



Merge DICOM Toolkit™ V. 5.9.0

C/C++ SAMPLE APPLICATION GUIDE

Merge Healthcare
900 Walnut Ridge Drive
Hartland, WI 53029
USA

© Copyright 2019 Merge Healthcare Incorporated, an IBM Company.

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

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

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

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

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

U.S. GOVERNMENT RESTRICTED RIGHTS:

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

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



Manufacturer's Address

Merge Healthcare Incorporated
900 Walnut Ridge Drive
Hartland, WI 53029

For assistance, please contact Merge Healthcare Customer Support:

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

Part	Date	Revision	Description
COM-3672	April 2019	1.0	Updated bi-annually

Contents

OVERVIEW	10
Documentation Road Map.....	10
Conventions.....	11
GENERAL.....	12
Environment	12
Files	12
THE STORAGE AND STORAGE COMMITMENT SERVICE CLASSES	14
Service Definition	14
Storage Service Class User Requirements.....	14
Storage Service Class Provider Requirements	15
Storage Commitment Service Class User Requirements	15
Storage Commitment Service Class Provider Requirements.....	16
Summary.....	16
Composite Services Supported.....	16
Commands Supported	22
C_STORE_RQ	22
C_STORE_RSP	22
Valid Messages	22
STORAGE SERVICE CLASS SAMPLE APPLICATIONS.....	23
Configuration	23
SCU Configuration.....	23
SCP Configuration	23
General Configuration.....	24
Sample SCU.....	24
Overview of Program Operation	24
Asynchronous Operations	28
Sample SCP	29
Overview of Program Operation	29
Asynchronous Operations	32
THE QUERY/RETRIEVE SERVICE CLASS	32
Service Definition	32
Commands Supported	34
C-FIND Command	34
C-MOVE Command.....	34
C-GET Command	34
Extended Negotiation for Relational Queries	34
Valid Messages	35
QUERY/RETRIEVE SERVICE CLASS SAMPLE APPLICATIONS	35
Configuration	37

Sample Client (User)	37
Running the SCU	37
The SCU User Interface	39
DICOM Related Functions	39
User Interface Functions	42
Sample Server (Provider)	44
Running the SCP	44
DICOM Related Functions	45
Other Functions	48
Sample Application Include File: qr.h	50
THE QUERY/RETRIEVE C-GET SERVICE SAMPLE APPLICATIONS	50
Composite Services Supported by the GET SCU/SCP	51
Sample Client (User)	52
Running the SCU	52
The SCU User Interface	52
Sample Server (Provider)	53
Running the SCP	53
THE BASIC PRINT SERVICE CLASSES	54
Service Definition	54
Print Service Class User Requirements	54
Print Service Class Provider Requirements	55
Normalized Services Supported	56
Commands Supported	57
Basic Grayscale Print Management Meta SOP Class	59
Valid Messages	61
PRINT SERVICE CLASS SAMPLE APPLICATIONS	62
Configuration	62
Sample SCU	63
Overview of Program Operation	63
Sample SCP	67
Overview of Program Operation	67
THE MODALITY WORKLIST SERVICE CLASS	69
Service Definition	69
Composite Services Supported by the SCU and SCP	69
Commands Supported	69
Valid Messages	70
MODALITY WORKLIST AND MODALITY PERFORMED PROCEDURE STEP SAMPLE APPLICATIONS	71
Configuration	72
Sample Client (User)	72

