accusoft (www.accusoft.com) and configuring that license in our `mergecom.pro` configuration file.

Note: To use the JPEG2000 compressor or decompressor in your applications, you must purchase a separate license from Accusoft.

Unicode Support

The Merge DICOM Toolkit for 32-bit Windows supports Unicode conversion of DICOM defined character sets with and without code extensions. Two optional shared object libraries, `icuuc49.dll` and `icudt49.dll`, are distributed with the toolkit and are used to perform Unicode character set conversion. Users that wish to use Unicode conversion functions must call `MC_Enable_Unicode_Conversion` to initialize the shared object libraries and ensure the dependency files listed in the table below are available at runtime. Existing users that have no plan to use the Unicode conversion functions don't need to deploy the two shared objects and their dependency files.

Dependency files of `icuuc49.dll` and `icudt49.dll` on 32-bit Windows platform:

Unicode conversion library	Dependency File	Description
icuuc49.dll icudt49.dll	msvcr80.dll	Microsoft runtime
	msvcp80.dll	Microsoft runtime
	advapi32.dll	Microsoft runtime
	kernel32.dll	Microsoft runtime

Installation

The following notes provide some details regarding installation requirements and procedures for the Merge DICOM Toolkit running on Windows operating systems.

Merge DICOM Toolkit Requirements

The distribution media contains archived, ready-to-use versions of the Merge DICOM Toolkit libraries. In order to use the libraries on an Intel based computer, the following hardware and software requirements must be met. Please note that the user must meet the requirements for a Windows NT operating system installation before the requirements for Merge DICOM Toolkit installation. We also assume you have installed the Borland C++ Development Suite, with Design Tools.

Hardware

- An Intel 32-bit x86 based microprocessor.
- One (or more) hard disks, with approximately 30 megabytes of free disk space for the installation. For compilation and execution of the sample applications, 40 megabytes should be sufficient. Image storage requires extra storage space.
- An Ethernet network adapter card which is supported by Windows.

Software

- A 32-bit Microsoft® Windows Platform configured to use the TCP/IP transport services.
- Borland® C++ Builder 2006 or later.

Post-Installation

Perform post-installation configuration.

This section describes the necessary post-installation procedures that must be completed before Merge DICOM Toolkit is ready to be used.

Environment Variables

Merge DICOM Toolkit requires that an environment variable called MERGE_INI be set to point to the current initialization file (shipped as `merge.ini`). If this environment variable is not set, Merge DICOM Toolkit will look in the execution directory of your application to find this configuration file.

Directory and File Paths

When specifying the directory or file path for a windows program, it is necessary to declare the explicit directory or file path. The examples below are from the `merge.ini` file and the `mergecom.pro` file.

merge.ini

```
[MergeCOM3]
# MergeCOM-3 system profile parameters
MERGECOM_3_PROFILE = C:\mc3apps\mergecom.pro
```

```
# MergeCOM-3 service and message definitions
MERGECOM_3_SERVICES = C:\mc3apps\mergecom.srv
# MergeCOM-3 application configurations
MERGECOM_3_APPLICATIONS = C:\mc3apps\mergecom.app
# Message log parameters.
LOG_FILE = C:\mc3apps\merge.log # Name of log file
```

mergecom.pro

```
#####
#           MergeCOM-3 MESSAGE CONFIGURATION SECTION
#####
[MESSAGE_PARMS]
    LARGE_DATA_STORE      = MEM # | FILE | VMEM Default = MEM
    LARGE_DATA_SIZE       = 200
    OBOW_BUFFER_SIZE      = 4096
    DICTIONARY_ACCESS     = MEM # | FILE Default = MEM
    DICTIONARY_FILE       = C:\mc3msg\mrgcom3.dct
    TEMP_FILE_DIRECTORY   = C:\temp
    MSG_INFO_FILE        = C:\mc3msg\mrgcom3.msg
```

Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me.txt	Information on this release of the toolkit.
	0_bcc	Information concerning how this distribution was created.
mc3apps	comp.c	Sample compression/decompression application.
	ct.img	Example CT image file. (This file is generated by mc3file and can be regenerated by the user, if need be.)
	duplicate.c	Sample for using MC_Standard_Compressor/MC_Standard_Decompressor via MC_Duplicate_Message.
	general_util.c	General utilities for all sample programs.
	general_util.h	General utilities for all sample programs.
	makefile	Makefile for example programs.
	med_fsu.c	Media File Set Updater Application.
	merge.ini	Merge DICOM Toolkit Initialization Configuration File. (Used by all sample applications.)

Directory	File	Description
	mergecom.app	Merge DICOM Toolkit Application Profile Configuration File.
	mergecom.pro	Merge DICOM Toolkit System Profile Configuration File.
	mergecom.srv	Merge DICOM Toolkit Service Profile Configuration File.
	mpeg2dicom.c	Sample for packing/unpacking MPEG2 streams into/from DICOM files.
	prnt_scp.c	Sample Print SCP Application.
	prnt_scu.c	Sample Print SCU Application
	prnt_svc.h	Sample Print Application header file.
	qr.h	Sample Query/Retrieve Application Include file.
	qr_get_scp.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_get_scu.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_scp.c	Sample Query/Retrieve SCP Application.
	qr_scu.c	Sample Query/Retrieve SCU Application.
	qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
	sreport.c	Sample Structured Report Application
	ssl_samp.c	SSL callbacks for SSL SCU and SSL SCP.
	ssl_samp.h	Header file for SSL callbacks for SSL SCU and SSL SCP.
	ssl_scp.c	Sample SCP application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	ssl_scp.h	Header file for sample SSL SCP application.

Directory	File	Description
	ssl_scu.c	Sample SCU application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	stor_scp.c	Sample Storage SCP Application.
	stor_scu.c	Sample Storage SCU Application.
	work.dat	Database flat file, used by the Modality Worklist SCP application.
	workdata.c	Sample database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	workdata.h	Sample header file for database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	work_scp.c	Sample Modality Worklist and Modality Performed Procedure Step SCP Application.
	work_scu.c	Sample Modality Worklist and Modality Performed Procedure Step SCU Application.

Directory	File	Description
mc3bin	genconf.exe	Configuration source file generation utility.
	gendict.exe	Dictionary source file generation utility.
	mc3comp.exe	Compare the values within two DICOM message or file objects.
	mc3conv.exe	Convert a DICOM message or file object into a new transfer syntax.
	mc3dcomb.exe	Runtime Dictionary Combine utility.
	mc3dict.exe	Runtime DICOM Data Dictionary utility.
	mc3echo.exe	DICOM Echo Test Utility.
	mc3file.exe	Generate a DICOM message object.
	mc3icomb.exe	Runtime Info Combine utility.
	mc3info.exe	Runtime Message Database generation utility.
	mc3list.exe	List a DICOM message object.
	mc3valid.exe	Validate a DICOM message object.
mc3doc	Database.pdf	Merge DICOM Toolkit DICOM Database Manual.
	Platform.pdf	This document.
	Refer.pdf	Merge DICOM Toolkit Reference Manual.
	Sample.pdf	Merge DICOM Toolkit Sample Applications Guide.
	User.pdf	Merge DICOM Toolkit User's Manual.
mc3inc	diction.h	DICOM Data Dictionary macros.
	mc3items.h	Library include file for use with mc3adv.lib.
	mc3media.h	Library include file for use with mc3adv.lib.
	mc3msg.h	Library include file for use with mc3adv.lib.
	mc3services.h	Library include file for use with mc3adv.lib.
	mc3status.h	Library include file for use with mc3adv.lib.

Directory	File	Description
	mergecom.h	Library include file for use with mc3adv.lib.
	pcd.h	Pegasus include file.
	pic.h	Pegasus include file.
	stdtypes.h	Pegasus include file.
mc3lib	icuuc49.dll	Unicode conversion library.
	icudt49.dll	Unicode conversion data.
	jansson.dll	Library for encoding, decoding and manipulating JSON data.
	libxml2.dll	XML parsing library
	mc3adll.lib	Merge DICOM Toolkit export library for mc3adv.dll.
	mc3adv.dll	Merge DICOM Toolkit dynamic link library.
	mc3adv.lib	Merge DICOM Toolkit software object code library.
	picn20.dll	Pegasus dispatcher DLL.
	picn6120.dll	Pegasus library for JPEG-LS decompression.
	picn6220.dll	Pegasus library for JPEG Lossless compression.
	picn6320.dll	Pegasus library for JPEG Lossless decompression.
	picn6420.dll	Pegasus library for JPEG Lossy compression.
	picn6520.dll	Pegasus library for JPEG Lossy decompression.
	picn6820.dll	Pegasus library for JPEG 2000 compression.
	picn6920.dll	Pegasus library for JPEG 2000 decompression.
	picnm.lib	Pegasus export library for Pegasus DLLs.
mc3msg	default.pfl	Default configuration file used by mc3file.
	diction.pfl	Runtime DICOM Data Dictionary profile.

Directory	File	Description
	message.txt	DICOM message formats for INFO purposes.
	info.pfl	Runtime Message Database profile.
	mc3dcomb.pfl	Sample configuration file for use with mc3dcomb utility.
	mc3icomb.cfg	Sample configuration file for use with mc3icomb utility.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

32-Bit Linux® on Intel® x86 (008-91126)

Supported Configurations

The following table describes the system requirements for the Merge DICOM Toolkit.

Category	Requirement
Hardware	Linux supported PC hardware.
Software (base)	Red Hat Linux 9.0 or later ¹ .
Software (development)	GCC 3.3.2.

The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for 32-Bit Linux on Intel x86 is provided in two forms: a **shared object** and a **static library**.

Static library

A static library is a collection of subroutines that are callable by your programs. To use them, simply link them with your program.

The compiler flags needed to link with the Merge DICOM Toolkit static library are:

¹ Merge DICOM Toolkit for Linux has been developed using Red Hat Linux. Although it should function under any distribution of Linux, Red Hat Linux is the supported version if distribution specific problems arise.

Flag	Type	Description
-ldl	Compile time	Link with dynamic loading library

Shared object

A shared object is similar to a static library. It contains entry points for your application to use and call and contains code that will be executed by many different modules. The difference, however, is that the code is not included in the executable file built by the linker or loader. Instead, the code is loaded at runtime when the resources are requested. The code is then mapped into the process's address space.

The use of the two types of libraries is exactly the same: they are “linked” into an application program by the system loader after the application has been compiled. The way the system loader constructs the executable is different, however.

When a static library is linked with an application, an executable is produced that contains the code of the application and the code of the library. This is not true with code produced and linked with shared object methods. The application and the shared “library” must be compiled and linked with special compiler flags.

The compiler flags needed to link with the Merge DICOM Toolkit shared object are:

Flag	Type	Description
-fPIC	Compile Time	Produced position independent code.
-D_REENTRANT	Compile Time	Produce reentrant code that is thread safe.
-ldl	Compile Time	Link with dynamic loading library.

Also, the `LD_LIBRARY_PATH` environment variable will need to be modified to contain the path of the `mc3adv.so` file. The following is an excerpt from the `man` page of `ld` and describes the function of the `LD_LIBRARY_PATH` environment variable:

“`LD_LIBRARY_PATH` is a list of directories in which to search for libraries specified with the `-l` option. Multiple directories are separated by a colon.”
 “It is also used to specify libraries search path to the run-time linker, that is, if `LD_LIBRARY_PATH` exists in the environment, the run-time linker searches the directories named in it, before the default directories for the shared objects to be linked with the program at execution.”

It should be noted that the Merge DICOM Toolkit shared object (`mc3adv.so`) is designed to be “dynamic-safe” and optimized. This means that the shared object is safe when more than one application executed the same code at the same time.

Miscellaneous Notes

Threading Support

The Merge DICOM Toolkit for RedHat Linux supports multi-threaded applications. Please see the *User's Manual* for details on the limitations for using Merge DICOM Toolkit with multiple threads.

Compression Support

The Merge DICOM Toolkit for RedHat Linux supports the PICTools (formerly known as Pegasus) libraries for compression/decompression from Accusoft (formerly Pegasus Imaging).

The Lossless and Lossy JPEG compressors can be utilized within your application without purchasing an additional license from Accusoft. However, the Lossy and Lossless JPEG Pegasus libraries are limited to compress and decompress at a maximum rate of 3 frames per second. This limit can be removed by purchasing a license from Accusoft (www.accusoft.com) and configuring that license in our mergecom.pro configuration file.

Note: To use the JPEG2000 compressor or decompressor in your applications, you must purchase a separate license from Accusoft.

Built in RLE compression/decompression is also available.

Unicode Support

The Merge DICOM Toolkit for 32-bit Linux supports Unicode conversion of DICOM defined character sets with and without code extensions. Two optional shared object libraries, libicuuc.so.49 and libicudata.so.49, are distributed with the toolkit and are used to perform Unicode character set conversion. Users that wish to use Unicode conversion functions must call `MC_Enable_Unicode_Conversion` to initialize the shared object libraries and ensure the dependency files listed in the table below are available at runtime. Existing users that have no plan to use the Unicode conversion functions don't need to deploy the two shared objects and their dependency files.

Dependency files of libicuuc.so.49 and libicudata.so.49 on 32-bit Linux platform:

Unicode conversion library	Dependency File	Description
libicuuc.so.49 libicudata.so.49	libpthread.so.0	Threading library
	libdl.so.2	Dynamic loading library
	libstdc++.so.5	Standard C++ library
	libm.so.6	Math library
	libgcc_s.so.1	GCC library

Unicode conversion library	Dependency File	Description
	libc.so.6	C library
	ld-linux.so.2	Linux library

Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_inx	Information concerning how this distribution was created.
	setup.sh	Environment variables setup executable for sh.
	setup.csh	Environment variables setup executable for csh.
mc3apps	comp.c	Sample compression/decompression application.
	ct.img	Example CT image file. (This file is generated by mc3file and can be regenerated by the user, if need be.)
	duplicate.c	Sample for using MC_Standard_Compressor/MC_Standard_De compressor via MC_Duplicate_Message.
	general_util.c	General utilities for all sample programs.
	general_util.h	General utilities for all sample programs.
	inetd_echo_scp.c	Sample DICOM Echo SCP, using the inetd functionality.
	makefile	Makefile for example programs.
	med_fsu.c	Media File Set Updater Application.
	merge.ini	Merge DICOM Toolkit Initialization Configuration File. (Used by all sample applications.)

