Merge DICOM Toolkit™

5.16.0

Platform Notes

Merge Healthcare Incorporated
900 Walnut Ridge Drive
Hartland, WI 53029
USA

Symbols Glossary:

Symbol	Title
	Manufacturer
	Consult Instructions for Use

The full symbols glossary can be viewed at https://www.merative.com/content/dam/merative/documents/brief/Merge_Healthcare_Symbols_Glossary.pdf.

For application support or to report issues with user documentation, contact Customer Support:

1-877-741-5369 (North America)
+31.20.514.5073 (Europe, the Middle East and Africa)
1800 316 746 (Australia)

MC3Support@merative.com

Part	Date	Revision	Description
COM-5281	January 2023	1.0	Updated bi-annually

The latest version of this document can be found at https://mergecustomer.force.com/mergeusercommunity/login.

Contents

Chapter 1. Overview

The Merge DICOM Toolkit™ for provides DICOM functionality through Merge's Applications Programming Interface to various operating system. 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.

1.1. DICOM Toolkit Documentation Roadmap

The Merge DICOM Toolkit documentation is structured as shown in the figure below.

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.

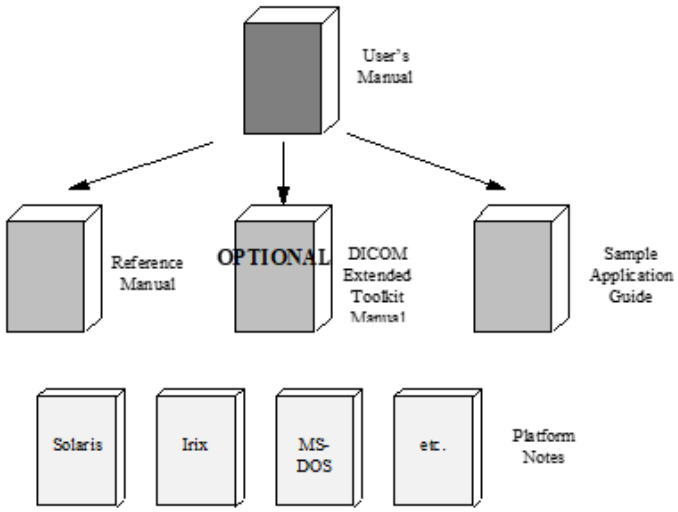
The User's Manual is the foundation for all other documentation because it explains the concepts of DICOM and the DICOM toolkit. Before plunging into the Reference Manual or Sample Application Guide you should be comfortable with the material in the User Manual.

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.

1.1.1. 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 in source form are described in the Sample Application Guide.



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

1.2. Conventions

This manual follows a few formatting conventions.

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

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

1.3. Summary of Supported Platforms

- 32-Bit Microsoft Windows using Microsoft Visual C++2005 (008-91204)
- 64-Bit Microsoft Windows using Microsoft Visual C++2005 (008-91208)
- 64-Bit Microsoft Windows using Microsoft Visual C++ 2019 (New Edition) (89-00349-00)
- 32-Bit Microsoft Windows using Borland® C++ (008-91205)
- 32-Bit Linux® on Intel® x86 (008-91126)
- 64-Bit Linux® on Intel® x86-64 (008-91132)
- 64-Bit Linux® on Intel® x86-64 (New Edition) (89-00350-00)
- 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 with Universal Binaries (008-91301)
- 64-Bit Mac OS® X on Intel® x64 (008-91303)
- 64-Bit macOS® with Universal Binaries on Intel® x64 and ARM64 M1 (New Edition) (89-00359-00)
- 32-Bit Android on ARMv7-A (008-91998)
- 64-Bit Android on ARMv8-A (89-00161-00)
- 64-Bit iOS on ARMv8-A (008-91990)
- ~~32-Bit Linux on ARMv7-A (008-92001)~~ - retired
- 64-Bit Linux on ARMv8-A (89-00163-00)

Chapter 2. 32-Bit Microsoft Windows using Microsoft Visual C++ 2005 (008-91204)

2.1. 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).

2.2. 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 the static library with your program.

Dynamic link libraries, or DLL for short, are the cornerstone of 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.

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 (mc3adv.lib).

Compile Options	Description
/MD	Application is multithreaded and uses C libraries in the form of DLLs. This option should be used when linking with the DLL (mc3adll.lib).
/MDd	Application is multithreaded and uses debug C libraries in the form of DLLs. This option should be used when linking with the DLL (mc3adlld.lib).

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 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.

2.2.1. Third-Party Components Used

The third-party components used by the Merge DICOM Toolkit for 32-bit Windows are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	49.1.2 - read the Important Information below this table
libxml2	Conversion DICOM to/from XML	2.9.10
jansson	Conversion DICOM to/from JSON	2.7
PICTools (aka Pegasus)	Image compression/decompression libraries from Accusoft	2.00.676

Important Information

Unicode encoding/decoding library ICU4C v49.1.2 was introduced in version 4.7.0 of the Merge DICOM C/C++ Toolkit. Since then, a number of vulnerabilities have been reported against v49.1.2 of the ICU4C libraries (see NIST NVD - ICU4C Vulnerabilities).

Although all the high severity vulnerabilities have been addressed and fixed in newer versions of the ICU4C libraries, an upgrade is not feasible on this platform due to compiler requirements that would break backward compatibility.

Toolkit versions 4.7.0 and later are impacted.

Due to these vulnerabilities, starting from release 5.15.0 of the toolkit, the Unicode conversion is turned off by default in the toolkit. OEM customers may choose to manually turn it back on, if they, after their own assessment for their specific application scenario, feel that it is safe to use.

To enable the Unicode conversion, set ENABLE_ICU4C_LIBRARY configuration setting to 'Yes' in the [MEDIA_PARMS] section in mergecom.pro. Alternatively, the `MC_Set_Bool_Config_Value()` API can be used for the same purpose.

If ICU4C is no longer used/required, OEM customers may choose to remove the library files completely from their application product distribution.

2.3. Miscellaneous Notes

2.3.1. Threading Support

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

2.3.2. 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.

2.3.3. 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.

2.3.4. 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 do not 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

2.3.5. 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.

2.3.6. 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`, `sreport` 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.