Running the SCU	72
The SCU User Interface	73
Sample Server (Provider).....	75
Running the SCP	76
THE MEDIA STORAGE SERVICE CLASS	78
Service Definition	78
File-set Creator Requirements.....	79
File-set Reader Requirements.....	79
File-set Updator Requirements.....	80
Composite Services Supported.....	80
Valid Messages	83
MEDIA STORAGE SAMPLE APPLICATION	84
Configuration	84
FSU Configuration	84
General Configuration.....	84
Sample FSU	84
Overview of Program Operation	84
DUPLICATE SAMPLE APPLICATION	86
Configuration	87
General Configuration.....	87
Sample Duplicate	87
Overview of Program Operation	87
MPEG CONVERSION SAMPLE APPLICATION.....	89
Configuration	90
General Configuration.....	90
MPEG Conversion.....	90
Overview of Program Operation	90
COMP SAMPLE APPLICATION.....	90
Configuration	90
Overview of Program Operation	91
APPENDIX A: STORAGE SCU CONFORMANCE STATEMENT	92
0. Introduction.....	92
1. Implementation model.....	92
1.1 Application data flow diagram.....	92
1.2 Functional definition of Application Entity (AE).....	92
1.3 Sequencing of real-world activities	92
2. AE specifications	93
3. Communication profiles.....	97
3.1 Supported Communication Stacks	97
3.2 TCP/IP Stack	97
4. Extensions/specializations/privatizations	98

4.1 Standard extended/specialized/private SOPs	98
4.2 Private Transfer Syntaxes	98
5. Configuration	98
5.1 AE title/presentation address mapping.....	98
5.2 Configurable parameters	98
6. Support of extended character sets	98
APPENDIX B: STORAGE SCP CONFORMANCE STATEMENT.....	99
0. Introduction.....	99
1. Implementation model.....	99
1.1 Application data flow diagram.....	99
1.2 Functional definition of Application Entity (AE).....	99
1.3 Sequencing of real-world activities	100
2. AE specifications	100
2.1 AE Specification for MERGE_STORE_SCP	100
3. Profiles.....	107
3.1 Supported Communication Stacks	107
3.2 TCP/IP Stack	107
4. Extensions/specializations/privatizations	107
4.1 Standard extended/specialized/private SOPs	107
4.2 Private Transfer Syntaxes	107
5. Configuration	108
5.1 MERGE_STORE_SCP Configuration files.....	108
6. Support of extended character sets	108
APPENDIX C: QUERY/RETRIEVE SCU CONFORMANCE STATEMENT	109
0. Introduction.....	109
1. Implementation Model.....	109
1.1 Application data flow diagram.....	109
1.2 Functional definition of Application Entity (AE).....	109
2. AE Specifications	110
2.1 AE specification for MERGE_QR_SCU.....	110
3. Profiles.....	116
3.1 Supported Communication Stacks	116
3.2 TCP/IP Stack	116
4. Extensions/specializations/privatizations	116
4.1 Standard extended/specialized/private SOPs	116
4.2 Private Transfer Syntaxes	116
5. Configuration	117
5.1 MERGE_QR_SCU Configuration Files	117
6. Support of extended character sets	117
APPENDIX D: QUERY/RETRIEVE SCP CONFORMANCE STATEMENT	118
Introduction.....	118

1. Implementation Model	118
1.1 Application data flow diagram	118
1.2 Functional definition of Application Entity (AE)	118
2. AE Specifications	119
2.1 AE specification for MERGE_QR_SCP	119
3. Profiles	128
3.1 Supported Communication Stacks	128
3.2 TCP/IP Stack	128
4. Extensions/specializations/privatizations	128
4.1 Standard extended/specialized/private SOPs	128
4.2 Private Transfer Syntaxes	129
5. Configuration	129
5.1 MERGE_QR_SCP Configuration Files	129
6. Support of extended character sets	129
APPENDIX E: PRINT SCU CONFORMANCE STATEMENT	130
0. Introduction	130
1. Implementation model	130
1.1 Application data flow diagram	130
1.2 Functional definition of Application Entity (AE)	131
1.3 Sequencing of real-world activities	131
2. AE specifications	131
2.1 AE Specification for MERGE_PRINT_SCU	131
3. Profiles	136
3.1 Supported Communication Stacks	136
3.2 TCP/IP Stack	136
4. Extensions/specializations/privatizations	136
4.1 Standard extended/specialized/private SOPs	136
4.2 Private Transfer Syntaxes	136
5. Configuration	137
5.1 MERGE_PRINT_SCU Configuration files	137
6. Support of extended character sets	137
APPENDIX F: PRINT SCP CONFORMANCE STATEMENT	138
0. Introduction	138
1. Implementation model	138
1.1 Application data flow diagram	138
1.2 Functional definition of Application Entity (AE)	139
1.3 Sequencing of real-world activities	139
2. AE specifications	139
2.1 AE Specification for MERGE_PRINT_SCP	139
3. Profiles	146
3.1 Supported Communication Stacks	146
3.2 TCP/IP Stack	146