Directory	File	Description
	mergecom.app	Merge DICOM Toolkit Application Profile Configuration File.
	mergecom.pro	Merge DICOM Toolkit System Profile Configuration File.
	mergecom.srv	Merge DICOM Toolkit Service Profile Configuration File.
	mpeg2dicom.c	Sample for packing/unpacking MPEG2 streams into/from DICOM files.
	prnt_scp.c	Sample Print SCP Application.
	prnt_scu.c	Sample Print SCU Application.
	prnt_svc.h	Sample Print Application header file
	qr.h	Sample Query/Retrieve Application Include file.
	qr_get_scp.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_get_scu.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_scp.c	Sample Query/Retrieve SCP Application.
	qr_scu.c	Sample Query/Retrieve SCU Application.
	qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
	sreport.c	Sample Structured Report Application
	ssl_samp.c	SSL callbacks for SSL SCU and SSL SCP.
	ssl_samp.h	Header file for SSL callbacks for SSL SCU and SSL SCP.
	ssl_scp.c	Sample SCP application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	ssl_scp.h	Header file for sample SSL SCP application.

Directory	File	Description
	ssl_scu.c	Sample SCU application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	stor_scp.c	Sample Storage SCP Application.
	stor_scu.c	Sample Storage SCU Application.
	work.dat	Database flat file, used by the Modality Worklist SCP application.
	workdata.c	Sample database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	workdata.h	Sample header file for database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	work_scp.c	Sample Modality Worklist and Modality Performed Procedure Step SCP Application.
	work_scu.c	Sample Modality Worklist and Modality Performed Procedure Step SCU Application.

Directory	File	Description
mc3bin	genconf	Configuration source file generation utility.
	gendict	Dictionary source file generation utility.
	mc3comp	Compare the values within two DICOM message or file objects.
	mc3conv	Convert a DICOM message or file object into a new transfer syntax.
	mc3dcomb	Runtime Dictionary Combine utility.
	mc3dict	Runtime DICOM Data Dictionary utility.
	mc3echo	DICOM Echo Test Utility.
	mc3file	Generate a DICOM message object.
	mc3icomb	Runtime Info Combine utility.
	mc3info	Runtime Message Database generation utility.
	mc3list	List a DICOM message object.
	mc3valid	Validate a DICOM message object.
mc3doc	Database.pdf	Merge DICOM Toolkit DICOM <i>Database Manual</i> .
	Platform.pdf	This document.
	Refer.pdf	Merge DICOM Toolkit <i>Reference Manual</i> .
	Sample.pdf	Merge DICOM Toolkit <i>Sample Applications Guide</i> .
	User.pdf	Merge DICOM Toolkit <i>User's Manual</i> .
mc3inc	bmp.h	Pegasus include file.
	diction.h	DICOM Data Dictionary macros.
	mc3items.h	Library include file for use with mc3adv.a.
	mc3media.h	Library include file for use with mc3adv.a.
	mc3msg.h	Library include file for use with mc3adv.a.

Directory	File	Description
	mc3services.h	Library include file for use with mc3adv.a.
	mcstatus.h	Library include file for use with mc3adv.a.
	mergecom.h	Library include file for use with mc3adv.a.
	pcd.h	Pegasus include file.
	pic.h	Pegasus include file.
	stdtypes.h	Pegasus include file.
mc3lib	libcudata.so.49	Unicode conversion data.
	libcuc.so.49	Unicode conversion library.
	libxml2.so.2.9.3	XML parsing library.
	libpicl20.so	Pegasus shared object which needs to be linked in when you compile your source.
	mc3adv.a	Merge DICOM Toolkit software static library.
	mc3adv.so	Merge DICOM Toolkit software shared object.
	picn6120.ssm	Pegasus library for JPEG-LS decompression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picn6220.ssm	Pegasus library for JPEG Lossless compression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picn6320.ssm	Pegasus library for JPEG Lossless decompression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picn6420.ssm	Pegasus library for JPEG Lossy compression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.

Directory	File	Description
	picn6520.ssm	Pegasus library for JPEG Lossy decompression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picn6820.ssm	Pegasus library for JPEG 2000 compression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picn6920.ssm	Pegasus library for JPEG 2000 decompression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
mc3msg	default.pfl	Default configuration file used by mc3file.
	diction.pfl	Runtime DICOM Data Dictionary profile.
	message.txt	DICOM message formats for INFO purposes.
	info.pfl	Runtime Message Database profile.
	mc3dcomb.pfl	Sample configuration file for use with mc3dcomb utility.
	mc3icomb.cfg	Sample configuration file for use with mc3icomb utility.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

64-Bit Linux® on Intel® x86-64 (008-91132)

Supported Configurations

The following table describes the system requirements for the Merge DICOM Toolkit.

Category	Requirement
Hardware	Intel x86-64 Pentium processor or AMD64 processor.

Software (base)	Fedora Core 3 or later ² .
Software (development)	GCC 4.0.2.

The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for 64-Bit Fedora Core Linux on Intel x86-64 is provided in two forms: a **shared object** and a **static library**.

Static library

A static library is a collection of subroutines that are callable by your programs. To use them, simply link them with your program.

The compiler flags needed to link with the Merge DICOM Toolkit static library are:

Flag	Type	Description
-ldl	Compile time	Link with dynamic loading library.

Shared object

A shared object is similar to a static library. It contains entry points for your application to use and call and contains code that will be executed by many different modules. The difference, however, is that the code is not included in the executable file built by the linker or loader. Instead, the code is loaded at runtime when the resources are requested. The code is then mapped into the processes address space.

The use of the two types of libraries is exactly the same: they are “linked” into an application program by the system loader after the application has been compiled. The way the system loader constructs the executable is different, however.

When a static library is linked with an application, an executable is produced that contains the code of the application and the code of the library. This is not true with code produced and linked with shared object methods. The application and the shared “library” must be compiled and linked with special compiler flags.

The compiler flags used to generate the Merge DICOM Toolkit static library are:

Flag	Type	Description
-m64	Compile Time	Produce 64 bit code.
-O3	Compile Time	Optimization level.
-D_REENTRANT	Compile Time	Produce reentrant code that is thread safe.

² Merge DICOM Toolkit for Fedora Core Linux has been developed using Fedora Core 3. Although it should function under any distribution of Linux, Fedora Core Linux is the supported version if distribution specific problems arise.

The compiler flags used to generate the Merge DICOM Toolkit shared object are:

Flag	Type	Description
-fPIC	Compile Time	Produced position independent code.
-m64	Compile Time	Produce 64 bit code.
-O3	Compile Time	Optimization level.
-D_REENTRANT	Compile Time	Produce reentrant code that is thread safe.

The compiler flags needed to link with the Merge DICOM Toolkit dynamic library are:

Flag	Type	Description
-ldl	Compile time	Link with dynamic loading library.

Also, the `LD_LIBRARY_PATH` environment variable will need to be modified to contain the path of the `mc3adv.so` file. The following is an excerpt from the man page of `ld` and describes the function of the `LD_LIBRARY_PATH` environment variable:

“`LD_LIBRARY_PATH` is a list of directories in which to search for libraries specified with the `-l` option. Multiple directories are separated by a colon.”
 “It is also used to specify libraries search path to the run-time linker, that is, if `LD_LIBRARY_PATH` exists in the environment, the run-time linker searches the directories named in it, before the default directories for the shared objects to be linked with the program at execution.”

It should be noted that the Merge DICOM Toolkit shared object (`mc3adv.so`) is designed to be “dynamic-safe” and optimized. This means that the shared object is safe when more than one application executed the same code at the same time.

Miscellaneous Notes

Threading Support

The Merge DICOM Toolkit for Fedora Core Linux supports multi-threaded applications. Please see the User’s Manual for details on the limitations for using Merge DICOM Toolkit with multiple threads.

Compression Support

The Merge DICOM Toolkit for Fedora Core Linux supports the PICTools (formerly known as Pegasus) libraries for compression/decompression from Accusoft (formerly Pegasus Imaging).

The Lossless and Lossy JPEG compressors can be utilized within your application without purchasing an additional license from Accusoft. However, the Lossy and Lossless JPEG Pegasus libraries are limited to compress and decompress at a maximum rate of 3 frames per second. This limit can be removed by purchasing a license from Accusoft (www.accusoft.com) and configuring that license in our mergecom.pro configuration file.

Note: To use the JPEG2000 compressor or decompressor in your applications, you must purchase a separate license from Accusoft.

Built in RLE compression/decompression is also available.

Unicode Support

The Merge DICOM Toolkit for 64-bit Linux supports Unicode conversion of DICOM defined character sets with and without code extensions. Two optional shared object libraries, libicuuc.so.49 and libicudata.so.49, are distributed with the toolkit and are used to perform Unicode character set conversion. Users that wish to use Unicode conversion functions must call `MC_Enable_Unicode_Conversion` to initialize the shared object libraries and ensure the dependency files listed in the table below are available at runtime. Existing users that have no plan to use the Unicode conversion functions don't need to deploy the two shared objects and their dependency files.

Dependency files of libicuuc.so.49 and libicudata.so.49 on 64-bit Linux platform:

Unicode conversion library	Dependency File	Description
libicuuc.so.49 libicudata.so.49	libpthread.so.0	Threading library
	libdl.so.2	Dynamic loading library
	libstdc++.so.6	Standard C++ library
	libm.so.6	Math library
	libgcc_s.so.1	GCC library
	libc.so.6	C library
	ld-linux-x86-64.so.2	Linux library

Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_Inx_64	Information concerning how this distribution was created.
	setup.sh	Environment variables setup executable for sh.
	setup.csh	Environment variables setup executable for csh.
mc3apps	comp.c	Sample compression/decompression application.
	ct.img	Example CT image file. (This file is generated by mc3file and can be regenerated by the user, if need be.)
	duplicate.c	Sample for using MC_Standard_Compressor/MC_Standard_De compressor via MC_Duplicate_Message.
	general_util.c	General utilities for all sample programs.
	general_util.h	General utilities for all sample programs.
	inetd_echo_scp.c	Sample DICOM Echo SCP, using the inetd functionality.
	makefile	Makefile for example programs.
	med_fsu.c	Media File Set Updater Application.
	merge.ini	Merge DICOM Toolkit Initialization Configuration File. (Used by all sample applications.)
	mergecom.app	Merge DICOM Toolkit Application Profile Configuration File.
	mergecom.pro	Merge DICOM Toolkit System Profile Configuration File.
	mergecom.srv	Merge DICOM Toolkit Service Profile Configuration File.

Directory	File	Description
	mpeg2dicom.c	Sample for packing/unpacking MPEG2 streams into/from DICOM files.
	prnt_scp.c	Sample Print SCP Application.
	prnt_scu.c	Sample Print SCU Application.
	prnt_svc.h	Sample Print Application header file.
	qr.h	Sample Query/Retrieve Application Include file.
	qr_get_scp.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_get_scu.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_scp.c	Sample Query/Retrieve SCP Application.
	qr_scu.c	Sample Query/Retrieve SCU Application.
	qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
	sreport.c	Sample Structured Report Application
	ssl_samp.c	SSL callbacks for SSL SCU and SSL SCP.
	ssl_samp.h	Header file for SSL callbacks for SSL SCU and SSL SCP.
	ssl_scp.c	Sample SCP application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	ssl_scp.h	Header file for sample SSL SCP application.
	ssl_scu.c	Sample SCU application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	stor_scp.c	Sample Storage SCP Application.
	stor_scu.c	Sample Storage SCU Application.

Directory	File	Description
	work.dat	Database flat file, used by the Modality Worklist SCP application.
	workdata.c	Sample database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	workdata.h	Sample header file for database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	work_scp.c	Sample Modality Worklist and Modality Performed Procedure Step SCP Application.
	work_scu.c	Sample Modality Worklist and Modality Performed Procedure Step SCU Application.
mc3bin	genconf	Configuration source file generation utility.
	gendict	Dictionary source file generation utility.
	mc3comp	Compare the values within two DICOM message or file objects.
	mc3conv	Convert a DICOM message or file object into a new transfer syntax.
	mc3dcomb	Runtime Dictionary Combine utility.
	mc3dict	Runtime DICOM Data Dictionary utility.
	mc3echo	DICOM Echo Test Utility.
	mc3file	Generate a DICOM message object.
	mc3icomb	Runtime Info Combine utility.
	mc3info	Runtime Message Database generation utility.
	mc3list	List a DICOM message object.
	mc3valid	Validate a DICOM message object.
mc3doc	Database.pdf	Merge DICOM Toolkit DICOM <i>Database Manual</i> .
	Platform.pdf	This document.
	Refer.pdf	Merge DICOM Toolkit <i>Reference Manual</i> .

Directory	File	Description
	Sample.pdf	Merge DICOM Toolkit <i>Sample Applications Guide</i> .
	User.pdf	Merge DICOM Toolkit <i>User's Manual</i> .
mc3inc	bmp.h	Pegasus include file.
	diction.h	DICOM Data Dictionary macros.
	mc3items.h	Library include file for use with mc3adv.a.
	mc3media.h	Library include file for use with mc3adv.a.
	mc3msg.h	Library include file for use with mc3adv.a.
	mc3services.h	Library include file for use with mc3adv.a.
	mcstatus.h	Library include file for use with mc3adv.a.
	mergecom.h	Library include file for use with mc3adv.a.
	pcd.h	Pegasus include file.
	pic.h	Pegasus include file.
	stdtypes.h	Pegasus include file.
mc3lib	libicudata.so.49	Unicode conversion data.
	libicuuc.so.49	Unicode conversion library.
	libxml2.so.2.9.3	XML parsing library
	libpiclx20.so	Pegasus shared object which needs to be linked in when you compile your source.
	mc3adv.a	Merge DICOM Toolkit software static library.
	mc3adv.so	Merge DICOM Toolkit software shared object.
	picx6120.ssm	Pegasus library for JPEG-LS decompression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.