2.4. Installation

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

2.4.1. 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. 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.

a. 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.

b. Software

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

2.4.2. 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.

a. 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. Alternatively, the pathname of the initialization file can be set at run time using one of the `MC_Set_MergeINI()` or `MC_Set_MergeINI_Unicode()` APIs.

b. 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    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
```

2.5. 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 needed.)
	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.

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.
	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.
	qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
	SampleApps.vsprops	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.

Directory	File	Description
	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.
	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.
	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.

Directory	File	Description
	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.
	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.
	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.
	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 (library version 49.1.2).
	icudt49.dll	Unicode conversion data (library version 49.1.2).
	jansson.dll	Library for encoding, decoding and manipulating JSON data (library version: 2.7).
	jansson.lib	Library for encoding, decoding and manipulating JSON data (library version: 2.7).
	libxml2.dll	XML parsing library (library version: 2.9.10).
	libxml2.lib	XML parsing library (library version: 2.9.10).
	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.

Directory	File	Description
	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.
	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.

Chapter 3. 64-Bit Microsoft Windows using Microsoft Visual C++ 2005 (008-91208)

3.1. 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

3.2. 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 the static library with your program.

Dynamic link libraries, or DLL for short, are the cornerstone of 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 process 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 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 (mc3adv64.lib).

Compile Options	Description
/MD	Application is multithreaded and uses C libraries in the form of DLLs. This option should be used when linking with the DLL (mc3adll64.lib).
/MDd	Application is multithreaded and uses debug C libraries in the form of DLLs. This option should be used when linking with the DLL (mc3adll64d.lib).

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:AMD64	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.

3.2.1. Third-Party Components Used

The third-party components used by the Merge DICOM Toolkit for 32-bit Windows are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	49.1.2 - read the Important Information below this table
libxml2	Conversion DICOM to/from XML	2.9.10
jansson	Conversion DICOM to/from JSON	2.7
PICTools (aka Pegasus)	Image compression/decompression libraries from Accusoft	2.00.676

Important Information

Unicode encoding/decoding library ICU4C v49.1.2 was introduced in version 4.7.0 of the Merge DICOM C/C++ Toolkit. Since then, a number of vulnerabilities have been reported against v49.1.2 of the ICU4C libraries (see NIST NVD - ICU4C Vulnerabilities).

Although all the high severity vulnerabilities have been addressed and fixed in newer versions of the ICU4C libraries, an upgrade is not feasible on this platform due to compiler requirements that would break backward compatibility.

Toolkit versions 4.7.0 and later are impacted.

Due to these vulnerabilities, starting from release 5.15.0 of the toolkit, the Unicode conversion is turned off by default in the toolkit. OEM customers may choose to manually turn it back on, if they, after their own assessment for their specific application scenario, feel that it is safe to use.

To enable the Unicode conversion, set ENABLE_ICU4C_LIBRARY configuration setting to 'Yes' in the [MEDIA_PARMS] section in mergecom.pro. Alternatively, the `MC_Set_Bool_Config_Value()` API can be used for the same purpose.

If ICU4C is no longer used/required, OEM customers may choose to remove the library files completely from their application product distribution.

For the 64-bit Windows platform using Microsoft Visual C++ 2005 specifically, OEM customers have the alternative to upgrade the build environment for their application to use a newer, more modern compiler, in which case they can upgrade to use the new edition of the toolkit for 64-bit Windows, described in Chapter 4, which is built using Microsoft Visual C++ 2019. This new edition uses ICU4C version 70.1.0, which addresses and resolves all the important vulnerabilities.

3.3. Miscellaneous Notes

3.3.1. Threading Support

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

3.3.2. 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.

3.3.3. 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.

3.3.4. 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 do not 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

3.3.5. 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.

3.3.6. 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`, `sreport` 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.

3.4. Installation

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

3.4.1. 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. 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.

a. 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.

b. Software

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

3.5. 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.

3.5.1. 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. Alternatively, the pathname of the initialization file can be set at run time using one of the `MC_Set_MergeINI()` or `MC_Set_MergeINI_Unicode()` APIs.

3.5.2. 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 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
```

3.6. 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 needed.)
	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.
	qr_util.c	Sample Query/Retrieve and Worklist Management Application utility functions.
	SampleApps.vsprops	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.

Directory	File	Description
	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.
	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.

Directory	File	Description
	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.
	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 (library version 49.1.2).
	icudt49.dll	Unicode conversion data (library version 49.1.2).
	jansson.dll	Library for encoding, decoding and manipulating JSON data (library version: 2.7).
	jansson.lib	Library for encoding, decoding and manipulating JSON data (library version: 2.7).
	libxml2.dll	XML parsing library (library version: 2.9.10).
	libxml2.lib	XML parsing library (library version: 2.9.10).
	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.
	mc3adv64.pdb	Debug database for Merge DICOM Toolkit dynamic link library.
	mc3adv64d.dll	Merge DICOM Toolkit dynamic link library for debug builds.

Directory	File	Description
	mc3adv64d.pdb	Debug database for Merge DICOM Toolkit dynamic link library.
	picx20.dll	Pegasus dispatcher DLL.
	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.

Chapter 4. 64-Bit Microsoft Windows using Microsoft Visual C++ 2019 (New Edition) (89-00349-00)

4.1. 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 10 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 2019

4.2. 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 the static library with your program.

Dynamic link libraries, or DLL for short, are the cornerstone of 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 process 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 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 (mc3adv64.lib).

Compile Options	Description
/MD	Application is multithreaded and uses C libraries in the form of DLLs. This option should be used when linking with the DLL (mc3adll64.lib).
/MDd	Application is multithreaded and uses debug C libraries in the form of DLLs. This option should be used when linking with the DLL (mc3adll64d.lib).

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:AMD64	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.

4.2.1. Third-Party Components Used

The third-party components used by the Merge DICOM Toolkit for 64-bit Windows (New Edition) are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	70.1.0
libxml2	Conversion DICOM to/from XML	2.10.2
jansson	Conversion DICOM to/from JSON	2.14
PICTools (aka Pegasus)	Image compression/decompression libraries from Accusoft	2.00.676

4.3. Miscellaneous Notes

4.3.1. Threading Support

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

4.3.2. 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.

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

4.3.4. 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, icuuc70.dll and icudt70.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 do not need to deploy the two shared objects and their dependency files.

Dependency files of icuuc70.dll and icudt70.dll on 64-bit Windows platform:

Unicode conversion library	Dependency File	Description
icuuc70.dll icudt70.dll	vcruntime140.dll	Microsoft VS Runtime Library
	vcruntime140_1.dll	Microsoft C Runtime Library
	msvcp80.dll	Microsoft C++ Runtime Library
	advapi32.dll	Microsoft runtime
	kernel32.dll	Microsoft runtime

4.3.5. 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.