4. Extensions/specializations/privatizations	146
4.1 Standard extended/specialized/private SOPs	146
4.2 Private Transfer Syntaxes	146
5. Configuration	146
5.1 MERGE_PRINT_SCP Configuration Files	146
6. Support of extended character sets	147
APPENDIX G: WORK_SCU CONFORMANCE STATEMENT	148
0. Introduction.....	148
1. Implementation Model.....	148
1.1 Application data flow diagram.....	148
1.2 Functional definition of Application Entity (AE).....	149
2 AE Specifications	149
2.1 AE specification for MERGE_WORK_SCU.....	149
3 Profiles.....	151
3.1 Supported Communication Stacks	151
3.2 TCP/IP Stack	151
4. Extensions/specializations/privatizations	151
4.1 Standard extended/specialized/private SOPs	151
4.2 Private Transfer Syntaxes	151
5. Configuration	151
5.1 MERGE_WORK_SCU AE Configuration Files.....	151
6. Support of extended character sets	152
APPENDIX H: WORK_SCP CONFORMANCE STATEMENT	153
0. Introduction.....	153
1. Implementation Model.....	153
1.1 Application data flow diagram.....	153
1.2 Functional definition of Application Entity (AE).....	153
2 AE Specifications	154
2.1 AE specification for MERGE_WORK_SCP	154
3. Profiles.....	157
3.1 Supported Communication Stacks	157
3.2 TCP/IP Stack	157
4. Extensions/specializations/privatizations	157
4.1 Standard extended/specialized/private SOPs	157
4.2 Private Transfer Syntaxes	157
5. Configuration	157
6. Support of extended character sets	158
APPENDIX I: MEDIA FSU CONFORMANCE STATEMENT	159
0. Introduction.....	159
1. Implementation Model.....	159
1.1 Application Data Flow Diagram	159

1.2 Functional Definition of Application Entity	160
1.3 Sequencing of real-world activities	161
1.4 File Meta Information Options	161
2. AE Specifications	161
2.1 AE Specification for MERGE_MEDIA.....	161
2.2 AE Specification for MERGE_MEDIA_FSU	167
3. Profiles.....	167
3.1 Supported Communication Stacks	167
3.2 TCP/IP Stack	168
3.3 Augmented and Private Application Profiles	168
4. Extensions, Specializations, and Privatizations of SOP Classes and Transfer Syntaxes.....	168
5. Configuration	168
5.1 MERGE_MEDIA_FSU and MERGE_MEDIA Configuration Files	168
5.2 File Meta Information for MERGE_MEDIA_FSU AE	169
6. Support of extended character sets	169

Overview

This manual is designed to give the user of the Merge DICOM Toolkit an example of how to implement several of the DICOM Service Classes. Each DICOM 3.0 Service Class is defined and thoroughly explained in the Digital Imaging and Communications in Medicine (DICOM) standard Version 3.0, Part 4. Although only the Storage, Query/Retrieve, Print, Modality Worklist and Media Service Classes are discussed in this guide, Merge DICOM Toolkit supports other DICOM service classes.

The examples described herein use the ANSI-C programming language along with the Merge DICOM Toolkit library of ANSI-C functions to implement each of the Service Classes in a limited yet instructional fashion.