Directory	File	Description
	picx6220.ssm	Pegasus library for JPEG Lossless compression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picx6320.ssm	Pegasus library for JPEG Lossless decompression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picx6420.ssm	Pegasus library for JPEG Lossy compression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picx6520.ssm	Pegasus library for JPEG Lossy decompression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picx6820.ssm	Pegasus library for JPEG 2000 compression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picx6920.ssm	Pegasus library for JPEG 2000 decompression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
mc3msg	default.pfl	Default configuration file used by mc3file.
	diction.pfl	Runtime DICOM Data Dictionary profile.
	message.txt	DICOM message formats for INFO purposes.
	info.pfl	Runtime Message Database profile.
	mc3dcomb.pfl	Sample configuration file for use with mc3dcomb utility.
	mc3icomb.cfg	Sample configuration file for use with mc3icomb utility.

Directory	File	Description
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

32-Bit Solaris™ 10 on Intel® x86 (008-91117)

Supported Configurations

The following table describes the system requirements for the Merge DICOM Toolkit.

Category	Requirement
Hardware	Solaris 10 on Intel x86 supported Ethernet Network Interface Card.
Software (base)	Solaris 10 Operating System.
Software (development)	GCC compiler.

The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for 32-Bit Solaris™ 10 on Intel® x86 is provided in two forms: a shared object and a static library.

Static library

A static library is a collection of subroutines that are callable by your programs. To use them, simply link them with your program.

The compiler flags needed to link with the Merge DICOM Toolkit static library are:

Flag	Type	Description
-O3	Compile time	Optimization level.
-D_REENTRANT	Compile time	This option specifies that the library is reentrant and can be used in a multi-threaded environment.
-ldl	Compile time	Link with dynamic loading library.

Shared objects

A shared object is similar to a static library. It contains entry points for your application to use and call and contains code that will be executed by many different modules. The difference, however, is that the code is not included in the executable file built by the linker or loader. Instead, the code is loaded at runtime when the resources are requested. The code is then mapped into the processes address space.

The use of the two types of libraries is exactly the same: they are “linked” into an application program by the system loader after the application has been compiled. The way the system loader constructs the executable is different, however.

When a static library is linked with an application, an executable is produced that contains the code of the application and the code of the library. This is not true with code produced and linked with shared object methods. The application and the shared “library” must be compiled and linked with special compiler flags.

The compiler flags needed to link with the Merge DICOM Toolkit shared object are:

Flag	Type	Description
-O3	Compile time	Optimization level.
-fPIC	Compile time	Produced position independent code.
-D_REENTRANT	Compile time	This option specifies that the library is reentrant and can be used in a multi-threaded environment.
-ldl	Compile time	Link with dynamic loading library.

Also, the `LD_LIBRARY_PATH` environment variable will need to be modified to contain the path of the `mc3adv.so` file. The following is an excerpt from the man page of `ld` and describes the function of the `LD_LIBRARY_PATH` environment variable:

“`LD_LIBRARY_PATH` is a list of directories in which to search for libraries specified with the `-l` option. Multiple directories are separated by a colon.”
 “It is also used to specify libraries search path to the run-time linker, that is, if `LD_LIBRARY_PATH` exists in the environment, the run-time linker searches the directories named in it, before the default directories for the shared objects to be linked with the program at execution.”

It should be noted that the Merge DICOM Toolkit shared object (`mc3adv.so`) is designed to be “dynamic-safe” and optimized. This means that the shared object is safe when more than one application executed the same code at the same time.

Miscellaneous Notes

Threading Support

The Merge DICOM Toolkit for Solaris on Intel x86 supports multi-threaded applications. Please see the *User’s Manual* for details on the limitations for using Merge DICOM Toolkit with multiple threads.

Compression Support

The Merge DICOM Toolkit for Solaris on Intel x86 now supports the PICTools (formerly known as Pegasus) libraries for compression/decompression from Accusoft (formerly Pegasus Imaging).

The Lossless and Lossy JPEG compressors can be utilized within your application without purchasing an additional license from Accusoft. However, the Lossy and Lossless JPEG Pegasus libraries are limited to compress and decompress at a maximum rate of 3 frames per second. This limit can be removed by purchasing a license from Accusoft (www.accusoft.com) and configuring that license in our mergecom.pro configuration file.

Note: To use the JPEG2000 compressor or decompressor in your applications, you must purchase a separate license from Accusoft.

Built in RLE compression/decompression is also available.

Unicode Support

The Merge DICOM Toolkit for Solaris on Intel x86 supports Unicode conversion of DICOM defined character sets with and without code extensions. Two optional shared object libraries, libicuuc.so.49 and libicudata.so.49, are distributed with the toolkit and are used to perform Unicode character set conversion. Users that wish to use Unicode conversion functions must call MC_Enable_Unicode_Conversion to initialize the shared object libraries and ensure the dependency files listed in the table below are available at runtime. Existing users that have no plan to use the Unicode conversion functions don't need to deploy the two shared objects and their dependency files.

Dependency files of libicuuc.so.49 and libicudata.so.49 for Solaris on Intel x86 platform:

Unicode conversion library	Dependency File	Description
libicuuc.so.49 libicudata.so.49	libpthread.so.1	Threading library
	libc.so.1	C library
	libstdc++.so.6	Standard C++ library
	libm.so.2	Math library
	libgcc_s.so.1	GCC library

Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_x86	Information concerning how this distribution was created.
	setup.sh	Environment variables setup executable for sh.
	setup.csh	Environment variables setup executable for csh.
mc3apps	comp.c	Sample compression/decompression application.
	ct.img	Example CT image file. (This file is generated by mc3file and can be regenerated by the user, if need be.)
	duplicate.c	Sample for using MC_Standard_Compressor/MC_Standard_De compressor via MC_Duplicate_Message.
	general_util.c	General utilities for all sample programs.
	general_util.h	General utilities for all sample programs.
	inetd_echo_scp.c	Sample DICOM Echo SCP, using the inetd functionality.
	makefile	Makefile for example programs.
	med_fsu.c	Media File Set Updater Application.
	merge.ini	Merge DICOM Toolkit Initialization Configuration File. (Used by all sample applications.)
	mergecom.app	Merge DICOM Toolkit Application Profile Configuration File.
	mergecom.pro	Merge DICOM Toolkit System Profile Configuration File.
	mergecom.srv	Merge DICOM Toolkit Service Profile Configuration File.

Directory	File	Description
	mpeg2dicom.c	Sample for packing/unpacking MPEG2 streams into/from DICOM files.
	prnt_scp.c	Sample Print SCP Application.
	prnt_scu.c	Sample Print SCU Application.
	prnt_svc.h	Sample Print Application header file.
	qr.h	Sample Query/Retrieve Application Include file.
	qr_get_scp.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_get_scu.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_scp.c	Sample Query/Retrieve SCP Application.
	qr_scu.c	Sample Query/Retrieve SCU Application.
	qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
	sreport.c	Sample Structured Report Application
	ssl_samp.c	SSL callbacks for SSL SCU and SSL SCP.
	ssl_samp.h	Header file for SSL callbacks for SSL SCU and SSL SCP.
	ssl_scp.c	Sample SCP application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	ssl_scp.h	Header file for sample SSL SCP application.
	ssl_scu.c	Sample SCU application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	stor_scp.c	Sample Storage SCP Application.
	stor_scu.c	Sample Storage SCU Application.

Directory	File	Description
	work.dat	Database flat file, used by the Modality Worklist SCP application.
	workdata.c	Sample database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	workdata.h	Sample header file for database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	work_scp.c	Sample Modality Worklist and Modality Performed Procedure Step SCP Application.
	work_scu.c	Sample Modality Worklist and Modality Performed Procedure Step SCU Application.
mc3bin	genconf	Configuration source file generation utility.
	gendict	Dictionary source file generation utility.
	mc3comp	Compare the values within two DICOM message or file objects.
	mc3conv	Convert a DICOM message or file object into a new transfer syntax.
	mc3dcomb	Runtime Dictionary Combine utility.
	mc3dict	Runtime DICOM Data Dictionary utility.
	mc3echo	DICOM Echo Test Utility.
	mc3file	Generate a DICOM message object.
	mc3icomb	Runtime Info Combine utility.
	mc3info	Runtime Message Database generation utility.
	mc3list	List a DICOM message object.
	mc3valid	Validate a DICOM message object.
mc3doc	Database.pdf	Merge DICOM Toolkit DICOM <i>Database Manual</i> .
	Platform.pdf	This document.
	Refer.pdf	Merge DICOM Toolkit <i>Reference Manual</i> .

Directory	File	Description
	Sample.pdf	Merge DICOM Toolkit <i>Sample Applications Guide</i> .
	User.pdf	Merge DICOM Toolkit <i>User's Manual</i> .
mc3inc	bmp.h	Pegasus include file.
	diction.h	DICOM Data Dictionary macros.
	mc3items.h	Library include file for use with mc3adv.a.
	mc3media.h	Library include file for use with mc3adv.a.
	mc3msg.h	Library include file for use with mc3adv.a.
	mc3services.h	Library include file for use with mc3adv.a.
	mcstatus.h	Library include file for use with mc3adv.a.
	mergecom.h	Library include file for use with mc3adv.a.
	pcd.h	Pegasus include file.
	pic.h	Pegasus include file.
	stdtypes.h	Pegasus include file.
mc3lib	libcudata.so.49	Unicode conversion data.
	libcuuc.so.49	Unicode conversion library.
	libpics20.so	Pegasus shared object which needs to be linked in when you compile your source.
	libxml2.so.2.9.3	XML parsing library.
	mc3adv.a	Merge DICOM Toolkit software static library.
	mc3adv.so	Merge DICOM Toolkit software shared object.
	picn6120.ssm	Pegasus library for JPEG-LS decompression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picn6220.ssm	Pegasus library for JPEG Lossless compression. This file must be copied to the

Directory	File	Description
		/usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picn6320.ssm	Pegasus library for JPEG Lossless decompression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picn6420.ssm	Pegasus library for JPEG Lossy compression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picn6520.ssm	Pegasus library for JPEG Lossy decompression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picn6820.ssm	Pegasus library for JPEG 2000 compression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
	picn6920.ssm	Pegasus library for JPEG 2000 decompression. This file must be copied to the /usr/local/lib/pegasus directory, or must be in a directory pointed to by the SSMPATH environment variable.
mc3msg	default.pfl	Default configuration file used by mc3file.
	diction.pfl	Runtime DICOM Data Dictionary profile.
	message.txt	DICOM message formats for INFO purposes.
	info.pfl	Runtime Message Database profile.
	mc3dcomb.pfl	Sample configuration file for use with mc3dcomb utility.
	mc3icomb.cfg	Sample configuration file for use with mc3icomb utility.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

64-Bit Solaris™ 10 on Intel® x64 (008-91116)

Supported Configurations

The following table describes the system requirements for the Merge DICOM Toolkit.

Category	Requirement
Hardware	Solaris 10 on 64-bit Intel x86 supported Ethernet Network Interface Card.
Software (base)	64-bit Solaris 10 Operating System.
Software (development)	gcc compiler.

The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for 64-Bit Solaris™ 10 on Intel® x64 is provided in two forms: a shared object and a static library.

Static library

A static library is a collection of subroutines that are callable by your programs. To use them, simply link them with your program.

The compiler flags needed to link with the Merge DICOM Toolkit static library are:

Flag	Type	Description
-O3	Compile time	Optimization level.
-D_REENTRANT	Compile time	This option specifies that the library is reentrant and can be used in a multi-threaded environment.
-ldl	Compile time	Link with dynamic loading library.

Shared object

A shared object is similar to a static library. It contains entry points for your application to use and call and contains code that will be executed by many different modules. The difference, however, is that the code is not included in the executable file built by the linker or loader. Instead, the code is loaded at runtime when the resources are requested. The code is then mapped into the processes address space.

The usage of the two types of libraries is exactly the same: they are “linked” into an application program by the system loader after the application has been compiled. The way the system loader constructs the executable is different, however.

When a static library is linked with an application, an executable is produced that contains the code of the application and the code of the library. This is not true

with code produced and linked with shared object methods. The application and the shared “library” must be compiled and linked with special compiler flags.

The compiler flags needed to link with the Merge DICOM Toolkit shared object are:

Flag	Type	Description
-O3	Compile time	Optimization level.
-fPIC	Compile time	Produced position independent code.
-D_REENTRANT	Compile time	This option specifies that the library is reentrant and can be used in a multi-threaded environment.
-ldl	Compile time	Link with dynamic loading library.

Also, the `LD_LIBRARY_PATH` environment variable will need to be modified to contain the path of the `mc3adv.so` file. The following is an excerpt from the `man` page of `ld` and describes the function of the `LD_LIBRARY_PATH` environment variable:

“`LD_LIBRARY_PATH` is a list of directories in which to search for libraries specified with the `-l` option. Multiple directories are separated by a colon.”
 “It is also used to specify libraries search path to the run-time linker, that is, if `LD_LIBRARY_PATH` exists in the environment, the run-time linker searches the directories named in it, before the default directories for the shared objects to be linked with the program at execution.”

It should be noted that the Merge DICOM Toolkit shared object (`mc3adv.so`) is designed to be “dynamic-safe” and optimized. This means that the shared object is safe when more than one application executed the same code at the same time.

Miscellaneous Notes

Threading Support

The Merge DICOM Toolkit for Solaris on Intel x64 supports multi-threaded applications. Please see the *User’s Manual* for details on the limitations for using Merge DICOM Toolkit with multiple threads.

Compression Support

The Merge DICOM Toolkit for Solaris on Intel x64 does not support the Pegasus compression/decompression libraries.

Built in RLE compression/decompression is available.

Unicode Support

The Merge DICOM Toolkit for Solaris on Intel x64 supports Unicode conversion of DICOM defined character sets with and without code extensions. Two optional

shared object libraries, libicuuc.so.49 and libicudata.so.49, are distributed with the toolkit and are used to perform Unicode character set conversion. Users that wish to use Unicode conversion functions must call `MC_Enable_Unicode_Conversion` to initialize the shared object libraries and ensure the dependency files listed in the table below are available at runtime. Existing users that have no plan to use the Unicode conversion functions don't need to deploy the two shared objects and their dependency files.