4.3.6. 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`, `sreport` 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.

4.4. Installation

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

4.4.1. 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. 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.

a. 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.

b. Software

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

4.5. Post-Installation

Perform post-installation configuration

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

4.5.1. 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. Alternatively, the pathname of the initialization file can be set at run time using one of the `MC_Set_MergeINI()` or `MC_Set_MergeINI_Unicode()` APIs.

4.5.2. 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 defi-
    nitions 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 Default = MEM

   LARGE_DATA_SIZE= 200

   OBOW_BUFFER_SIZE= 4096

   ...

   DICTIONARY_ACCESS= MEM # | FILEDefault = MEM
```

```
...
DICTIONARY_FILE= C:\mc3db\mrgcom3.dct
TEMP_FILE_DIRECTORY= C:\temp
MSG_INFO_FILE= C:\mc3db\mrgcom3.msg
```

4.6. 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 needed.)
	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.
	prnt_scp.c	Sample Print SCP Application.
	prnt_scp.vcproj	VS 2005 project file.

Directory	File	Description
	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.vsprops	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.

Directory	File	Description
	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.
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.

Directory	File	Description
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.
	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	icuuc70.dll	Unicode conversion library (library version 70.1.0).
	icudt70.dll	Unicode conversion data (library version 70.1.0)..
	jansson.dll	Library for encoding, decoding and manipulating JSON data (library version: 2.14).
	jansson.lib	Library for encoding, decoding and manipulating JSON data (library version: 2.14).
	libxml2.dll	XML parsing library (library version: 2.10.2).
	libxml2.lib	XML parsing library (library version: 2.10.2).
	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.
	mc3adv64.pdb	Debug database for Merge DICOM Toolkit dynamic link library.
	mc3adv64d.dll	Merge DICOM Toolkit dynamic link library for debug builds.
	mc3adv64d.pdb	Debug database for Merge DICOM Toolkit dynamic link library.
	picx20.dll	Pegasus dispatcher DLL.
	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.

Directory	File	Description
	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.

Chapter 5. 32-Bit Microsoft Windows using Borland® C++ (008-91205)

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

5.2. 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 the static library with your program.

Dynamic link libraries, or DLL for short, are the cornerstone of 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 process 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 compile options in the following table should be used:

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

The link options in the following table 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 link line. Some additional compile options were used with the sample applications. The following describes their usage.

- `-w-a-s` - 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-p-r` - 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.

5.2.1. Third-Party Components Used

The third-party components used by the Merge DICOM Toolkit for 32-Bit Microsoft Windows using Borland® C++ are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	49.1.2 - read the Important Information below this table
libxml2	Conversion DICOM to/from XML	2.9.10
jansson	Conversion DICOM to/from JSON	2.7
PICTools (aka Pegasus)	Image compression/decompression libraries from Accusoft	2.00.676

Important Information

Unicode encoding/decoding library ICU4C v49.1.2 was introduced in version 4.7.0 of the Merge DICOM C/C++ Toolkit. Since then, a number of vulnerabilities have been reported against v49.1.2 of the ICU4C libraries (see NIST NVD - ICU4C Vulnerabilities).

Although all the high severity vulnerabilities have been addressed and fixed in newer versions of the ICU4C libraries, an upgrade is not feasible on this platform due to compiler requirements that would break backward compatibility.

Toolkit versions 4.7.0 and later are impacted.

Due to these vulnerabilities, starting from release 5.15.0 of the toolkit, the Unicode conversion is turned off by default in the toolkit. OEM customers may choose to manually turn it back on, if they, after their own assessment for their specific application scenario, feel that it is safe to use.

To enable the Unicode conversion, set ENABLE_ICU4C_LIBRARY configuration setting to 'Yes' in the [MEDIA_PARMS] section in mergecom.pro. Alternatively, the `MC_Set_Bool_Config_Value()` API can be used for the same purpose.

If ICU4C is no longer used/required, OEM customers may choose to remove the library files completely from their application product distribution.

5.3. Miscellaneous Notes

5.3.1. Threading Support

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

5.3.2. 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.

5.3.3. 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 do not 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

5.3.4. 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.

5.4. Installation

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

5.4.1. The 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. 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.

a. 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.

b. Software

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

5.5. 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.

5.5.1. 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. Alternatively, the pathname of the initialization file can be set at run time using one of the **MC_Set_MergeINI()** or **MC_Set_MergeINI_Unicode() APIs**.

5.5.2. 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 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
```

5.6. 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 needed.)
	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.)
	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.

Directory	File	Description
	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.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.

Directory	File	Description
	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.
	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 (library version 49.1.2).
	icudt49.dll	Unicode conversion data (library version 49.1.2).
	jansson.dll	Library for encoding, decoding and manipulating JSON data (library version: 2.7).
	jansson.lib	Library for encoding, decoding and manipulating JSON data (library version: 2.7).
	libxml2.dll	XML parsing library (library version: 2.9.10).
	libxml2.lib	XML parsing library (library version: 2.9.10).
	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.

Directory	File	Description
	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.
	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.

Chapter 6. 32-Bit Linux® on Intel® x86 (008-91126)

6.1. 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.

[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.

6.2. 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 the static library 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 process 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	Produce 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.

6.2.1. Third-Party Components Used

The third-party components used by the Merge DICOM Toolkit for 32-Bit Linux® on Intel® x86 are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	49.1.2 - read the Important Information below this table
libxml2	Conversion DICOM to/from XML	2.9.10
jansson	Conversion DICOM to/from JSON	2.7
PICTools (aka Pegasus)	Image compression/decompression libraries from Accusoft	2.00.676

Important Information

Unicode encoding/decoding library ICU4C v49.1.2 was introduced in version 4.7.0 of the Merge DICOM C/C++ Toolkit. Since then, a number of vulnerabilities have been reported against v49.1.2 of the ICU4C libraries (see NIST NVD - ICU4C Vulnerabilities).

Although all the high severity vulnerabilities have been addressed and fixed in newer versions of the ICU4C libraries, an upgrade is not feasible on this platform due to compiler requirements that would break backward compatibility.

Toolkit versions 4.7.0 and later are impacted.

Due to these vulnerabilities, starting from release 5.15.0 of the toolkit, the Unicode conversion is turned off by default in the toolkit. OEM customers may choose to manually turn it back on, if they, after their own assessment for their specific application scenario, feel that it is safe to use.