Documentation Road Map

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

Read Me FIRST!

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 Guides you should be comfortable with the material in the User's Manual.

The *Reference Manual* is where you go for detailed information on the DICOM Toolkit. This includes the Application Programming Interface (API), toolkit configuration, the runtime object database, and status logging. The Reference Manual also includes a general DICOM conformance statement for the toolkit.

The *DICOM Message Dictionary Specification and Generation Manual* is an optional extension that describes how to privately extend, and generate, Merge DICOM Toolkit DICOM dictionary and message information binary files. This manual is provided along with the executables for your specific platform that generate the binary files.

Sample Applications

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

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

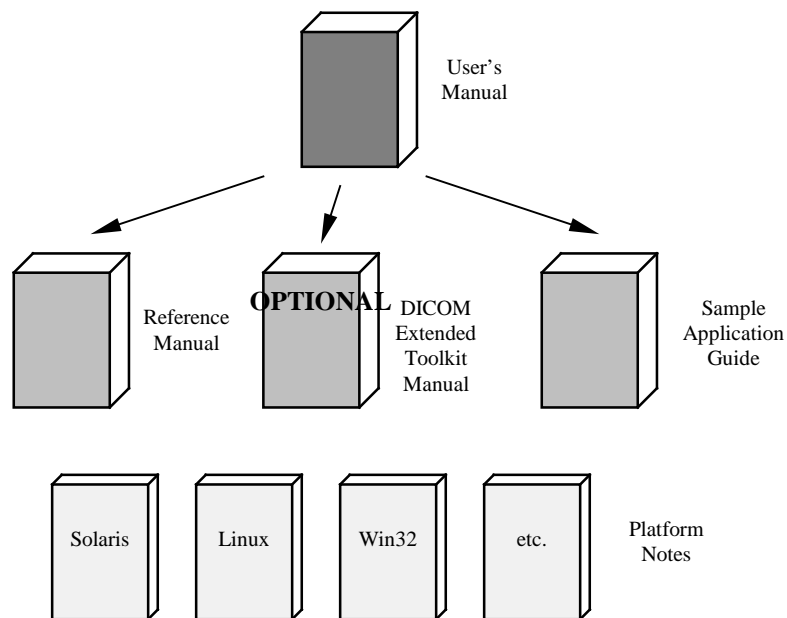


Figure 1: Merge DICOM Toolkit Documentation Roadmap

Conventions

This manual follows a few formatting conventions.

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

Sample Margin Note

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

Performance Tuning

Portions of the document that can relate directly to the performance of your application are marked with the special margin note **Performance Tuning**.

Sample commands appear in **bold courier** font, while 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.

General

Environment

All Merge DICOM Toolkit applications require an environment variable named **MERGE_INI** to be defined and set. This variable informs the application as to the name of the initialization file discussed below.

How **MERGE_INI** is set depends on which shell is being used. If the user is running the Bourne or Korn shell, the command,

```
$ MERGE_INI=./merge.ini; export MERGE_INI
```

will accomplish this assuming that `merge.ini` is the name of the initialization file. If the user is running the C shell, the command

```
% setenv MERGE_INI ./merge.ini
```

will accomplish the same. The general formats for the above two commands are:

```
$ MERGE_INI=./fname.ini; export MERGE_INI      (Bourne Shell)
```

```
% setenv MERGE_INI ./fname.ini      (C Shell)
```

where `fname` is the name of the file containing the initialization information.

Files

The following is a description of the files used to implement the toolkit sample applications:

Table 1: Source Code Files

Filename	Description
<code>makefile</code>	The makefile used to generate all Merge DICOM Toolkit sample applications.
<code>dictionary.h</code>	The header file that contains the dictionary definitions.
<code>mc3msg.h</code>	The header file that contains definitions for the message objects functionality.
<code>mcstatus.h</code>	The header file that contains definitions for statuses and return codes.
<code>mergecom.h</code>	The header file that contains definitions for the merge library network functionality.
<code>stor_scp.c</code>	The C source code file that contains the source code for implementation of the Storage and Storage Commitment Service Class Provider.