Dependency files of libicuuc.so.49 and libicudata.so.49 for Solaris on Intel x64 platform:

Unicode conversion library	Dependency File	Description
libicuuc.so.49 libicudata.so.49	libpthread.so.1	Threading library
	libc.so.1	C library
	libstdc++.so.6	Standard C++ library
	libm.so.2	Math library
	libgcc_s.so.1	GCC library

Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_x64	Information concerning how this distribution was created.
	setup.sh	Environment variables setup executable for sh.
	setup.csh	Environment variables setup executable for csh.
mc3apps	comp.c	Sample compression/decompression application.
	ct.img	Example CT image file. (This file is generated by mc3file and can be regenerated by the user, if need be.)

Directory	File	Description
	duplicate.c	Sample for using MC_Standard_Compressor/MC_Standard_De compressor via MC_Duplicate_Message.
	general_util.c	General utilities for all sample programs.
	general_util.h	General utilities for all sample programs.
	inetd_echo_scp.c	Sample DICOM Echo SCP, using the inetd functionality.
	makefile	Makefile for example programs.
	med_fsu.c	Media File Set Updater Application.
	merge.ini	Merge DICOM Toolkit Initialization Configuration File. (Used by all sample applications.)
	mergecom.app	Merge DICOM Toolkit Application Profile Configuration File.
	mergecom.pro	Merge DICOM Toolkit System Profile Configuration File.
	mergecom.srv	Merge DICOM Toolkit Service Profile Configuration File.
	mpeg2dicom.c	Sample for packing/unpacking MPEG2 streams into/from DICOM files.
	prnt_scp.c	Sample Print SCP Application.
	prnt_scu.c	Sample Print SCU Application.
	prnt_svc.h	Sample Print Application header file.
	qr.h	Sample Query/Retrieve Application Include file.
	qr_get_scp.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_get_scu.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_scp.c	Sample Query/Retrieve SCP Application.
	qr_scu.c	Sample Query/Retrieve SCU Application.

Directory	File	Description
	qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
	sreport.c	Sample Structured Report Application
	ssl_samp.c	SSL callbacks for SSL SCU and SSL SCP.
	ssl_samp.h	Header file for SSL callbacks for SSL SCU and SSL SCP.
	ssl_scp.c	Sample SCP application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	ssl_scp.h	Header file for sample SSL SCP application.
	ssl_scu.c	Sample SCU application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	stor_scp.c	Sample Storage SCP Application.
	stor_scu.c	Sample Storage SCU Application.
	work.dat	Database flat file, used by the Modality Worklist SCP application.
	workdata.c	Sample database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	workdata.h	Sample header file for database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	work_scp.c	Sample Modality Worklist and Modality Performed Procedure Step SCP Application.
	work_scu.c	Sample Modality Worklist and Modality Performed Procedure Step SCU Application.

Directory	File	Description
mc3bin	genconf	Configuration source file generation utility.
	gendict	Dictionary source file generation utility.
	mc3comp	Compare the values within two DICOM message or file objects.
	mc3conv	Convert a DICOM message or file object into a new transfer syntax.
	mc3dcomb	Runtime Dictionary Combine utility.
	mc3dict	Runtime DICOM Data Dictionary utility.
	mc3echo	DICOM Echo Test utility.
	mc3file	Generate a DICOM message object.
	mc3icomb	Runtime Info Combine utility.
	mc3info	Runtime Message Database generation utility.
	mc3list	List a DICOM message object.
	mc3valid	Validate a DICOM message object.
mc3doc	Database.pdf	Merge DICOM Toolkit DICOM <i>Database Manual</i> .
	Platform.pdf	This document.
	Refer.pdf	Merge DICOM Toolkit <i>Reference Manual</i> .
	Sample.pdf	Merge DICOM Toolkit <i>Sample Applications Guide</i> .
	User.pdf	Merge DICOM Toolkit <i>User's Manual</i> .
mc3inc	diction.h	DICOM Data Dictionary macros.
	mc3items.h	Library include file for use with mc3adv.a.
	mc3media.h	Library include file for use with mc3adv.a.
	mc3msg.h	Library include file for use with mc3adv.a.
	mc3services.h	Library include file for use with mc3adv.a.

Directory	File	Description
	mcstatus.h	Library include file for use with mc3adv.a.
	mergecom.h	Library include file for use with mc3adv.a.
mc3lib	libcudata.so.49	Unicode conversion data.
	libcucuc.so.49	Unicode conversion library.
	libxml2.so.2.9.3	XML parsing library.
	mc3adv.a	Merge DICOM Toolkit software static library.
	mc3adv.so	Merge DICOM Toolkit software shared object.
mc3msg	default.pfl	Default configuration file used by mc3file.
	diction.pfl	Runtime DICOM Data Dictionary profile.
	message.txt	DICOM message formats for INFO purposes.
	info.pfl	Runtime Message Database profile.
	mc3dcomb.pfl	Sample configuration file for use with mc3dcomb utility.
	mc3icomb.cfg	Sample configuration file for use with mc3icomb utility.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

32-Bit Solaris™ 8 on SPARC® using Sun™ Compiler (008-91119)

Supported Configurations

The following table describes the system requirements for the Merge DICOM Toolkit.

Category	Requirement
Hardware	Solaris supported Ethernet Network Interface Card.
Software (base)	Solaris 8 Operating System.

Software (development)	Sun Studio C Compiler v10.
------------------------	----------------------------

The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for 32-Bit Solaris 8 on SPARC using Sun Compiler is provided in two forms: a **shared object** and a **static library**.

Static library

A static library is a collection of subroutines that are callable by your programs. To use them, simply link them with your program.

The compiler flags needed to link with the Merge DICOM Toolkit static library are:

Flag	Type	Description
-Xa	Compile time	Specifies ANSI C conformance.
-misalign	Compile time	Handle misaligned data.
-xO3	Compile time	Optimization enabled.
-ldl	Compile time	Link with dynamic loading library.

Shared object

A shared object is similar to a static library. It contains entry points for your application to use and call and contains code that will be executed by many different modules. The difference, however, is that the code is not included in the executable file built by the linker or loader. Instead, the code is loaded at runtime when the resources are requested. The code is then mapped into the processes address space.

The use of the two types of libraries is exactly the same: they are “linked” into an application program by the system loader after the application has been compiled. The way the system loader constructs the executable is different, however.

When a static library is linked with an application, an executable is produced that contains the code of the application and the code of the library. This is not true with code produced and linked with shared object methods. The application and the shared “library” must be compiled and linked with special compiler flags.

The compiler flags needed to link with the Merge DICOM Toolkit shared object are:

Flag	Type	Description
-Xa	Compile time	Specifies ANSI C conformance.
-KPIC	Compile time	Produced position independent code.
-misalign	Compile time	Handle misaligned data.

-xO3	Compile time	Optimization enabled.
-ldl	Compile time	Link with dynamic loading library.

Also, the `LD_LIBRARY_PATH` environment variable will need to be modified to contain the path of the `mc3adv.so` file. The following is an excerpt from the `man` page of `ld` and describes the function of the `LD_LIBRARY_PATH` environment variable:

“`LD_LIBRARY_PATH` is a list of directories in which to search for libraries specified with the `-l` option. Multiple directories are separated by a colon.”
 “It is also used to specify libraries search path to the run-time linker, that is, if `LD_LIBRARY_PATH` exists in the environment, the run-time linker searches the directories named in it, before the default directories for the shared objects to be linked with the program at execution.”

It should be noted that the Merge DICOM Toolkit shared object (`mc3adv.so`) is designed to be “dynamic-safe” and optimized. This means that the shared object is safe when more than one application executed the same code at the same time.

Miscellaneous Notes

Threading Support

The Merge DICOM Toolkit for Solaris supports multi-threaded applications. Please see the *User's Manual* for details on the limitations for using Merge DICOM Toolkit with multiple threads.

Compression Support

The Merge DICOM Toolkit for Solaris 8 supports the PICTools (formerly known as Pegasus) libraries for compression/decompression from Accusoft (formerly Pegasus Imaging).

Previous versions of this Merge DICOM Toolkit distributed two versions of the library: one that supported the Pegasus libraries and another that did not. Starting with the 3.7.0 release, only one version of the library is distributed that supports the Pegasus libraries.

The Lossless and Lossy JPEG compressors can be utilized within your application without purchasing an additional license from Accusoft. However, the Lossy and Lossless JPEG Pegasus libraries are limited to compress and decompress at a maximum rate of 3 frames per second. This limit can be removed by purchasing a license from Accusoft (www.accusoft.com) and configuring that license in our `mergecom.pro` configuration file.

Note: To use the JPEG2000 compressor or decompressor in your applications, you must purchase a separate license from Accusoft.

Unicode Support

The Merge DICOM Toolkit for Solaris on Sparc using Sun compiler supports Unicode conversion of DICOM defined character sets with and without code

extensions. Two optional shared object libraries, libicuuc.so.49 and libicudata.so.49, are distributed with the toolkit and are used to perform Unicode character set conversion. Users that wish to use Unicode conversion functions must call `MC_Enable_Unicode_Conversion` to initialize the shared object libraries and ensure the dependency files listed in the table below are available at runtime. Existing users that have no plan to use the Unicode conversion functions don't need to deploy the two shared objects and their dependency files.

Dependency files of libicuuc.so.49 and libicudata.so.49 for Solaris on Sparc using Sun compiler platform:

Unicode conversion library	Dependency File	Description
libicuuc.so.49 libicudata.so.49	libpthread.so.1	Threading library
	libc.so.1	C library
	libCrun.so.1	C runtime library
	libm.so.1	Math library
	libdl.so.1	Dynamic loading library
	libthread.so.1	Threading library
	libc_psr.so.1	SUN library

Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_sol	Information concerning how this distribution was created.
	setup.sh	Environment variables setup executable for sh.
	setup.csh	Environment variables setup executable for csh.
mc3apps	comp.c	Sample compression/decompression application.
	ct.img	Example CT image file. (This file is generated by mc3file and can be regenerated by the user, if need be.)
	duplicate.c	Sample for using MC_Standard_Compressor/MC_Standard_De compressor via MC_Duplicate_Message.
	general_util.c	General utilities for all sample programs.
	general_util.h	General utilities for all sample programs.
	inetd_echo_scp.c	Sample DICOM Echo SCP, using the inetd functionality.
	makefile	Makefile for example programs.
	med_fsu.c	Media File Set Updater Application.
	merge.ini	Merge DICOM Toolkit Initialization Configuration File. (Used by all sample applications.)
	mergecom.app	Merge DICOM Toolkit Application Profile Configuration File.
	mergecom.pro	Merge DICOM Toolkit System Profile Configuration File.
	mergecom.srv	Merge DICOM Toolkit Service Profile Configuration File.
	mpeg2dicom.c	Sample for packing/unpacking MPEG2 streams into/from DICOM files.
prnt_scp.c	Sample Print SCP Application.	

prnt_scu.c	Sample Print SCU Application.
prnt_svc.h	Sample Print Application header file.
qr.h	Sample Query/Retrieve Application Include file.
qr_get_scp.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
qr_get_scu.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
qr_scp.c	Sample Query/Retrieve SCP Application.
qr_scu.c	Sample Query/Retrieve SCU Application.
qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
sreport.c	Sample Structured Report Application
ssl_samp.c	SSL callbacks for SSL SCU and SSL SCP.
ssl_samp.h	Header file for SSL callbacks for SSL SCU and SSL SCP.
ssl_scp.c	Sample SCP application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
ssl_scp.h	Header file for sample SSL SCP application.
ssl_scu.c	Sample SCU application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
stor_scp.c	Sample Storage SCP Application.
stor_scu.c	Sample Storage SCU Application.
work.dat	Database flat file, used by the Modality Worklist SCP application.
workdata.c	Sample database functions used by Modality Worklist and Performed Procedure Step SCP Application.

	workdata.h	Sample header file for database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	work_scp.c	Sample Modality Worklist and Modality Performed Procedure Step SCP Application.
	work_scu.c	Sample Modality Worklist and Modality Performed Procedure Step SCU Application.
mc3bin	genconf	Configuration source file generation utility.
	gendict	Dictionary source file generation utility.
	mc3comp	Compare the values within two DICOM message or file objects.
	mc3conv	Convert a DICOM message or file object into a new transfer syntax.
	mc3dcomb	Runtime Dictionary Combine utility.
	mc3dict	Runtime DICOM Data Dictionary utility.
	mc3echo	DICOM Echo Test utility.
	mc3file	Generate a DICOM message object.
	mc3icomb	Runtime Info Combine utility.
	mc3info	Runtime Message Database generation utility.
	mc3list	List a DICOM message object.
	mc3valid	Validate a DICOM message object.
mc3doc	Database.pdf	Merge DICOM Toolkit DICOM <i>Database Manual</i> .
	Platform.pdf	This document.
	Refer.pdf	Merge DICOM Toolkit <i>Reference Manual</i> .
	Sample.pdf	Merge DICOM Toolkit <i>Sample Applications Guide</i> .
	User.pdf	Merge DICOM Toolkit <i>User's Manual</i> .
mc3inc	bmp.h	Pegasus include file.
	diction.h	DICOM Data Dictionary macros.

	mc3items.h	Library include file for use with mc3adv.a.
	mc3media.h	Library include file for use with mc3adv.a.
	mc3msg.h	Library include file for use with mc3adv.a.
	mc3services.h	Library include file for use with mc3adv.a.
	mcstatus.h	Library include file for use with mc3adv.a.
	mergecom.h	Library include file for use with mc3adv.a.
	pcd.h	Pegasus include file.
	pic.h	Pegasus include file.
	stdtypes.h	Pegasus include file.
mc3lib	libcudata.so.49	Unicode conversion data
	libicuuc.so.49	Unicode conversion library
	libxml2.so.2.9.3	XML parsing library
	libpicu20.so	Pegasus shared object which needs to be linked in when you compile your source.
	mc3adv.a	Merge DICOM Toolkit software static library.
	mc3adv.so	Merge DICOM Toolkit software shared object.
	picu6120.so	Pegasus shared library for JPEG-LS decompression.
	picu6220.so	Pegasus shared library for JPEG Lossless compression.
	picu6320.so	Pegasus shared library for JPEG Lossless decompression.
	picu6420.so	Pegasus shared library for JPEG Lossy compression.
	picu6520.so	Pegasus shared library for JPEG Lossy decompression.
	picu6820.so	Pegasus shared library for JPEG 2000 compression.

	picu6920.so	Pegasus shared library for JPEG 2000 decompression.
mc3msg	default.pfl	Default configuration file used by mc3file.
	diction.pfl	Runtime DICOM Data Dictionary profile.
	message.txt	DICOM message formats for INFO purposes.
	info.pfl	Runtime Message Database profile.
	mc3dcomb.pfl	Sample configuration file for use with mc3dcomb utility.
	mc3icomb.cfg	Sample configuration file for use with mc3icomb utility.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

32-Bit Solaris™ 8 on SPARC® using GCC Compiler (008-91130)

Supported Configurations

The following table describes the system requirements for the Merge DICOM Toolkit.