To enable the Unicode conversion, set ENABLE_ICU4C_LIBRARY configuration setting to 'Yes' in the [MEDIA_PARMS] section in mergecom.pro. Alternatively, the `MC_Set_Bool_Config_Value()` API can be used for the same purpose.

If ICU4C is no longer used/required, OEM customers may choose to remove the library files completely from their application product distribution.

6.3. Miscellaneous Notes

6.3.1. Threading Support

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

6.3.2. 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.

6.3.3. 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 do not 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
	libc.so.6	C library
	ld-linux.so.2	Linux library

6.4. Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_lnx	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 needed.)
	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.
	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.

Directory	File	Description
	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.

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	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.

Directory	File	Description
mc3lib	libicudata.so.49	Unicode conversion data (library version 49.1.2).
	libicuuc.so.49	Unicode conversion library (library version 49.1.2).
	libxml2.so.2.9.10	XML parsing library(library version 2.9.10).
	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.
	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.

Chapter 7. 64-Bit Linux® on Intel® x86-64 (008-91132)

7.1. 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)	Ubuntu 5.10 or later .[2]
Software (development)	GCC 4.0.2.

[2]Merge DICOM Toolkit for Linux on Intel x64 has been developed using Ubuntu 5.10. Although it should function under any distribution of Linux, Ubuntu 5.10 Linux is the supported version if distribution specific problems arise.

7.2. The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for 64-Bit Linux on Intel 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 the static library 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 process 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	Produced 64 bit code.
`-O3`	Compile Time	Optimization level.
`-D_REENTRANT`	Compile Time	Produce reentrant code that is thread safe.

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.

7.2.1. Third-Party Components Used

The third-party components used by the Merge DICOM Toolkit for 64-Bit Linux® on Intel® x86-64 are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	49.1.2 - read the Important Information below this table
libxml2	Conversion DICOM to/from XML	2.9.10

Third-Party Component	Description	Version
jansson	Conversion DICOM to/from JSON	2.7
PICTools (aka Pegasus)	Image compression/decompression libraries from Accusoft	2.00.676

Important Information

Unicode encoding/decoding library ICU4C v49.1.2 was introduced in version 4.7.0 of the Merge DICOM C/C++ Toolkit. Since then, a number of vulnerabilities have been reported against v49.1.2 of the ICU4C libraries (see NIST NVD - ICU4C Vulnerabilities).

Although all the high severity vulnerabilities have been addressed and fixed in newer versions of the ICU4C libraries, an upgrade is not feasible on this platform due to compiler requirements that would break backward compatibility.

Toolkit versions 4.7.0 and later are impacted.

Due to these vulnerabilities, starting from release 5.15.0 of the toolkit, the Unicode conversion is turned off by default in the toolkit. OEM customers may choose to manually turn it back on, if they, after their own assessment for their specific application scenario, feel that it is safe to use.

To enable the Unicode conversion, set ENABLE_ICU4C_LIBRARY configuration setting to 'Yes' in the [MEDIA_PARMS] section in mergecom.pro. Alternatively, the `MC_Set_Bool_Config_Value()` API can be used for the same purpose.

If ICU4C is no longer used/required, OEM customers may choose to remove the library files completely from their application product distribution.

For the 64-bit Linux® on Intel® x86-64 platform specifically, OEM customers have the alternative to upgrade the build environment for their application to use a newer, more modern compiler, in which case they can upgrade to use the New Edition of the toolkit for 64-bit Linux® on Intel® x86 64, described in Chapter 8, which is built using the GCC compiler v9.4.0. This new edition uses ICU4C version 70.1.0, which addresses and resolves all the important vulnerabilities.

7.3. Miscellaneous Notes

7.3.1. Threading Support

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

7.3.2. Compression Support

The Merge DICOM Toolkit for Linux on Intel x64 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.

7.3.3. Unicode Support

The Merge DICOM Toolkit for Linux 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 do not 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.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

7.4. Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_lnx64	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 needed.)
	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.

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.
	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 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	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.

Directory	File	Description
	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 (library version 49.1.2).
	libicuuc.so.49	Unicode conversion library (library version 49.1.2).
	libpiclx20.so	Pegasus shared object which needs to be linked in when you compile your source.
	libxml2.so.2.9.10	XML parsing library (library version 2.9.10).
	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.
	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.

Directory	File	Description
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.

Chapter 8. 64-Bit Linux® on Intel® x86-64 (New Edition) (89-00350-00)

8.1. 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)	Ubuntu 20.04 or later.[2]
Software (development)	GCC 9.4.0.

[2]Merge DICOM Toolkit for Linux has been developed using Ubuntu 20.04. Although it should function under any distribution of Linux, Ubuntu Linux is the supported version if distribution specific problems arise.

8.2. The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for 64-Bit Linux 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 the static library 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 process 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.

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

Flag	Type	Description
`-fPIC`	Compile Time	Produce 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.

8.2.1. Third-Party Components Used

The third-party components used by the Merge DICOM Toolkit for 64-Bit Linux® on Intel® x86-64 (New Edition) are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	70.1.0
libxml2	Conversion DICOM to/from XML	2.10.2
jansson	Conversion DICOM to/from JSON	2.14
PICTools (aka Pegasus)	Image compression/decompression libraries from Accusoft	2.00.676

8.3. Miscellaneous Notes

8.3.1. Threading Support

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

8.3.2. Compression Support

The Merge DICOM Toolkit for Linux on Intel x64 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.

8.3.3. Unicode Support

The Merge DICOM Toolkit for Linux on Intel x64 supports Unicode conversion of DICOM defined character sets with and without code extensions. Two optional shared object libraries, libicuuc.so.70 and libicudata.so.70, 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.70 and libicudata.so.70 on 64-bit Linux platform:

Unicode conversion library	Dependency File	Description
libicuuc.so.70 libicudata.so.70	libpthread.so.1	Threading library
	libc.so.6	C library
	libstdc++.so.6	Standard C++ library
	libm.so.6	Math library
	libgcc_s.so.1	GCC library

8.4. Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_lnx64	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 needed.)
	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.
	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.

Directory	File	Description
	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.

Directory	File	Description
	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	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.70	Unicode conversion data (library version 70.1.0).
	libicuuc.so.70	Unicode conversion library (library version 70.1.0).
	libpiclx20.so	Pegasus shared object which needs to be linked in when you compile your source.
	libxml2.so.2.10.2	XML parsing library (library version 2.10.2).
	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.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

Chapter 9. 32-Bit Solaris™ 10 on Intel® x86 (008-91117)

9.1. 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 3.4.3.