Filename	Description
stor_scu.c	The C source code file that contains the source code for implementation of the Storage and Storage Commitment Service Class User.
prnt_svc.h	The header file that contains definitions for the Print applications.
prnt_scp.c	The C source code file that contains the source code for implementation of the Print Service Class Provider.
prnt_scu.c	The C source code file that contains the source code for implementation of the Print Service Class User.
med_fsu.c	The C source code file that contains the source code for implementation of the Media File Set Updator.
qr.h	The header file that contains definitions for the Modality Worklist and Query/Retrieve applications.
qr_scp.c	The C source code file that contains the source code for implementation of the Query/Retrieve Service Class Provider.
qr_scu.c	The C source code file that contains the source code for implementation of the Query/Retrieve Service Class User.
work_scp.c	The C source code file that contains the source code for the implementation of the Modality Worklist and Performed Procedure Step Service Class Provider.
work_scu.c	The C source code file that contains the source code for the implementation of the Modality Worklist and Performed Procedure Step Service Class User.
workdata.c	The C source code file that contains a database implementation used by the Modality Worklist and Performed Procedure Step Service Class Provider
workdata.h	The header file that contains definitions for the Modality Worklist and Performed Procedure database functions.
qr_util.c	The C source code file that contains the source code for utility functions used by the Query/Retrieve implementation and the Modality Worklist implementation.
mc3adv.<ext>	The Merge DICOM Toolkit library.
mergecom.app	The Application Profile.
merge.ini	The Initialization File.
mergecom.pro	The Network Profile.
work.dat	The Modality Worklist and Performed Procedure Step database file.

Filename	Description
duplicate.c	The sample C source code file that uses <code>MC_Duplicate_Message</code> to change the transfer syntax of a message, whether it's from/to encapsulated/non-encapsulated. Utilizes the registering of compression callbacks.

The Storage and Storage Commitment Service Classes

The DICOM Storage Service Class defines the context for the transfer of images from one DICOM application entity to another. The Storage service does not, however, specify that the receiver of the images take ownership for the safekeeping of the images. The DICOM Storage Commitment Service Class defines a mechanism for the sender of the images to ask the receiver to commit to storing the images.

The following is an overview of the definition of the Storage and Storage Commitment Service Classes as they relate to an application developer using the Merge DICOM Toolkit. If you require greater detail concerning the service classes than is provided here, please refer to Part 4, Annex B and Annex J of the DICOM standard.

Service Definition

The service definition can be broken down into the actions of **Service Class Users (SCU's)** and **Service Class Providers (SCP's)**. An SCU sends messages to an SCP. Note that DICOM Storage SOP Instances are defined for both image and non-image objects. In client/server terminology, the SCU's role is that of a client; the SCP's role is that of a server. DICOM does not specify the *application* in which the Storage Service Class is used. It only defines a transport mechanism for exchanging Storage Service Class objects.

Now we will look more closely at the behavior of SCUs and SCPs for the Storage and Storage Commitment Service classes.

Storage Service Class User Requirements

From the point of view of an application developer using Merge DICOM Toolkit, the behavior of a Storage Service Class SCU is very simple. An SCU will perform the following actions to transfer one or more images to an SCP.

1. Open an association with a Storage Service Class SCP.
2. Format a DICOM message containing the image to be transferred.
3. Send the image to the SCP.
4. When asynchronous operations have been negotiated, and we haven't reached the max number of operations invoked, if no data to read yet, just go ahead and send the next request message. Otherwise, wait for the response from the SCP.

5. When asynchronous operations are not negotiated, receive the response from the SCP and examine the resulting status of the send.
6. At this point the SCU may format and send another image to the SCP, or close the association.