Category	Requirement
Hardware	Solaris supported Ethernet Network Interface Card.
Software (base)	Solaris 8 Operating System.
Software (development)	gcc 3.4.6

The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for 32-Bit Solaris 8 on SPARC using GCC Compiler is provided in two forms: a **shared object** and a **static library**.

Static library

A static library is a collection of subroutines that are callable by your programs. To use them, simply link them with your program.

The compiler flags needed to link with the Merge DICOM Toolkit static library are:

Flag	Type	Description
-O3	Compile time	Optimization level.
-ldl	Compile time	Link with dynamic loading library.

Shared object

A shared object is similar to a static library. It contains entry points for your application to use and call and contains code that will be executed by many different modules. The difference, however, is that the code is not included in the executable file built by the linker or loader. Instead, the code is loaded at runtime when the resources are requested. The code is then mapped into the processes address space.

The use of the two types of libraries is exactly the same: they are “linked” into an application program by the system loader after the application has been compiled. The way the system loader constructs the executable is different, however.

When a static library is linked with an application, an executable is produced that contains the code of the application and the code of the library. This is not true with code produced and linked with shared object methods. The application and the shared “library” must be compiled and linked with special compiler flags.

The compiler flags needed to link with the Merge DICOM Toolkit shared object are:

Flag	Type	Description
-O3	Compile time	Optimization level
-fPIC	Compile Time	Produced position independent code
-ldl	Compile time	Link with dynamic loading library

Also, the `LD_LIBRARY_PATH` environment variable will need to be modified to contain the path of the `mc3adv.so` file. The following is an excerpt from the `man` page of `ld` and describes the function of the `LD_LIBRARY_PATH` environment variable:

“`LD_LIBRARY_PATH` is a list of directories in which to search for libraries specified with the `-l` option. Multiple directories are separated by a colon.”
 “It is also used to specify libraries search path to the run-time linker, that is, if `LD_LIBRARY_PATH` exists in the environment, the run-time linker searches the directories named in it, before the default directories for the shared objects to be linked with the program at execution.”

It should be noted that the Merge DICOM Toolkit shared object (`mc3adv.so`) is designed to be “dynamic-safe” and optimized. This means that the shared object is safe when more than one application executed the same code at the same time.

Miscellaneous Notes

Threading Support

The Merge DICOM Toolkit for Solaris 8 supports multi-threaded applications. Please see the *User's Manual* for details on the limitations for using Merge DICOM Toolkit with multiple threads.

Compression Support

The Merge DICOM Toolkit for Solaris 8 supports the PICTools (formerly known as Pegasus) libraries for compression/decompression from Accusoft (formerly Pegasus Imaging).

Previous versions of this Merge DICOM Toolkit distributed two versions of the library: one that supported the Pegasus libraries and another that did not. Starting with the 3.7.0 release, only one version of the library is distributed that supports the Pegasus libraries.

The Lossless and Lossy JPEG compressors can be utilized within your application without purchasing an additional license from Accusoft . However, the Lossy and Lossless JPEG Pegasus libraries are limited to compress and decompress at a maximum rate of 3 frames per second. This limit can be removed by purchasing a license from Accusoft (www.accusoft.com) and configuring that license in our mergecom.pro configuration file.

Note: To use the JPEG2000 compressor or decompressor in your applications, you must purchase a separate license from Accusoft .

Unicode Support

The Merge DICOM Toolkit for Solaris on Sparc using GCC compiler supports Unicode conversion of DICOM defined character sets with and without code extensions. Two optional shared object libraries, libicuuc.so.49 and libicudata.so.49, are distributed with the toolkit and are used to perform Unicode character set conversion. Users that wish to use Unicode conversion functions must call MC_Enable_Unicode_Conversion to initialize the shared object libraries and ensure the dependency files listed in the table below are available at runtime. Existing users that have no plan to use the Unicode conversion functions don't need to deploy the two shared objects and their dependency files.

Dependency files of libicuuc.so.49 and libicudata.so.49 for Solaris on Sparc using GCC compiler platform:

Unicode conversion library	Dependency File	Description
libicuuc.so.49 libicudata.so.49	libpthread.so.1	Threading library
	libc.so.1	C library
	libstdc++.so.6	Standard C++ library

Unicode conversion library	Dependency File	Description
	libm.so.1	Math library
	libgcc_s.so.1	GCC library
	libdl.so.1	Dynamic loading library
	libthread.so.1	Threading library
	libc_psr.so.1	SUN library

Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_sol8	Information concerning how this distribution was created.
	setup.sh	Environment variables setup executable for sh.
	setup.csh	Environment variables setup executable for csh.
mc3apps	comp.c	Sample compression/decompression application.
	ct.img	Example CT image file. (This file is generated by mc3file and can be regenerated by the user, if need be.)
	duplicate.c	Sample for using MC_Standard_Compressor/MC_Standard_De compressor via MC_Duplicate_Message.
	general_util.c	General utilities for all sample programs.
	general_util.h	General utilities for all sample programs.
	inetd_echo_scp.c	Sample DICOM Echo SCP, using the inetd functionality.
	makefile	Makefile for example programs.

med_fsu.c	Media File Set Updater Application.
merge.ini	Merge DICOM Toolkit Initialization Configuration File. (Used by all sample applications.)
mergecom.app	Merge DICOM Toolkit Application Profile Configuration File.
mergecom.pro	Merge DICOM Toolkit System Profile Configuration File.
mergecom.srv	Merge DICOM Toolkit Service Profile Configuration File.
mpeg2dicom.c	Sample for packing/unpacking MPEG2 streams into/from DICOM files.
prnt_scp.c	Sample Print SCP Application.
prnt_scu.c	Sample Print SCU Application.
prnt_svc.h	Sample Print Application header file.
qr.h	Sample Query/Retrieve Application Include file.
qr_get_scp.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
qr_get_scu.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
qr_scp.c	Sample Query/Retrieve SCP Application.
qr_scu.c	Sample Query/Retrieve SCU Application.
qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
sreport.c	Sample Structured Report Application
ssl_samp.c	SSL callbacks for SSL SCU and SSL SCP.
ssl_samp.h	Header file for SSL callbacks for SSL SCU and SSL SCP.
ssl_scp.c	Sample SCP application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.

	ssl_scp.h	Header file for sample SSL SCP application.
	ssl_scu.c	Sample SCU application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	stor_scp.c	Sample Storage SCP Application.
	stor_scu.c	Sample Storage SCU Application.
	work.dat	Database flat file, used by the Modality Worklist SCP application.
	workdata.c	Sample database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	workdata.h	Sample header file for database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	work_scp.c	Sample Modality Worklist and Modality Performed Procedure Step SCP Application.
	work_scu.c	Sample Modality Worklist and Modality Performed Procedure Step SCU Application.
mc3bin	genconf	Configuration source file generation utility.
	gendict	Dictionary source file generation utility.
	mc3comp	Compare the values within two DICOM message or file objects.
	mc3conv	Convert a DICOM message or file object into a new transfer syntax.
	mc3dcomb	Runtime Dictionary Combine utility.
	mc3dict	Runtime DICOM Data Dictionary utility.
	mc3echo	DICOM Echo Test Utility.
	mc3file	Generate a DICOM message object.
	mc3icomb	Runtime Info Combine utility.
	mc3info	Runtime Message Database generation utility.
	mc3list	List a DICOM message object.

	mc3valid	Validate a DICOM message object.
mc3doc	Database.pdf	Merge DICOM Toolkit DICOM <i>Database Manual</i> .
	Platform.pdf	This document.
	Refer.pdf	Merge DICOM Toolkit <i>Reference Manual</i> .
	Sample.pdf	Merge DICOM Toolkit <i>Sample Applications Guide</i> .
	User.pdf	Merge DICOM Toolkit <i>User's Manual</i> .
mc3inc	bmp.h	Pegasus include file.
	diction.h	DICOM Data Dictionary macros
	mc3items.h	Library include file for use with mc3adv.a.
	mc3media.h	Library include file for use with mc3adv.a.
	mc3msg.h	Library include file for use with mc3adv.a.
	mc3services.h	Library include file for use with mc3adv.a.
	mcstatus.h	Library include file for use with mc3adv.a.
	mergecom.h	Library include file for use with mc3adv.a.
	pcd.h	Pegasus include file.
	pic.h	Pegasus include file.
	stdtypes.h	Pegasus include file.
mc3lib	libcudata.so.49	Unicode conversion data.
	libicuuc.so.49	Unicode conversion library.
	libpicu20.so	Pegasus shared object which needs to be linked in when you compile your source.
	libxml2.so.2.9.3	XML parsing library.
	mc3adv.a	Merge DICOM Toolkit software static library.
	mc3adv.so	Merge DICOM Toolkit software shared object.

	picu6120.so	Pegasus shared library for JPEG-LS decompression.
	picu6220.so	Pegasus shared library for JPEG Lossless compression.
	picu6320.so	Pegasus shared library for JPEG Lossless decompression.
	picu6420.so	Pegasus shared library for JPEG Lossy compression.
	picu6520.so	Pegasus shared library for JPEG Lossy decompression.
	picu6820.so	Pegasus shared library for JPEG 2000 compression.
	picu6920.so	Pegasus shared library for JPEG 2000 decompression.
mc3msg	default.pfl	Default configuration file used by mc3file.
	diction.pfl	Runtime DICOM Data Dictionary profile.
	message.txt	DICOM message formats for INFO purposes.
	info.pfl	Runtime Message Database profile.
	mc3dcomb.pfl	Sample configuration file for use with mc3dcomb utility.
	mc3icomb.cfg	Sample configuration file for use with mc3icomb utility.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

32-Bit Mac OS® X Using Universal Binaries (008-91301)

Supported Configurations

The following table describes the system requirements for the Merge DICOM Toolkit.

Category	Requirement
Hardware	Mac OS X supported hardware.
Software (base)	Mac OS X v10.4 or later.
Software (development)	i686-apple-darwin8-gcc-4.0.1 (GCC) 4.0.1 (Apple Computer, Inc. build 5250).

The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for 32-Bit Mac OS® X Using Universal Binaries is provided as a **static library** and a **dynamic library**.

Static library

A static library is a collection of subroutines that are callable by your programs. To use them, simply link it with your program. The following compilation options were utilized to create the static library:

Compile Options	Description
-arch i386	Support Intel based MacOS X computers
-arch ppc	Support PowerPC based MacOS X computers
-isysroot /Developer/SDKs/MacOSX10.4u.sdk	Use the MacOS X 10.4 SDK.
-O2	Enable optimization.
-D_REENTRANT	Produce reentrant code that is thread safe.

Dynamic library

A dynamic library is similar to a static library. It contains entry points for your application to use and call and contains code that will be executed by many different modules. The difference, however, is that the code is not included in the executable file built by the linker or loader. Instead, the code is loaded at runtime when the resources are requested. The code is then mapped into the processes address space.

The following compile options were used to generate the dynamic library:

Compile Options	Description
-arch i386	Support Intel based MacOS X computers.
-arch ppc	Support PowerPC based MacOS X computers.
-isysroot /Developer/SDKs/MacOSX10.4u .sdk	Use the MacOS X 10.4 SDK.

-O2	Enable optimization.
-D_REENTRANT	Produce reentrant code that is thread safe.

The following link options were used to generate the dynamic library:

Link Options	Description
-dynamiclib	Create a dynamic library.
-install_name mc3adv.dylib	Set the name of the dynamic library.
-arch i386	Support Intel based MacOS X computers.
-arch ppc	Support PowerPC based MacOS X computers.
-single_module	Build the library so it only has one module.
-flat_namespace	Force the output to be built as a flat namespace image.
-Wl, -syslibroot, /Developer/SDKs/MacOSX10.4u.sdk	Use the MacOS X 10.4 SDK.

Miscellaneous Notes

Threading Support

The Merge DICOM Toolkit for MacOS X with Universal Binaries supports multi-thread applications. It utilizes the pthreads library to support multiple threads.

Compression Support

The Merge DICOM Toolkit for MacOS X with Universal Binaries does not support the Pegasus libraries for JPEG compression and decompression.

Unicode Support

The Merge DICOM Toolkit for 32-bit MacOS X with Universal Binaries supports Unicode conversion of DICOM defined character sets with and without code extensions. Two optional shared object libraries, libicuuc.49.dylib and libicudata.49.dylib, are distributed with the toolkit and are used to perform Unicode character set conversion. Users that wish to use Unicode conversion functions must call MC_Enable_Unicode_Conversion to initialize the shared object libraries and ensure the dependency files listed in the table below are available at runtime. Existing users that have no plan to use the Unicode conversion functions don't need to deploy the two shared objects and their dependency files.