9.2. 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 the static library 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 process 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	Produce 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.

9.2.1. Third-Party Components Used

The third-party components used by the Merge DICOM Toolkit for 32-Bit Solaris™ 10 on Intel® x86 are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	49.1.2 - read the Important Information below this table
libxml2	Conversion DICOM to/from XML	2.9.10
jansson	Conversion DICOM to/from JSON	2.7
PICTools (aka Pegasus)	Image compression/decompression libraries from Accusoft	2.00.676

Important Information

Unicode encoding/decoding library ICU4C v49.1.2 was introduced in version 4.7.0 of the Merge DICOM C/C++ Toolkit. Since then, a number of vulnerabilities have been reported against v49.1.2 of the ICU4C libraries (see NIST NVD - ICU4C Vulnerabilities).

Although all the high severity vulnerabilities have been addressed and fixed in newer versions of the ICU4C libraries, an upgrade is not feasible on this platform due to compiler requirements that would break backward compatibility.

Toolkit versions 4.7.0 and later are impacted.

Due to these vulnerabilities, starting from release 5.15.0 of the toolkit, the Unicode conversion is turned off by default in the toolkit. OEM customers may choose to manually turn it back on, if they, after their own assessment for their specific application scenario, feel that it is safe to use.

To enable the Unicode conversion, set ENABLE_ICU4C_LIBRARY configuration setting to 'Yes' in the [MEDIA_PARMS] section in mergecom.pro. Alternatively, the `MC_Set_Bool_Config_Value()` API can be used for the same purpose.

If ICU4C is no longer used/required, OEM customers may choose to remove the library files completely from their application product distribution.

9.3. Miscellaneous Notes

9.3.1. Threading Support

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

9.3.2. 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.

9.3.3. Unicode Support

The Merge DICOM Toolkit for 32-bit Solaris 10 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 do not 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.so.2	Linux library

9.4. 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 needed.)
	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.
	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.

Directory	File	Description
	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.

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	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.

Directory	File	Description
mc3lib	libicudata.so.49	Unicode conversion data (library version 49.1.2).
	libicuuc.so.49	Unicode conversion library (library version 49.1.2).
	libxml2.so.2.9.10	XML parsing library (library version 2.9.10)
	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.
	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.
	mrgcom3.dct	Runtime DICOM Data Dictionary File.
	mrgcom3.msg	Runtime Message Database File.

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

10.1. 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)	64-bit Solaris 10 Operating System.
Software (development)	GCC 3.4.3.

10.2. 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 the static library 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 process 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.

10.2.1. Third-Party Components Used

The third-party components used by the Merge DICOM Toolkit for 64-Bit Solaris™ 10 on Intel® x64 are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	49.1.2 - read the Important Information below this table
libxml2	Conversion DICOM to/from XML	2.9.10
jansson	Conversion DICOM to/from JSON	2.7

Important Information

Unicode encoding/decoding library ICU4C v49.1.2 was introduced in version 4.7.0 of the Merge DICOM C/C++ Toolkit. Since then, a number of vulnerabilities have been reported against v49.1.2 of the ICU4C libraries (see NIST NVD - ICU4C Vulnerabilities).

Although all the high severity vulnerabilities have been addressed and fixed in newer versions of the ICU4C libraries, an upgrade is not feasible on this platform due to compiler requirements that would break backward compatibility.

Toolkit versions 4.7.0 and later are impacted.

Due to these vulnerabilities, starting from release 5.15.0 of the toolkit, the Unicode conversion is turned off by default in the toolkit. OEM customers may choose to manually turn it back on, if they, after their own assessment for their specific application scenario, feel that it is safe to use.

To enable the Unicode conversion, set ENABLE_ICU4C_LIBRARY configuration setting to 'Yes' in the [MEDIA_PARMS] section in mergecom.pro. Alternatively, the `MC_Set_Bool_Config_Value()` API can be used for the same purpose.

If ICU4C is no longer used/required, OEM customers may choose to remove the library files completely from their application product distribution.

10.3. Miscellaneous Notes

10.3.1. Threading Support

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

10.3.2. 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.

10.3.3. 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 do not 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

10.4. 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 needed.)
	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.
	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.

Directory	File	Description
	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.

Directory	File	Description
	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 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	libicudata.so.49	Unicode conversion data (library version 49.1.2).
	libicuuc.so.49	Unicode conversion library (library version 49.1.2).
	libxml2.so.2.9.10	XML parsing library (library version 2.9.10).
	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.

Chapter 11. 32-Bit Solaris™ 8 on SPARC® using Sun™ Compiler (008-91119)

11.1. 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.

11.2. 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 the static library 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 process 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.

11.2.1. Third-Party Components Used

The third-party components used by the Merge DICOM Toolkit for 32-Bit Solaris™ 8 on SPARC® using Sun™ Compiler are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	49.1.2 - read the Important Information below this table
libxml2	Conversion DICOM to/from XML	2.9.10
jansson	Conversion DICOM to/from JSON	2.7
PICTools (aka Pegasus)	Image compression/decompression libraries from Accusoft	2.00.676

Important Information

Unicode encoding/decoding library ICU4C v49.1.2 was introduced in version 4.7.0 of the Merge DICOM C/C++ Toolkit. Since then, a number of vulnerabilities have been reported against v49.1.2 of the ICU4C libraries (see NIST NVD - ICU4C Vulnerabilities).

Although all the high severity vulnerabilities have been addressed and fixed in newer versions of the ICU4C libraries, an upgrade is not feasible on this platform due to compiler requirements that would break backward compatibility.

Toolkit versions 4.7.0 and later are impacted.

Due to these vulnerabilities, starting from release 5.15.0 of the toolkit, the Unicode conversion is turned off by default in the toolkit. OEM customers may choose to manually turn it back on, if they, after their own assessment for their specific application scenario, feel that it is safe to use.

To enable the Unicode conversion, set ENABLE_ICU4C_LIBRARY configuration setting to 'Yes' in the [MEDIA_PARMS] section in mergecom.pro. Alternatively, the `MC_Set_Bool_Config_Value()` API can be used for the same purpose.

If ICU4C is no longer used/required, OEM customers may choose to remove the library files completely from their application product distribution.

11.3. Miscellaneous Notes

11.3.1. Threading Support

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

11.3.2. 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.

11.3.3. 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 do not 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

11.4. Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	readme	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 needed.)
	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.
	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.

Directory	File	Description
	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.