Storage Service Class Provider Requirements

The behavior of a Storage Service Class SCP is also straight forward when using Merge DICOM Toolkit:

1. Receive associations from Storage Service Class SCUs.
2. Receive and process storage request messages sent from SCUs over these associations.
3. Send response messages as a result of processing storage request messages. Response messages will contain a status code as defined in *Table 2*.

Table 2: Storage Service Class Response Codes

Service Status	Meaning
REFUSED	Out of Resources.
ERROR	Dataset doesn't match request image type. Can't Understand.
WARNING	Coercion of Data Elements. Dataset doesn't match request image type. Elements Discarded
SUCCESS	

Storage Commitment Service Class User Requirements

From the point of view of an application developer using Merge DICOM Toolkit, the behavior of a Storage Commitment Service Class SCU is also very simple. An SCU will perform the following actions to request commitment of DICOM Storage instances from an SCP.

1. Open an association with the Storage Commitment Service Class SCP.
2. Format an N-ACTION-RQ message containing a list of the SOP Instances to be committed and send it to the SCP.
3. Receive association from the SCP.
4. At this point the SCU may format and send another N-ACTION-RQ message to the SCP, or close the association.
5. Wait for incoming association from the SCP.

6. Read and process the N_EVENT_REPORT request message sent from the SCP.
7. Send response messages as a result of processing the N_EVENT_REPORT request message.

Storage Commitment Service Class Provider Requirements

The behavior of a Storage Commitment Service Class SCP is also straightforward when using Merge DICOM Toolkit:

1. Receive associations from the SCU.
2. Receive and process N_ACTION request messages sent from SCU over the association.
3. Open an association with the SCU. (Note that in this case the SCP is acting as a client and opening an association. This is one of the few exceptions in DICOM, all of which are for sending of N-EVENT-REPORT request messages, where the SCP is the client.)
4. Format N_EVENT_REPORT request message and send it to the SCU over the association.
5. Receive the response message from the SCU.
6. Send additional N-EVENT-REPORT request messages to the SCU.
7. Close the association.

Summary

As can be seen from these simple descriptions of SCU and SCP behavior, Merge DICOM Toolkit transparently handles the majority of the DICOM implementation details. The sample application code described in this manual demonstrates how to use the Merge DICOM Toolkit to implement SCUs and SCPs within the Storage and Storage Commitment Service Classes.

Composite Services Supported

The DICOM standard specifies a number of composite services or “SOP Classes” which may be supported within the Storage Service Class by an SCU or SCP (see *Table 3*). An SCU or SCP may support all, or a subset of, these composite services and be conformant to the DICOM Storage Service Class.

Table 3: Storage Service Class Composite Services

Merge DICOM Toolkit Service Name	SOP Class Name	SOP Class UID
STANDARD_CR	Computed Radiography Image Storage	1.2.840.10008.5.1.4.1.1.1
STANDARD_CT	CT Image Storage	1.2.840.10008.5.1.4.1.1.2