Dependency files of libcuc.49.dylib and libcudata.49.dylib for 32-bit MacOS X with Universal Binaries platform:

Unicode conversion library	Dependency File	Description
libcuc.49.dylib libcudata.49.dylib	libstdc++.6.dylib	Standard C++ library
	libgcc_s.1.dylib	GCC library
	libSystem.B.dylib	Mac OS X library

Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_osxi	Information concerning how this distribution was created.
	setup.sh	Environment variables setup executable for sh.
	setup.csh	Environment variables setup executable for csh.
mc3apps	comp.c	Sample compression/decompression application.
	ct.img	Example CT image file. (This file is generated by mc3file and can be regenerated by the user, if need be.)
	duplicate.c	Sample for using MC_Standard_Compressor/MC_Standard_De compressor via MC_Duplicate_Message.
	general_util.c	General utilities for all sample programs.
	general_util.h	General utilities for all sample programs.
	inetd_echo_scp.c	Sample DICOM Echo SCP, using the inetd functionality.
	makefile	Makefile for example programs.
	med_fsu.c	Media File Set Updater Application.

merge.ini	Merge DICOM Toolkit Initialization Configuration File. (Used by all sample applications.)
mergecom.app	Merge DICOM Toolkit Application Profile Configuration File.
mergecom.pro	Merge DICOM Toolkit System Profile Configuration File.
mergecom.srv	Merge DICOM Toolkit Service Profile Configuration File.
mpeg2dicom.c	Sample for packing/unpacking MPEG2 streams into/from DICOM files.
prnt_scp.c	Sample Print SCP Application.
prnt_scu.c	Sample Print SCU Application.
prnt_svc.h	Sample Print Application header file.
qr.h	Sample Query/Retrieve Application Include file.
qr_get_scp.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
qr_get_scu.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
qr_scp.c	Sample Query/Retrieve SCP Application.
qr_scu.c	Sample Query/Retrieve SCU Application.
qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
sreport.c	Sample Structured Report Application
ssl_samp.c	SSL callbacks for SSL SCU and SSL SCP.
ssl_samp.h	Header file for SSL callbacks for SSL SCU and SSL SCP.
ssl_scp.c	Sample SCP application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
ssl_scp.h	Header file for sample SSL SCP application.

	ssl_scu.c	Sample SCU application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	stor_scp.c	Sample Storage SCP Application.
	stor_scu.c	Sample Storage SCU Application.
	work.dat	Database flat file, used by the Modality Worklist SCP application.
	workdata.c	Sample database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	workdata.h	Sample header file for database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	work_scp.c	Sample Modality Worklist and Modality Performed Procedure Step SCP Application.
	work_scu.c	Sample Modality Worklist and Modality Performed Procedure Step SCU Application.
mc3bin	genconf	Configuration source file generation utility.
	gendict	Dictionary source file generation utility.
	mc3comp	Compare the values within two DICOM message or file objects.
	mc3conv	Convert a DICOM message or file object into a new transfer syntax.
	mc3dcomb	Runtime Dictionary Combine utility.
	mc3dict	Runtime DICOM Data Dictionary utility.
	mc3echo	DICOM Echo Test Utility.
	mc3file	Generate a DICOM message object.
	mc3icomb	Runtime Info Combine utility.
	mc3info	Runtime Message Database generation utility.
	mc3list	List a DICOM message object.
	mc3valid	Validate a DICOM message object.

mc3doc	Database.pdf	Merge DICOM Toolkit DICOM <i>Database Manual</i> .
	Platform.pdf	This document.
	Refer.pdf	Merge DICOM Toolkit <i>Reference Manual</i> .
	Sample.pdf	Merge DICOM Toolkit <i>Sample Applications Guide</i> .
	User.pdf	Merge DICOM Toolkit <i>User's Manual</i> .
mc3inc	diction.h	DICOM Data Dictionary macros.
	mc3items.h	Library include file for use with mc3adv.a.
	mc3media.h	Library include file for use with mc3adv.a.
	mc3msg.h	Library include file for use with mc3adv.a.
	mc3services.h	Library include file for use with mc3adv.a.
	mcstatus.h	Library include file for use with mc3adv.a.
	mergecom.h	Library include file for use with mc3adv.a.
mc3lib	libcudata49.dylib	Unicode conversion data.
	libcudata49.dylib.ppc	Unicode conversion data for Power PC based machine. This file must be renamed to libcudata.49.dylib before running application.
	libcuuc.49.dylib	Unicode conversion library.
	libxml2.2.dylib	XML parsing library.
	mc3adv.a	Merge DICOM Toolkit software static library.
	mc3adv.dylib	Merge DICOM Toolkit software dynamic library.
mc3msg	default.pfl	Default configuration file used by mc3file.
	diction.pfl	Runtime DICOM Data Dictionary profile.
	message.txt	DICOM message formats for INFO purposes.
	info.pfl	Runtime Message Database profile.

	mc3dcomb.pfl	Sample configuration file for use with mc3dcomb utility.
	mc3icomb.cfg	Sample configuration file for use with mc3icomb utility.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

64-Bit Mac OS® X on Intel® x64 (008-91303)

Supported Configurations

The following table describes the system requirements for the Merge DICOM Toolkit.

Category	Requirement
Hardware	Mac OS X supported 64-bit Intel x86 hardware.
Software (base)	Mac OS X v10.5.6 or later.
Software (development)	i686-apple-darwin8-gcc-4.0.1 (GCC) 4.0.1 (Apple Computer, Inc. build 5484).

The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for 64-Bit Mac OS® X on Intel® x64 is provided as a **static library** and a **dynamic library**.

Static library

A static library is a collection of subroutines that are callable by your programs. To use them, simply link it with your program. The following compilation options were utilized to create the static library:

Compile Options	Description
-arch i386	Support Intel based MacOS X computers.
-m64	Generate code for a 64-bit environment.
-O2	Enable optimization.
-D_REENTRANT	Produce reentrant code that is thread safe.

Dynamic library

A dynamic library is similar to a static library. It contains entry points for your application to use and call and contains code that will be executed by many

different modules. The difference, however, is that the code is not included in the executable file built by the linker or loader. Instead, the code is loaded at runtime when the resources are requested. The code is then mapped into the processes address space.

The following compile options were used to generate the dynamic library:

Compile Options	Description
-arch i386	Support Intel based MacOS X computers
-m64	Generate code for a 64-bit environment
-O2	Enable optimization.
-D_REENTRANT	Produce reentrant code that is thread safe.

The following link options were used to generate the dynamic library:

Link Options	Description
-dynamiclib	Create a dynamic library.
-install_name mc3adv.dylib	Set the name of the dynamic library.
-arch i386	Support Intel based MacOS X computers.
-single_module	Build the library so it only has one module.
-flat_namespace	Force the output to be built as a flat namespace image.

Miscellaneous Notes

Threading Support

The Merge DICOM Toolkit for 64-bit MacOS X supports multi-threaded applications. It utilizes the pthreads library to support multiple threads.

Compression Support

The Merge DICOM Toolkit for 64-bit MacOS X does not support the Pegasus libraries for JPEG compression and decompression, only RLE compression is supported.

Unicode Support

The Merge DICOM Toolkit for 64-bit MacOS X on Intel supports Unicode conversion of DICOM defined character sets with and without code extensions. Two optional shared object libraries, libicuuc.49.dylib and libicudata.49.dylib, are distributed with the toolkit and are used to perform Unicode character set

conversion. Users that wish to use Unicode conversion functions must call `MC_Enable_Unicode_Conversion` to initialize the shared object libraries and ensure the dependency files listed in the table below are available at runtime. Existing users that have no plan to use the Unicode conversion functions don't need to deploy the two shared objects and their dependency files.

Dependency files of `libcuc.49.dylib` and `libcudata.49.dylib` for 64-bit MacOS X on Intel platform:

Unicode conversion library	Dependency File	Description
libcuc.49.dylib libcudata.49.dylib	libstdc++.6.dylib	Standard C++ library
	libgcc_s.1.dylib	GCC library
	libSystem.B.dylib	Mac OS X library

Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_osxi64	Information concerning how this distribution was created.
	setup.sh	Environment variables setup executable for sh.
	setup.csh	Environment variables setup executable for csh.
mc3apps	comp.c	Sample compression/decompression application.
	ct.img	Example CT image file. (This file is generated by mc3file and can be regenerated by the user, if need be.)
	duplicate.c	Sample for using <code>MC_Standard_Compressor/MC_Standard_Decompressor</code> via <code>MC_Duplicate_Message</code> .
	general_util.c	General utilities for all sample programs.
	general_util.h	General utilities for all sample programs.

Directory	File	Description
	inetd_echo_scp.c	Sample DICOM Echo SCP, using the inetd functionality.
	makefile	Makefile for example programs.
	med_fsu.c	Media File Set Updater Application.
	merge.ini	Merge DICOM Toolkit Initialization Configuration File. (Used by all sample applications.)
	mergecom.app	Merge DICOM Toolkit Application Profile Configuration File.
	mergecom.pro	Merge DICOM Toolkit System Profile Configuration File.
	mergecom.srv	Merge DICOM Toolkit Service Profile Configuration File.
	mpeg2dicom.c	Sample for packing/unpacking MPEG2 streams into/from DICOM files.
	prnt_scp.c	Sample Print SCP Application.
	prnt_scu.c	Sample Print SCU Application.
	prnt_svc.h	Sample Print Application header file.
	qr.h	Sample Query/Retrieve Application Include file.
	qr_get_scp.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_get_scu.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_scp.c	Sample Query/Retrieve SCP Application.
	qr_scu.c	Sample Query/Retrieve SCU Application.
	qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
	sreport.c	Sample Structured Report Application
	ssl_samp.c	SSL callbacks for SSL SCU and SSL SCP.

Directory	File	Description
	ssl_samp.h	Header file for SSL callbacks for SSL SCU and SSL SCP.
	ssl_scp.c	Sample SCP application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	ssl_scp.h	Header file for sample SSL SCP application.
	ssl_scu.c	Sample SCU application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	stor_scp.c	Sample Storage SCP Application.
	stor_scu.c	Sample Storage SCU Application.
	work.dat	Database flat file, used by the Modality Worklist SCP application.
	workdata.c	Sample database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	workdata.h	Sample header file for database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	work_scp.c	Sample Modality Worklist and Modality Performed Procedure Step SCP Application.
	work_scu.c	Sample Modality Worklist and Modality Performed Procedure Step SCU Application.

Directory	File	Description
mc3bin	genconf	Configuration source file generation utility.
	gendict	Dictionary source file generation utility.
	mc3comp	Compare the values within two DICOM message or file objects.
	mc3conv	Convert a DICOM message or file object into a new transfer syntax.
	mc3dcomb	Runtime Dictionary Combine utility.
	mc3dict	Runtime DICOM Data Dictionary utility.
	mc3echo	DICOM Echo Test Utility.
	mc3file	Generate a DICOM message object.
	mc3icomb	Runtime Info Combine utility.
	mc3info	Runtime Message Database generation utility.
	mc3list	List a DICOM message object.
	mc3valid	Validate a DICOM message object.
mc3doc	Database.pdf	Merge DICOM Toolkit DICOM <i>Database Manual</i> .
	Platform.pdf	This document.
	Refer.pdf	Merge DICOM Toolkit <i>Reference Manual</i> .
	Sample.pdf	Merge DICOM Toolkit <i>Sample Applications Guide</i> .
	User.pdf	Merge DICOM Toolkit <i>User's Manual</i> .
mc3inc	diction.h	DICOM Data Dictionary macros.
	mc3items.h	Library include file for use with mc3adv.a.
	mc3media.h	Library include file for use with mc3adv.a.
	mc3msg.h	Library include file for use with mc3adv.a.
	mc3services.h	Library include file for use with mc3adv.a.

Directory	File	Description
	mcstatus.h	Library include file for use with mc3adv.a.
	mergecom.h	Library include file for use with mc3adv.a.
mc3lib	libcudata.49.dylib	Unicode conversion data.
	libcuc.49.dylib	Unicode conversion library.
	libxml2.2.dylib	XML parsing library.
	mc3adv.a	Merge DICOM Toolkit software static library.
	mc3adv.dylib	Merge DICOM Toolkit software object code dynamic library.
mc3msg	default.pfl	Default configuration file used by mc3file.
	diction.pfl	Runtime DICOM Data Dictionary profile.
	message.txt	DICOM message formats for INFO purposes.
	info.pfl	Runtime Message Database profile.
	mc3dcomb.pfl	Sample configuration file for use with mc3dcomb utility.
	mc3icomb.cfg	Sample configuration file for use with mc3icomb utility.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

32-Bit Android on ARMv7-A (008-91998)

Supported Configurations

The following table describes the system requirements for the Merge DICOM Toolkit.

Category	Requirement
Hardware	Android supported ARMv7-A hardware.
Software (base)	Android 5.1 (Lollipop) or later.

Software (development)	Android NDK Revision 10e (arm gcc-4.8 toolchain)
------------------------	--

The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for Android on ARMv7-A is provided in two forms: a **shared object** and a **static library**.

Static library

A static library is a collection of subroutines that are callable by your programs. To use them, simply link them with your program. There are no specific compiler options needed to link with the static library.

Shared object

A shared object is similar to a static library. It contains entry points for your application to use and call and contains code that will be executed by many different modules. The difference, however, is that the code is not included in the executable file built by the linker or loader. Instead, the code is loaded at runtime when the resources are requested. The code is then mapped into the processes address space.

The use of the two types of libraries is exactly the same: they are “linked” into an application program by the system loader after the application has been compiled. The way the system loader constructs the executable is different, however.

When a static library is linked with an application, an executable is produced that contains the code of the application and the code of the library. This is not true with code produced and linked with shared object methods. The application and the shared “library” must be compiled and linked with special compiler flags.

The compiler flags used to generate the Merge DICOM Toolkit static library and shared object are:

Flag	Type	Description
-O3	Compile Time	Optimization level.
-fsigned-char	Compile Time	char data type is treated as unsigned on ARM architecture.
-fpic	Compile Time	Produce position independent code.
-D_REENTRANT	Compile Time	Produce reentrant code that is thread safe.

Also, the `LD_LIBRARY_PATH` environment variable will need to be modified to contain the path of the `libmc3adv.so` file. The following is an excerpt from the man page of `ld` and describes the function of the `LD_LIBRARY_PATH` environment variable:

“`LD_LIBRARY_PATH` is a list of directories in which to search for libraries specified with the `-l` option. Multiple directories are separated by a colon.”
 “It is also used to specify libraries search path to the run-time linker, that is, if

`LD_LIBRARY_PATH` exists in the environment, the run-time linker searches the directories named in it, before the default directories for the shared objects to be linked with the program at execution.”