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	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.

Directory	File	Description
mc3lib	libicudata.so.49	Unicode conversion data (library version 49.1.2)
	libicuuc.so.49	Unicode conversion library (library version 49.1.2)
	libxml2.so.2.9.10	XML parsing library (library version 2.9.10)
	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.
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.

Chapter 12. 32-Bit Solaris™ 8 on SPARC® using GCC Compiler (008-91130)

12.1. 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

12.2. 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 the static library 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 process 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	Produce 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.

12.2.1. Third-Party Components Used

The third-party components used by the Merge DICOM Toolkit for 32-Bit Solaris™ 8 on SPARC® using GCC Compiler are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	49.1.2 - read the Important Information below this table
libxml2	Conversion DICOM to/from XML	2.9.10
jansson	Conversion DICOM to/from JSON	2.7
PICTools (aka Pegasus)	Image compression/decompression libraries from Accusoft	2.00.676

Important Information

Unicode encoding/decoding library ICU4C v49.1.2 was introduced in version 4.7.0 of the Merge DICOM C/C++ Toolkit. Since then, a number of vulnerabilities have been reported against v49.1.2 of the ICU4C libraries (see NIST NVD - ICU4C Vulnerabilities).

Although all the high severity vulnerabilities have been addressed and fixed in newer versions of the ICU4C libraries, an upgrade is not feasible on this platform due to compiler requirements that would break backward compatibility.

Toolkit versions 4.7.0 and later are impacted.

Due to these vulnerabilities, starting from release 5.15.0 of the toolkit, the Unicode conversion is turned off by default in the toolkit. OEM customers may choose to manually turn it back on, if they, after their own assessment for their specific application scenario, feel that it is safe to use.

To enable the Unicode conversion, set ENABLE_ICU4C_LIBRARY configuration setting to 'Yes' in the [MEDIA_PARMS] section in mergecom.pro. Alternatively, the `MC_Set_Bool_Config_Value()` API can be used for the same purpose.

If ICU4C is no longer used/required, OEM customers may choose to remove the library files completely from their application product distribution.

12.3. Miscellaneous Notes

12.3.1. Threading Support

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

12.3.2. 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.

12.3.3. 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 do not 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
	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

12.4. 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 needed.)
	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.
	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.
	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.

Directory	File	Description
	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 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	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.

Directory	File	Description
mc3lib	libicudata.so.49	Unicode conversion data (library version 49.1.2).
	libicuuc.so.49	Unicode conversion library (library version 49.1.2).
	libpicu20.so	Pegasus shared object which needs to be linked in when you compile your source.
	libxml2.so.2.9.10	XML parsing library (library version 2.9.10).
	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.

Chapter 13. 32-Bit Mac OS® X with Universal Binaries (008-91301)

13.1. 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).

13.2. The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for 32-Bit Mac OS® X with Universal Binaries is provided as 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 the static library 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 process 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.

13.2.1. Third-Party Components Used

The third-party components used by the Merge DICOM Toolkit for 32-Bit Mac OS® X with Universal Binaries are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	49.1.2 - read the Important Information below this table
libxml2	Conversion DICOM to/from XML	2.9.10
jansson	Conversion DICOM to/from JSON	2.7

Important Information

Unicode encoding/decoding library ICU4C v49.1.2 was introduced in version 4.7.0 of the Merge DICOM C/C++ Toolkit. Since then, a number of vulnerabilities have been reported against v49.1.2 of the ICU4C libraries (see NIST NVD - ICU4C Vulnerabilities).

Although all the high severity vulnerabilities have been addressed and fixed in newer versions of the ICU4C libraries, an upgrade is not feasible on this platform due to compiler requirements that would break backward compatibility.

Toolkit versions 4.7.0 and later are impacted.

Due to these vulnerabilities, starting from release 5.15.0 of the toolkit, the Unicode conversion is turned off by default in the toolkit. OEM customers may choose to manually turn it back on, if they, after their own assessment for their specific application scenario, feel that it is safe to use.

To enable the Unicode conversion, set ENABLE_ICU4C_LIBRARY configuration setting to 'Yes' in the [MEDIA_PARMS] section in mergecom.pro. Alternatively, the `MC_Set_Bool_Config_Value()` API can be used for the same purpose.

If ICU4C is no longer used/required, OEM customers may choose to remove the library files completely from their application product distribution.

13.3. Miscellaneous Notes

13.3.1. 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.

13.3.2. Compression Support

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

13.3.3. 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 do not need to deploy the two shared objects and their dependency files.

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

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

13.4. 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 needed.)
	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.
	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.

Directory	File	Description
	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.

Directory	File	Description
	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 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	libicudata49.dylib	Unicode conversion data (library version 49.1.2).
	libicudata49.dylib.ppc	Unicode conversion data for Power PC based machine. This file must be renamed to libicudata.49.dylib before running application (library version 49.1.2).
	libicuuc.49.dylib	Unicode conversion library (library version 49.1.2).
	libxml2.2.dylib	XML parsing library (library version: 2.9.10)
	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.

Chapter 14. 64-Bit Mac OS® X on Intel® x64 (008-91303)

14.1. 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).

14.2. 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 process 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.

14.2.1. Third-Party Components Used

The third-party components used by the Merge DICOM Toolkit for 64-Bit Mac OS® X on Intel® x64 are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	49.1.2 - read the Important Information below this table
libxml2	Conversion DICOM to/from XML	2.9.10
jansson	Conversion DICOM to/from JSON	2.7

Important Information

Unicode encoding/decoding library ICU4C v49.1.2 was introduced in version 4.7.0 of the Merge DICOM C/C++ Toolkit. Since then, a number of vulnerabilities have been reported against v49.1.2 of the ICU4C libraries (see NIST NVD - ICU4C Vulnerabilities).

Although all the high severity vulnerabilities have been addressed and fixed in newer versions of the ICU4C libraries, an upgrade is not feasible on this platform due to compiler requirements that would break backward compatibility.

Toolkit versions 4.7.0 and later are impacted.

Due to these vulnerabilities, starting from release 5.15.0 of the toolkit, the Unicode conversion is turned off by default in the toolkit. OEM customers may choose to manually turn it back on, if they, after their own assessment for their specific application scenario, feel that it is safe to use.

To enable the Unicode conversion, set ENABLE_ICU4C_LIBRARY configuration setting to 'Yes' in the [MEDIA_PARMS] section in mergecom.pro. Alternatively, the `MC_Set_Bool_Config_Value()` API can be used for the same purpose.