Merge DICOM Toolkit Service Name	SOP Class Name	SOP Class UID
ENHANCED_CT_IMAGE	Enhanced CT Image Storage	1.2.840.10008.5.1.4.1.1.2.1
STANDARD_US_MF_RETIRED	Ultrasound Multi-frame Image Storage (Retired)	1.2.840.10008.5.1.4.1.1.3
STANDARD_US_MF	Ultrasound Multi-frame Image Storage	1.2.840.10008.5.1.4.1.1.3.1
STANDARD_MR	MR Image Storage	1.2.840.10008.5.1.4.1.1.4
ENHANCED_MR_IMAGE	Enhance MR Image Storage	1.2.840.10008.5.1.4.1.1.4.1
MR_SPECTROSCOPY	MR Spectroscopy Storage	1.2.840.10008.5.1.4.1.1.4.2
RAW_DATA	Raw Data Storage	1.2.840.10008.5.1.4.1.1.66
SPATIAL_FIDUCIALS	Spatial Fiducials Storage	1.2.840.10008.5.1.4.1.1.66.2
SPATIAL_REGISTRATION	Spatial Registration Storage	1.2.840.10008.5.1.4.1.1.66.1
DEFORMABLE_SPATIAL_REGISTRATION	Deformable Spatial Registration	1.2.840.10008.5.1.4.1.1.66.3
STANDARD_NM	Nuclear Medicine Image Storage	1.2.840.10008.5.1.4.1.1.5
STANDARD_NM_RETIRED	Nuclear Medicine Image Storage (Retired)	1.2.840.10008.5.1.4.1.1.5
STANDARD_US_RETIRED	Ultrasound Image Storage (Retired)	1.2.840.10008.5.1.4.1.1.6
STANDARD_US	Ultrasound Image Storage	1.2.840.10008.5.1.4.1.1.6.1
STANDARD_SEC_CAPTURE	Secondary Capture Image Storage	1.2.840.10008.5.1.4.1.1.7
SC_MULTIFRAME_GRAYSCALE_BYTE	Multi-frame Grayscale Byte Secondary Capture Image Storage	1.2.840.10008.5.1.4.1.1.7.2

Merge DICOM Toolkit Service Name	SOP Class Name	SOP Class UID
SC_MULTIFRAME _GRAYSCALE_WORD	Multi-frame Grayscale Word Secondary Capture Image Storage	1.2.840.10008.5.1.4.1.1.7.3
SC_MULTIFRAME _SINGLE_BIT	Multi-frame Single Bit Secondary Capture Image Storage	1.2.840.10008.5.1.4.1.1.7.1
SC_MULTIFRAME _TRUE_COLOR	Mulfi-frame True Color Secondary Capture Image Storage	1.2.840.10008.5.1.4.1.1.7.4
STANDARD_OVERLAY	Stand-alone Overlay Storage	1.2.840.10008.5.1.4.1.1.8
STANDARD_CURVE	Stand-alone Curve Storage	1.2.840.10008.5.1.4.1.1.9
STANDARD_MODALITY _LUT	Stand-alone Modality LUT Storage	1.2.840.10008.5.1.4.1.1.10
STANDARD_VOI_LUT	Stand-alone VOI LUT Storage	1.2.840.10008.5.1.4.1.1.11
STANDARD_XRAY _ANGIO	X-Ray Angiographic Image Storage	1.2.840.10008.5.1.4.1.1.12.1
STANDARD_XRAY_RF	X-Ray Radiofluoroscopic Image Storage	1.2.840.10008.5.1.4.1.1.12.2
STANDARD_XRAY _ANGIO_BIPLANE	X-Ray Angiographic Bi-Plane Image Storage	1.2.840.10008.5.1.4.1.1.12.3
ENHANCED_XA_IMAGE	Enhanced XA Image Storage	1.2.840.10008.5.1.4.1.1.12.1.1
ENHANCED_XRF_IMAGE	Enhanced XRF Image Storage	1.2.840.10008.5.1.4.1.1.12.2.1
STANDARD_PET	Positron Emission Tomography Image Storage	1.2.840.10008.5.1.4.1.1.128
STANDARD_PET_CURVE	Positron Emission Tomography Curve Storage	1.2.840.10008.5.1.4.1.1.129
STANDARD_RT_BEAMS _TREAT	RT Beams Treatment Record Storage	1.2.840.10008.5.1.4.1.1.481.4