It should be noted that the Merge DICOM Toolkit shared object (`libmc3adv.so`) is designed to be “dynamic-safe” and optimized. This means that the shared object is safe when more than one application executed the same code at the same time.

Miscellaneous Notes

Threading Support

The Merge DICOM Toolkit for Android supports multi-threaded applications. Please see the *User's Manual* for details on the limitations for using Merge DICOM Toolkit with multiple threads.

Compression Support

The Merge DICOM Toolkit for Android on ARMv7-A supports the AIMTools (formerly known as Pegasus) Imaging libraries for decompression only from Accusoft.

The JPEG2000, Lossy JPEG and the Lossless JPEG decompressors can be utilized within your application without purchasing an additional license from Accusoft. However, the JPEG2000, Lossy and Lossless JPEG Pegasus libraries are limited to decompressing at a maximum rate of 3 frames per second. This limit can be removed by purchasing a license from Accusoft Corporation (www.accusoft.com) and configuring that license in our `mergecom.pro` configuration file.

Built in RLE compression/decompression is also available.

Unicode Support

The Merge DICOM Toolkit for 32-bit Android supports Unicode conversion of DICOM defined character sets with and without code extensions. Two optional shared object libraries, `libicuuc.so.49` and `libicudata.so.49`, are distributed with the toolkit and are used to perform Unicode character set conversion. Users that wish to use Unicode conversion functions must call `MC_Enable_Unicode_Conversion` to initialize the shared object libraries and ensure the dependency files listed in the table below are available at runtime. Existing users that have no plan to use the Unicode conversion functions don't need to deploy the two shared objects and their dependency files.

Dependency files of `libicuuc.so.49` and `libicudata.so.49` on 32-bit Android platform:

Unicode conversion library	Dependency File	Description
libicuuc.so.49 libicudata.so.49	libdl.so	Dynamic loading library
	libc.so	C library

Unicode conversion library	Dependency File	Description
	libm.so	Math library

Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_andrARMv7A	Information concerning how this distribution was created.
	setup.sh	Environment variables setup executable for sh.
mc3apps	Application.apps.mk	Application.mk file for samples.
	comp.c	Sample compression/decompression application.
	ct.img	Example CT image file. (This file is generated by mc3file and can be regenerated by the user, if need be.)
	duplicate.c	Sample for using MC_Standard_Compressor/MC_Standard_De compressor via MC_Duplicate_Message.
	general_util.c	General utilities for all sample programs.
	general_util.h	General utilities for all sample programs.
	inetd_echo_scp.c	Sample DICOM Echo SCP, using the inetd functionality.
	makefile	Makefile for example programs.
	med_fsu.c	Media File Set Updater Application.
	merge.ini	Merge DICOM Toolkit Initialization Configuration File. (Used by all sample applications.)
	mergecom.app	Merge DICOM Toolkit Application Profile Configuration File.

Directory	File	Description
	mergecom.pro	Merge DICOM Toolkit System Profile Configuration File.
	mergecom.srv	Merge DICOM Toolkit Service Profile Configuration File.
	mpeg2dicom.c	Sample for packing/unpacking MPEG2 streams into/from DICOM files.
	prnt_scp.c	Sample Print SCP Application.
	prnt_scu.c	Sample Print SCU Application.
	prnt_svc.h	Sample Print Application header file.
	qr.h	Sample Query/Retrieve Application Include file.
	qr_get_scp.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_get_scu.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_scp.c	Sample Query/Retrieve SCP Application.
	qr_scu.c	Sample Query/Retrieve SCU Application.
	qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
	sreport.c	Sample Structured Report Application
	ssl_samp.c	SSL callbacks for SSL SCU and SSL SCP.
	ssl_samp.h	Header file for SSL callbacks for SSL SCU and SSL SCP.
	ssl_scp.c	Sample SCP application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	ssl_scp.h	Header file for sample SSL SCP application.
	ssl_scu.c	Sample SCU application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.

Directory	File	Description
	stor_scp.c	Sample Storage SCP Application.
	stor_scu.c	Sample Storage SCU Application.
	work.dat	Database flat file, used by the Modality Worklist SCP application.
	workdata.c	Sample database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	workdata.h	Sample header file for database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	work_scp.c	Sample Modality Worklist and Modality Performed Procedure Step SCP Application.
	work_scu.c	Sample Modality Worklist and Modality Performed Procedure Step SCU Application.
mc3bin	genconf	Configuration source file generation utility.
	gendict	Dictionary source file generation utility.
	mc3comp	Compare the values within two DICOM message or file objects.
	mc3conv	Convert a DICOM message or file object into a new transfer syntax.
	mc3dcomb	Runtime Dictionary Combine utility.
	mc3dict	Runtime DICOM Data Dictionary utility.
	mc3echo	DICOM Echo Test Utility.
	mc3file	Generate a DICOM message object.
	mc3icomb	Runtime Info Combine utility.
	mc3info	Runtime Message Database generation utility.
	mc3list	List a DICOM message object.
mc3valid	Validate a DICOM message object.	
mc3doc	Database.pdf	Merge DICOM Toolkit DICOM <u>Database Manual</u> .

Directory	File	Description
	Platform.pdf	This document.
	Refer.pdf	Merge DICOM Toolkit Reference Manual.
	Sample.pdf	Merge DICOM Toolkit Sample Applications Guide.
	User.pdf	Merge DICOM Toolkit User's Manual.
mc3inc	bmp.h	Pegasus include file.
	diction.h	DICOM Data Dictionary macros.
	mc3items.h	Library include file for use with libmc3adv.a.
	mc3media.h	Library include file for use with libmc3adv.a.
	mc3msg.h	Library include file for use with libmc3adv.a.
	mc3services.h	Library include file for use with libmc3adv.a.
	mcstatus.h	Library include file for use with libmc3adv.a.
	mergecom.h	Library include file for use with libmc3adv.a.
	pcd.h	Pegasus include file.
	pic.h	Pegasus include file.
	stdtypes.h	Pegasus include file.
mc3lib	libcudata.so.49	Unicode conversion data.
	libicuuc.so.49	Unicode conversion library.
	libmc3adv.a	Merge DICOM Toolkit software static library.
	libmc3adv.so	Merge DICOM Toolkit software shared object.
	libpicd20.so	Pegasus shared object which needs to be linked in when you compile your source.

Directory	File	Description
	libpicd6320.so	Pegasus library for JPEG Lossless decompression. This file must be in a directory pointed to by the LD_LIBRARY_PATH environment variable or the PEGASUS_OPCODE_PATH configuration setting in mergecom.pro.
	libpicd6520.so	Pegasus library for JPEG Lossy decompression. This file must be in a directory pointed to by the LD_LIBRARY_PATH environment variable or the PEGASUS_OPCODE_PATH configuration setting in mergecom.pro.
	libpicd6920.so	Pegasus library for JPEG 2000 decompression. This file must be in a directory pointed to by the LD_LIBRARY_PATH environment variable or the PEGASUS_OPCODE_PATH configuration setting in mergecom.pro.
	libxml2.la	libtool library file for libxml2
	libxml2.so	libxml2 shared object
mc3msg	default.pfl	Default configuration file used by mc3file.
	diction.pfl	Runtime DICOM Data Dictionary profile.
	message.txt	DICOM message formats for INFO purposes.
	info.pfl	Runtime Message Database profile.
	mc3dcomb.pfl	Sample configuration file for use with mc3dcomb utility.
	mc3icomb.cfg	Sample configuration file for use with mc3icomb utility.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

64-Bit Android on ARMv8-A (89-00161-00)

Supported Configurations

The following table describes the system requirements for the Merge DICOM Toolkit.

Category	Requirement
Hardware	Android supported ARMv8-A hardware.
Software (base)	Android 5.1 (Lollipop) or later.
Software (development)	Android NDK Revision 10e (aarch64 gcc-4.9 toolchain)

The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for Android on ARMv8-A is provided in two forms: a **shared object** and a **static library**.

Static library

A static library is a collection of subroutines that are callable by your programs. To use them, simply link them with your program. There are no specific compiler options needed to link with the static library.

Shared object

A shared object is similar to a static library. It contains entry points for your application to use and call and contains code that will be executed by many different modules. The difference, however, is that the code is not included in the executable file built by the linker or loader. Instead, the code is loaded at runtime when the resources are requested. The code is then mapped into the processes address space.

The use of the two types of libraries is exactly the same: they are “linked” into an application program by the system loader after the application has been compiled. The way the system loader constructs the executable is different, however.

When a static library is linked with an application, an executable is produced that contains the code of the application and the code of the library. This is not true with code produced and linked with shared object methods. The application and the shared “library” must be compiled and linked with special compiler flags.

The compiler flags used to generate the Merge DICOM Toolkit static library and shared object are:

Flag	Type	Description
-O3	Compile Time	Optimization level.
-fsigned-char	Compile Time	char data type is treated as unsigned on ARM architecture.

Flag	Type	Description
-fpic	Compile Time	Produce position independent code.
-D_REENTRANT	Compile Time	Produce reentrant code that is thread safe.

Also, the `LD_LIBRARY_PATH` environment variable will need to be modified to contain the path of the `libmc3adv.so` file. The following is an excerpt from the `man` page of `ld` and describes the function of the `LD_LIBRARY_PATH` environment variable:

“`LD_LIBRARY_PATH` is a list of directories in which to search for libraries specified with the `-l` option. Multiple directories are separated by a colon.”
 “It is also used to specify libraries search path to the run-time linker, that is, if `LD_LIBRARY_PATH` exists in the environment, the run-time linker searches the directories named in it, before the default directories for the shared objects to be linked with the program at execution.”

It should be noted that the Merge DICOM Toolkit shared object (`libmc3adv.so`) is designed to be “dynamic-safe” and optimized. This means that the shared object is safe when more than one application executed the same code at the same time.

Miscellaneous Notes

Threading Support

The Merge DICOM Toolkit for Android supports multi-threaded applications. Please see the *User's Manual* for details on the limitations for using Merge DICOM Toolkit with multiple threads.

Compression Support

The AIMTools libraries from Accusoft do not provide support for JPEG/J2K compression/decompression for the Merge DICOM Toolkit for Android on ARMv8-A.

Built in RLE compression/decompression is available.

Unicode Support

The Merge DICOM Toolkit for 64-bit Android supports Unicode conversion of DICOM defined character sets with and without code extensions. Two optional shared object libraries, `libicuuc.so.49` and `libcudata.so.49`, are distributed with the toolkit and are used to perform Unicode character set conversion. Users that wish to use Unicode conversion functions must call `MC_Enable_Unicode_Conversion` to initialize the shared object libraries and ensure the dependency files listed in the table below are available at runtime. Existing users that have no plan to use the Unicode conversion functions do not need to deploy the two shared objects and their dependency files.

Dependency files of libicuuc.so.49 and libcudata.so.49 on 64-bit Android platform:

Unicode conversion library	Dependency File	Description
libicuuc.so.49 libcudata.so.49	libdl.so	Dynamic loading library
	libc.so	C library
	libm.so	Math library

Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_andrARMv8A	Information concerning how this distribution was created.
	setup.sh	Environment variables setup executable for sh.
mc3apps	Application.apps.mk	Application.mk file for samples.
	comp.c	Sample compression/decompression application.
	ct.img	Example CT image file. (This file is generated by mc3file and can be regenerated by the user, if need be.)
	duplicate.c	Sample for using MC_Standard_Compressor/MC_Standard_De compressor via MC_Duplicate_Message.
	general_util.c	General utilities for all sample programs.
	general_util.h	General utilities for all sample programs.
	inetd_echo_scp.c	Sample DICOM Echo SCP, using the inetd functionality.
	makefile	Makefile for example programs.
	med_fsu.c	Media File Set Updater Application.

Directory	File	Description
	merge.ini	Merge DICOM Toolkit Initialization Configuration File. (Used by all sample applications.)
	mergecom.app	Merge DICOM Toolkit Application Profile Configuration File.
	mergecom.pro	Merge DICOM Toolkit System Profile Configuration File.
	mergecom.srv	Merge DICOM Toolkit Service Profile Configuration File.
	mpeg2dicom.c	Sample for packing/unpacking MPEG2 streams into/from DICOM files.
	prnt_scp.c	Sample Print SCP Application.
	prnt_scu.c	Sample Print SCU Application.
	prnt_svc.h	Sample Print Application header file.
	qr.h	Sample Query/Retrieve Application Include file.
	qr_get_scp.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_get_scu.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_scp.c	Sample Query/Retrieve SCP Application.
	qr_scu.c	Sample Query/Retrieve SCU Application.
	qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
	sreport.c	Sample Structured Report Application
	ssl_samp.c	SSL callbacks for SSL SCU and SSL SCP.
	ssl_samp.h	Header file for SSL callbacks for SSL SCU and SSL SCP.
	ssl_scp.c	Sample SCP application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.

Directory	File	Description
	ssl_scp.h	Header file for sample SSL SCP application.
	ssl_scu.c	Sample SCU application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	stor_scp.c	Sample Storage SCP Application.
	stor_scu.c	Sample Storage SCU Application.
	work.dat	Database flat file, used by the Modality Worklist SCP application.
	workdata.c	Sample database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	workdata.h	Sample header file for database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	work_scp.c	Sample Modality Worklist and Modality Performed Procedure Step SCP Application.
	work_scu.c	Sample Modality Worklist and Modality Performed Procedure Step SCU Application.