If ICU4C is no longer used/required, OEM customers may choose to remove the library files completely from their application product distribution.

For the 64-Bit Mac OS® X on Intel® x64 platform specifically, OEM customers have the alternative to upgrade the build environment for their application to use a newer, more modern compiler, in which case they can upgrade to use the New Edition of the toolkit for 64-bit macOS® with Universal Binaries on Intel® x64 and ARM64, described in Chapter 15, which is built using the Clang compiler v14.0.0. This new edition uses ICU4C version 70.1.0, which addresses and resolves all the important vulnerabilities.

14.3. Miscellaneous Notes

14.3.1. Threading Support

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

14.3.2. 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.

14.3.3. 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 do not need to deploy the two shared objects and their dependency files.

Dependency files of libicuuc.49.dylib and libicudata.49.dylib for 64-bit MacOS X on Intel platform:

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

14.4. 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 needed.)
	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.
	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.

Directory	File	Description
	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.

Directory	File	Description
	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 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	libicudata.49.dylib	Unicode conversion data (library version 49.1.2).
	libicuuc.49.dylib	Unicode conversion library (library version 49.1.2).
	libxml2.2.dylib	XML parsing library (library version: 2.9.10).
	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.

Chapter 15. 64-Bit macOS® with Universal Binaries on Intel® x64 and ARM64 M1 (New Edition) (89 00359 00)

15.1. Supported Configurations

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

Category	Requirement
Hardware	• macOS supported 64-bit Intel x86 hardware • macOS supported 64-bit ARM64 M1/M2 hardware
Software (base)	macOS v12.5.1 or later.
Software (development)	Apple clang version 14.0.0 (clang-1400.0.29.102)

15.2. The Merge DICOM Toolkit Libraries

The Merge DICOM Toolkit for 64-Bit macOS® on Intel® x64 and ARM64 M1/M2 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 x86_64`	Support Intel based macOS computers
`-arch arm64`	Support ARM64 based macOS 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 process address space.

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

Compile Options	Description
`-arch x86_64`	Support Intel based macOS computers
`-arch arm64`	Support ARM64 based macOS computers.
`-m64`	Generate code for a 64-bit environment
`-O2`	Enable optimization.
`-D_REENTRANT`	Produce reentrant code that is thread safe.
`-dynamic`	Compile to a dynamic library

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.

15.2.1. Third-Party Components Used

The third-party components used by the Merge DICOM Toolkit for 64-Bit Mac OS® X on Intel® x64 are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	70.1.0
libxml2	Conversion DICOM to/from XML	2.10.2
jansson	Conversion DICOM to/from JSON	2.14

15.3. Miscellaneous Notes

15.3.1. Threading Support

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

15.3.2. Compression Support

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

15.3.3. Unicode Support

The Merge DICOM Toolkit for 64-bit macOS® supports Unicode conversion of DICOM defined character sets with and without code extensions. Two optional shared object libraries, libicuuc.70.dylib and libicudata.70.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 do not need to deploy the two shared objects and their dependency files.

Dependency files of libicuuc.49.dylib and libicudata.49.dylib for 64-bit MacOS X on Intel platform:

Unicode conversion library	Dependency File	Description
libicuuc.70.dylib libicudata.70.dylib	libc++.1.dylib	Standard C++ library
	libSystem.B.dylib	Mac OS X library

15.4. Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_macos64ne	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 needed.)
	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.
	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.
	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.

Directory	File	Description
	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 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.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	libicudata.70.dylib	Unicode conversion data (library version 70.1.0)
	libicuuc.70.dylib	Unicode conversion library(library version 70.1.0)
	libxml2.2.dylib	XML parsing library (library version: 2.10.2)

Directory	File	Description
	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.

Chapter 16. 32-Bit Android on ARMv7-A (008-91998)

16.1. 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)

16.2. 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 the static library 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 process 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.

Flag	Type	Description
`-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.

16.2.1. Third-Party Components Used

The third-party components used by Merge DICOM Toolkit for 32-Bit Android on ARMv7-A are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	49.1.2 - read the Important Information below this table
libxml2	Conversion DICOM to/from XML	2.9.10
jansson	Conversion DICOM to/from JSON	2.7
AIMTools (aka Pegasus)	Image decompression libraries from Accusoft	2.00.042

Important Information

Unicode encoding/decoding library ICU4C v49.1.2 was introduced in version 4.7.0 of the Merge DICOM C/C++ Toolkit. Since then, a number of vulnerabilities have been reported against v49.1.2 of the ICU4C libraries (see NIST NVD - ICU4C Vulnerabilities).

Although all the high severity vulnerabilities have been addressed and fixed in newer versions of the ICU4C libraries, an upgrade is not feasible on this platform due to compiler requirements that would break backward compatibility.

ICU4C v49.1.2 was ported to the toolkit for 32-Bit Android on ARMv7-A in version 5.7.0. Toolkit versions 5.7.0 and later are impacted.

Due to these vulnerabilities, starting from release 5.15.0 of the toolkit, the Unicode conversion is turned off by default in the toolkit. OEM customers may choose to manually turn it back on, if they, after their own assessment for their specific application scenario, feel that it is safe to use.

To enable the Unicode conversion, set ENABLE_ICU4C_LIBRARY configuration setting to 'Yes' in the [MEDIA_PARMS] section in mergecom.pro. Alternatively, the `MC_Set_Bool_Config_Value()` API can be used for the same purpose.

If ICU4C is no longer used/required, OEM customers may choose to remove the library files completely from their application product distribution.

16.3. Miscellaneous Notes

16.3.1. Threading Support

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

16.3.2. 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 (**www.accusoft.com**) and configuring that license in our `mergecom.pro` configuration file.

Built-in RLE compression/decompression is also available.

16.3.3. 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 do not 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
	libm.so	Math library

16.4. 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 needed.)
	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.
	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.

Directory	File	Description
	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.

Directory	File	Description
	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	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	libicudata.so.49	Unicode conversion data (library version 49.1.2).
	libicuuc.so.49	Unicode conversion library(library version 49.1.2).
	libmc3adv.a	Merge DICOM Toolkit software static library.
	libmc3adv.so	Merge DICOM Toolkit software shared object.
	libpicd20.a	Pegasus static library.
	libpicd20.so	Pegasus shared object which needs to be linked in when you compile your source.
	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.

Directory	File	Description
	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 (library version: 2.9.10).
	libxml2.so	libxml2 shared object (library version: 2.9.10).
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.

Chapter 17. 64-Bit Android on ARMv8-A (89-00161-00)

17.1. 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)

17.2. 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 the static library 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 process 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.

Flag	Type	Description
-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.

17.2.1. Third-Party Components Used

The third-party components used by Merge DICOM Toolkit for 64-Bit Android on ARMv8-A are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	49.1.2 - read the Important Information below this table
libxml2	Conversion DICOM to/from XML	2.9.10
jansson	Conversion DICOM to/from JSON	2.7

Important Information

Unicode encoding/decoding library ICU4C v49.1.2 was introduced in version 4.7.0 of the Merge DICOM C/C++ Toolkit. Since then, a number of vulnerabilities have been reported against v49.1.2 of the ICU4C libraries (see NIST NVD - ICU4C Vulnerabilities).

Although all the high severity vulnerabilities have been addressed and fixed in newer versions of the ICU4C libraries, an upgrade is not feasible on this platform due to compiler requirements that would break backward compatibility.

ICU4C v49.1.2 was ported to the toolkit for 64-Bit Android on ARMv8-A in version 5.7.0. Toolkit versions 5.7.0 and later are impacted.

Due to these vulnerabilities, starting from release 5.15.0 of the toolkit, the Unicode conversion is turned off by default in the toolkit. OEM customers may choose to manually turn it back on, if they, after their own assessment for their specific application scenario, feel that it is safe to use.

To enable the Unicode conversion, set ENABLE_ICU4C_LIBRARY configuration setting to 'Yes' in the [MEDIA_PARMS] section in mergecom.pro. Alternatively, the MC_Set_Bool_Config_Value() API can be used for the same purpose.

If ICU4C is no longer used/required, OEM customers may choose to remove the library files completely from their application product distribution.

17.3. Miscellaneous Notes

17.3.1. Threading Support

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

17.3.2. 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.

17.3.3. 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 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 do not need to deploy the two shared objects and their dependency files.

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

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

17.4. 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.

Directory	File	Description
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 needed.)
	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.
	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.

Directory	File	Description
	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 Database Manual.
	Platform.pdf	This document.

Directory	File	Description
	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.
	mcstatus.h	Library include file for use with libmc3adv.a.
	mergecom.h	Library include file for use with libmc3adv.a.
mc3lib	libicudata.so.49	Unicode conversion data (library version 49.1.2).
	libicuuc.so.49	Unicode conversion library (library version 49.1.2).
	libmc3adv.a	Merge DICOM Toolkit software static library.
	libmc3adv.so	Merge DICOM Toolkit software shared object.
	libxml2.la	libtool library file for libxml2 (library version v2.9.10).
	libxml2.so	libxml2 shared object (library version: 2.9.10).
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.

Chapter 18. 64-Bit iOS on ARMv8-A (008-91990)

18.1. 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)

18.2. 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 the static library 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

18.2.1. Third-Party Components Used

The third-party components used by Merge DICOM Toolkit for 64-Bit iOS on ARMv8-A are listed in the following table.

Third-Party Component	Description	Version
ICU4C	Unicode encoding/decoding	49.1.2 - read the Important Information below this table

Third-Party Component	Description	Version
libxml2	Conversion DICOM to/from XML	2.9.10
jansson	Conversion DICOM to/from JSON	2.7

Important Information

Unicode encoding/decoding library ICU4C v49.1.2 was introduced in version 4.7.0 of the Merge DICOM C/C++ Toolkit. Since then, a number of vulnerabilities have been reported against v49.1.2 of the ICU4C libraries (see NIST NVD - ICU4C Vulnerabilities).

Although all the high severity vulnerabilities have been addressed and fixed in newer versions of the ICU4C libraries, an upgrade is not feasible on this platform due to compiler requirements that would break backward compatibility.

ICU4C v49.1.2 was ported to the toolkit for 64-Bit iOS on ARMv8-A in version 5.8.0. Toolkit versions 5.8.0 and later are impacted.

Due to these vulnerabilities, starting from release 5.15.0 of the toolkit, the Unicode conversion is turned off by default in the toolkit. OEM customers may choose to manually turn it back on, if they, after their own assessment for their specific application scenario, feel that it is safe to use.

To enable the Unicode conversion, set ENABLE_ICU4C_LIBRARY configuration setting to 'Yes' in the [MEDIA_PARMS] section in mergecom.pro. Alternatively, the `MC_Set_Bool_Config_Value()` API can be used for the same purpose.

If ICU4C is no longer used/required, OEM customers may choose to remove the library files completely from their application product distribution.

18.3. Miscellaneous Notes

18.3.1. Threading Support

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

18.3.2. 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.

18.3.3. Unicode Support

The Merge DICOM Toolkit for 64-bit iOS 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 do not need to deploy the two shared objects and their dependency files.

Dependency files of libicuuc.49.dylib and libicudata.49.dylib for 64-bit iOS platform:

Unicode conversion library	Dependency File	Description
libicuuc.49.dylib libicudata.49.dylib	libc++.1.dylib	Standard C++ library
	libSystem.B.dylib	Mac OS X library

18.4. 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.
	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 needed.)
	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.
	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.

Directory	File	Description
	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.

Directory	File	Description
	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 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	libicudata.49.dylib	Unicode conversion data.
	libicuuc.49.dylib	Unicode conversion library.
	libxml2.2.dylib	XML parsing library (library version: 2.9.10).
	libxml2.a	XML parsing library (library version: 2.9.10).
	mc3adv.a	Merge DICOM Toolkit software static library.
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.

Chapter 19. 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.

Chapter 20. 64-Bit Linux on ARMv8-A (89-00163-00)

20.1. 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

20.2. 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 the static library 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 process 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	Requirement
`-O3`	Compile Time	Optimization level.
`-D_REENTRANT`	Compile Time	Produce reentrant code that is thread safe.

20.2.1. Third-Party Components Used

The Merge DICOM Toolkit for 64-Bit Linux on ARMv8-A does not use any third party components.

20.3. Miscellaneous Notes

20.3.1. Threading Support

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

20.3.2. 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.

20.4. Files

The following files are contained in the Merge DICOM Toolkit:

Directory	File	Description
	read_me	Information on this release of the toolkit.
	0_lnxARMv8	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 needed.)
	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.
	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.
	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.

Directory	File	Description
	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 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.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	mc3adv.a	Merge DICOM Toolkit software static library.
	mc3adv.so	Merge DICOM Toolkit software shared object.

Directory	File	Description
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.