Directory	File	Description
mc3bin	genconf	Configuration source file generation utility.
	gendict	Dictionary source file generation utility.
	mc3comp	Compare the values within two DICOM message or file objects.
	mc3conv	Convert a DICOM message or file object into a new transfer syntax.
	mc3dcomb	Runtime Dictionary Combine utility.
	mc3dict	Runtime DICOM Data Dictionary utility.
	mc3echo	DICOM Echo Test utility.
	mc3file	Generate a DICOM message object.
	mc3icomb	Runtime Info Combine utility.
	mc3info	Runtime Message Database generation utility.
	mc3list	List a DICOM message object.
	mc3valid	Validate a DICOM message object.
mc3doc	Database.pdf	Merge DICOM Toolkit DICOM Database Manual.
	Platform.pdf	This document.
	Refer.pdf	Merge DICOM Toolkit Reference Manual.
	Sample.pdf	Merge DICOM Toolkit Sample Applications Guide.
	User.pdf	Merge DICOM Toolkit User's Manual.
mc3inc	diction.h	DICOM Data Dictionary macros
	mc3items.h	Library include file for use with libmc3adv.a.
	mc3media.h	Library include file for use with libmc3adv.a.
	mc3msg.h	Library include file for use with libmc3adv.a.
	mc3services.h	Library include file for use with libmc3adv.a.

Directory	File	Description
	mcstatus.h	Library include file for use with libmc3adv.a.
	mergecom.h	Library include file for use with libmc3adv.a.
mc3lib	libcudata.so.49	Unicode conversion data.
	libicuuc.so.49	Unicode conversion library.
	libmc3adv.a	Merge DICOM Toolkit software static library.
	libmc3adv.so	Merge DICOM Toolkit software shared object.
	libxml2.la	libtool library file for libxml2
	libxml2.so	libxml2 shared object
mc3msg	default.pfl	Default configuration file used by mc3file.
	diction.pfl	Runtime DICOM Data Dictionary profile.
	message.txt	DICOM message formats for INFO purposes.
	info.pfl	Runtime Message Database profile.
	mc3dcomb.pfl	Sample configuration file for use with mc3dcomb utility.
	mc3icomb.cfg	Sample configuration file for use with mc3icomb utility.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

64-Bit iOS on ARMv8-A (008-91990)

Supported Configurations

The following table describes the system requirements for the Merge DICOM Toolkit.

Category	Requirement
Hardware	iOS supported ARMv8-A hardware.

Software (base)	iOS 8.1 or later.
Software (development)	XCode 6.4 (Apple LLVM version 6.1.0 gcc)

The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for iOS on ARMv8-A is provided as a **static library**.

Static library

A static library is a collection of subroutines that are callable by your programs. To use them, simply link them with your program. There are no specific compiler options needed to link with the static library.

When a static library is linked with an application, an executable is produced that contains the code of the application and the code of the library.

The compiler flags used to generate the Merge DICOM Toolkit static library are:

Flag	Type	Description
-O2	Compile Time	Optimization level.
-D_REENTRANT	Compile Time	Produce reentrant code that is thread safe.
-isysroot	Compile Time	iPhoneOS6.1.sdk

Miscellaneous Notes

Threading Support

The Merge DICOM Toolkit for iOS supports multi-threaded applications. Please see the *User's Manual* for details on the limitations for using Merge DICOM Toolkit with multiple threads.

Compression Support

The Pegasus libraries do not provide support for JPEG/J2K compression/decompression for the Merge DICOM Toolkit for iOS on ARMv8-A.

Built in RLE compression/decompression is available.

Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_ios	Information concerning how this distribution was created.

Directory	File	Description
	setup.sh	Environment variables setup executable for sh.
	setup.csh	Environment variables setup executable for csh.
mc3apps	comp.c	Sample compression/decompression application.
	ct.img	Example CT image file. (This file is generated by mc3file and can be regenerated by the user, if need be.)
	duplicate.c	Sample for using MC_Standard_Compressor/MC_Standard_De compressor via MC_Duplicate_Message.
	general_util.c	General utilities for all sample programs.
	general_util.h	General utilities for all sample programs.
	inetd_echo_scp.c	Sample DICOM Echo SCP, using the inetd functionality.
	iOSSampleApp.zip	Xcode project exemplifying how to integrate the toolkit library into Swift application.
	makefile	Makefile for example programs.
	med_fsu.c	Media File Set Updater Application.
	merge.ini	Merge DICOM Toolkit Initialization Configuration File. (Used by all sample applications.)
	mergecom.app	Merge DICOM Toolkit Application Profile Configuration File.
	mergecom.pro	Merge DICOM Toolkit System Profile Configuration File.
	mergecom.srv	Merge DICOM Toolkit Service Profile Configuration File.
	mpeg2dicom.c	Sample for packing/unpacking MPEG2 streams into/from DICOM files.
	prnt_scp.c	Sample Print SCP Application.
	prnt_scu.c	Sample Print SCU Application.

Directory	File	Description
	prnt_svc.h	Sample Print Application header file.
	qr.h	Sample Query/Retrieve Application Include file.
	qr_get_scp.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_get_scu.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_scp.c	Sample Query/Retrieve SCP Application.
	qr_scu.c	Sample Query/Retrieve SCU Application.
	qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
	sreport.c	Sample Structured Report Application
	ssl_samp.c	SSL callbacks for SSL SCU and SSL SCP.
	ssl_samp.h	Header file for SSL callbacks for SSL SCU and SSL SCP.
	ssl_scp.c	Sample SCP application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	ssl_scp.h	Header file for sample SSL SCP application.
	ssl_scu.c	Sample SCU application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	stor_scp.c	Sample Storage SCP Application.
	stor_scu.c	Sample Storage SCU Application.
	work.dat	Database flat file, used by the Modality Worklist SCP application.
	workdata.c	Sample database functions used by Modality Worklist and Performed Procedure Step SCP Application.

Directory	File	Description
	workdata.h	Sample header file for database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	work_scp.c	Sample Modality Worklist and Modality Performed Procedure Step SCP Application.
	work_scu.c	Sample Modality Worklist and Modality Performed Procedure Step SCU Application.
mc3bin	genconf	Configuration source file generation utility.
	gendict	Dictionary source file generation utility.
	mc3comp	Compare the values within two DICOM message or file objects.
	mc3conv	Convert a DICOM message or file object into a new transfer syntax.
	mc3dcomb	Runtime Dictionary Combine utility.
	mc3dict	Runtime DICOM Data Dictionary utility.
	mc3echo	DICOM Echo Test Utility.
	mc3file	Generate a DICOM message object.
	mc3icomb	Runtime Info Combine utility.
	mc3info	Runtime Message Database generation utility.
	mc3list	List a DICOM message object.
	mc3valid	Validate a DICOM message object.
mc3doc	Database.pdf	Merge DICOM Toolkit DICOM <i>Database Manual</i> .
	Platform.pdf	This document.
	Refer.pdf	Merge DICOM Toolkit <i>Reference Manual</i> .
	Sample.pdf	Merge DICOM Toolkit <i>Sample Applications Guide</i> .
	User.pdf	Merge DICOM Toolkit <i>User's Manual</i> .
mc3inc	diction.h	DICOM Data Dictionary macros.

Directory	File	Description
	mc3items.h	Library include file for use with mc3adv.a.
	mc3media.h	Library include file for use with mc3adv.a.
	mc3msg.h	Library include file for use with mc3adv.a.
	mc3services.h	Library include file for use with mc3adv.a.
	mcstatus.h	Library include file for use with mc3adv.a.
	mergecom.h	Library include file for use with mc3adv.a.
mc3lib	libcudata.49.dylib	Unicode conversion data.
	libicuuc.49.dylib	Unicode conversion library.
	libxml2.2.dylib	XML parsing library.
	libxml2.a	XML parsing library.
	mc3adv.a	Merge DICOM Toolkit software static library.
mc3msg	default.pfl	Default configuration file used by mc3file.
	diction.pfl	Runtime DICOM Data Dictionary profile.
	message.txt	DICOM message formats for INFO purposes.
	info.pfl	Runtime Message Database profile.
	mc3dcomb.pfl	Sample configuration file for use with mc3dcomb utility.
	mc3icomb.cfg	Sample configuration file for use with mc3icomb utility.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

32-Bit Linux on ARMv7-A (008-92001) – RETIRED

The Merge DICOM Toolkit for Linux on ARMv7-A has been retired starting with release 5.5.0.

64-Bit Linux on ARMv8A (89-00163-00)

Supported Configurations

The following table describes the system requirements for the Merge DICOM Toolkit.

Category	Requirement
Hardware	Linux supported ARM hardware.
Software (base)	Linaro Linux 3.19 or later.
Software (development)	GCC compiler version 4.9.3

The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for Linux on ARMv8 is provided in two forms: a static library and a shared object.

Static library

A static library is a collection of subroutines that are callable by your programs. To use them, simply link them with your program. There are no specific compiler options needed to link with the static library.

Shared object

A shared object is similar to a static library. It contains entry points for your application to use and call and contains code that will be executed by many different modules. The difference, however, is that the code is not included in the executable file built by the linker or loader. Instead, the code is loaded at runtime when the resources are requested. The code is then mapped into the processes address space.

The use of the two types of libraries is exactly the same: they are “linked” into an application program by the system loader after the application has been compiled. The way the system loader constructs the executable is different, however.

When a static library is linked with an application, an executable is produced that contains the code of the application and the code of the library. This is not true with code produced and linked with shared object methods. The application and the shared “library” must be compiled and linked with special compiler flags.

The compiler flags used to generate the Merge DICOM Toolkit static library and shared object are:

Flag	Type	Description
-O3	Compile Time	Optimization level.
-D_REENTRANT	Compile Time	Produce reentrant code that is thread safe.

Miscellaneous Notes

Threading Support

The Merge DICOM Toolkit for Linux on ARMv8 supports multi-threaded applications. Please see the *User's Manual* for details on the limitations for using Merge DICOM Toolkit with multiple threads.

Compression Support

The Pegasus libraries do not provide support for JPEG/J2K compression/decompression for the Merge DICOM Toolkit for Linux on ARMv8-A.

Built in RLE compression/decompression is available.

Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_InxARMv8	Information concerning how this distribution was created.
	setup.sh	Environment variables setup executable for sh.
mc3apps	comp.c	Sample compression/decompression application.
	ct.img	Example CT image file. (This file is generated by mc3file and can be regenerated by the user, if need be.)
	duplicate.c	Sample for using MC_Standard_Compressor/MC_Standard_De compressor via MC_Duplicate_Message.
	general_util.c	General utilities for all sample programs.
	general_util.h	General utilities for all sample programs.
	inetd_echo_scp.c	Sample DICOM Echo SCP, using the inetd functionality.
	makefile	Makefile for example programs.
	med_fsu.c	Media File Set Updater Application.
	merge.ini	Merge DICOM Toolkit Initialization Configuration File. (Used by all sample applications.)

Directory	File	Description
	mergecom.app	Merge DICOM Toolkit Application Profile Configuration File.
	mergecom.pro	Merge DICOM Toolkit System Profile Configuration File.
	mergecom.srv	Merge DICOM Toolkit Service Profile Configuration File.
	mpeg2dicom.c	Sample for packing/unpacking MPEG2 streams into/from DICOM files.
	prnt_scp.c	Sample Print SCP Application.
	prnt_scu.c	Sample Print SCU Application.
	prnt_svc.h	Sample Print Application header file.
	qr.h	Sample Query/Retrieve Application Include file.
	qr_get_scp.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_get_scu.c	Sample Query/Retrieve SCP Application using C-GET for retrieval in source form.
	qr_scp.c	Sample Query/Retrieve SCP Application.
	qr_scu.c	Sample Query/Retrieve SCU Application.
	qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
	sreport.c	Sample Structured Report Application
	ssl_samp.c	SSL callbacks for SSL SCU and SSL SCP.
	ssl_samp.h	Header file for SSL callbacks for SSL SCU and SSL SCP.
	ssl_scp.c	Sample SCP application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	ssl_scp.h	Header file for sample SSL SCP application.

Directory	File	Description
	ssl_scu.c	Sample SCU application for the Storage Service Class using secure socket connections. Modification required to run on platforms other than Windows.
	stor_scp.c	Sample Storage SCP Application.
	stor_scu.c	Sample Storage SCU Application.
	work.dat	Database flat file, used by the Modality Worklist SCP application.
	workdata.c	Sample database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	workdata.h	Sample header file for database functions used by Modality Worklist and Performed Procedure Step SCP Application.
	work_scp.c	Sample Modality Worklist and Modality Performed Procedure Step SCP Application.
	work_scu.c	Sample Modality Worklist and Modality Performed Procedure Step SCU Application.

Directory	File	Description
mc3bin	genconf	Configuration source file generation utility.
	gendict	Dictionary source file generation utility.
	mc3comp	Compare the values within two DICOM message or file objects.
	mc3conv	Convert a DICOM message or file object into a new transfer syntax.
	mc3dcomb	Runtime Dictionary Combine utility.
	mc3dict	Runtime DICOM Data Dictionary utility.
	mc3echo	DICOM Echo Test utility.
	mc3file	Generate a DICOM message object.
	mc3icomb	Runtime Info Combine utility.
	mc3info	Runtime Message Database generation utility.
	mc3list	List a DICOM message object.
	mc3valid	Validate a DICOM message object.
mc3doc	Database.pdf	Merge DICOM Toolkit DICOM <i>Database Manual</i> .
	Platform.pdf	This document.
	Refer.pdf	Merge DICOM Toolkit <i>Reference Manual</i> .
	Sample.pdf	Merge DICOM Toolkit <i>Sample Applications Guide</i> .
	User.pdf	Merge DICOM Toolkit <i>User's Manual</i> .
mc3inc	diction.h	DICOM Data Dictionary macros
	mc3items.h	Library include file for use with mc3adv.a.
	mc3media.h	Library include file for use with mc3adv.a.
	mc3msg.h	Library include file for use with mc3adv.a.
	mc3services.h	Library include file for use with mc3adv.a.

Directory	File	Description
	mcstatus.h	Library include file for use with mc3adv.a.
	mergecom.h	Library include file for use with mc3adv.a.
mc3lib	mc3adv.a	Merge DICOM Toolkit software static library.
	mc3adv.so	Merge DICOM Toolkit software shared object.
mc3msg	default.pfl	Default configuration file used by mc3file.
	diction.pfl	Runtime DICOM Data Dictionary profile.
	message.txt	DICOM message formats for INFO purposes.
	info.pfl	Runtime Message Database profile.
	mc3dcomb.pfl	Sample configuration file for use with mc3dcomb utility.
	mc3icomb.cfg	Sample configuration file for use with mc3icomb utility.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.