Merge DICOM Toolkit Service Name	SOP Class Name	SOP Class UID
STANDARD_RT_BRACHY_TREAT	RT Brachy Treatment Record Storage	1.2.840.10008.5.1.4.1.1.481.6
STANDARD_RT_IMAGE	RT Image Storage	1.2.840.10008.5.1.4.1.1.481.1
STANDARD_RT_DOSE	RT Dose Storage	1.2.840.10008.5.1.4.1.1.481.2
STANDARD_RT_STRUCTURE_SET	RT Structure Set Storage	1.2.840.10008.5.1.4.1.1.481.3
STANDARD_RT_PLAN	RT Plan Storage	1.2.840.10008.5.1.4.1.1.481.5
STANDARD_RT_TREAT_SUM	RT Treatment Summary Record Storage	1.2.840.10008.5.1.4.1.1.481.7
STANDARD_RT_ION_BEAMS_TREAT	RT Ion Beams Treatment Record Storage	1.2.840.10008.5.1.4.1.1.481.9
STANDARD_RT_ION_PLAN	RT Ion Plan Storage	1.2.840.10008.5.1.4.1.1.481.8
STANDARD_DX_PRESENT	Digital X-Ray Image Storage - For Presentation	1.2.840.10008.5.1.4.1.1.1.1
STANDARD_DX_PROCESS	Digital X-Ray Image Storage - For Processing	1.2.840.10008.5.1.4.1.1.1.1.1
STANDARD_IO_PRESENT	Digital Intra-oral X-Ray Image Storage - For Presentation	1.2.840.10008.5.1.4.1.1.1.3
STANDARD_IO_PROCESS	Digital Intra-oral X-Ray Image Storage - For Processing	1.2.840.10008.5.1.4.1.1.1.3.1
STANDARD_MG_PRESENT	Digital Mammography Image Storage - For Presentation	1.2.840.10008.5.1.4.1.1.1.2
STANDARD_MG_PROCESS	Digital Mammography Image Storage - For Processing	1.2.840.10008.5.1.4.1.1.1.2.1
STANDARD_HARDCOPY_COLOR	Hardcopy Color Image Storage	1.2.840.10008.5.1.1.30
STANDARD_HARDCOPY_GRAYSCALE	Hardcopy Grayscale Image Storage	1.2.840.10008.5.1.1.29

Merge DICOM Toolkit Service Name	SOP Class Name	SOP Class UID
STANDARD_PRINT_STORAGE	Stored Print Storage	1.2.840.10008.5.1.1.27
STANDARD_VL_ENDOSCOPIC	VL Endoscopic Image Storage	1.2.840.10008.5.1.4.1.1.77.1.1
STANDARD_VL_MICROSCOPIC	VL Microscopic Image Storage	1.2.840.10008.5.1.4.1.1.77.1.2
STANDARD_VL_PHOTOGRAPHIC	VL Photographic Image Storage	1.2.840.10008.5.1.4.1.1.77.1.4
STANDARD_VL_SLIDE_MICROSCOPIC	VL Slide-Coordinates Microscopic Image Storage	1.2.840.10008.5.1.4.1.1.77.1.3
STANDARD_VIDEO_ENDOSCOPIC	Video Endoscopic Image Storage	1.2.840.10008.5.1.4.1.1.77.1.1.1
STANDARD_VIDEO_MICROSCOPIC	Video Microscopic Image Storage	1.2.840.10008.5.1.4.1.1.77.1.2.1
STANDARD_VIDEO_PHOTOGRAPHIC	Video Photographic Image Storage	1.2.840.10008.5.1.4.1.1.77.1.4.1
STANDARD_OPHTHALMIC_16_BIT	Ophthalmic 16 bit Photography Image Storage	1.2.840.10008.5.1.4.1.1.77.1.5.2
STANDARD_OPHTHALMIC_8_BIT	Ophthalmic 8 bit Photography Image Storage	1.2.840.10008.5.1.4.1.1.77.1.5.1
STEREOMETRIC_RELATIONSHIP	Stereometric Relationship Storage	1.2.840.10008.5.1.4.1.1.77.1.5.3
STANDARD_WAVEFORM_12_LEAD_ECG	12-lead ECG Waveform Storage	1.2.840.10008.5.1.4.1.1.9.1.1
STANDARD_WAVEFORM_AMBULATORY_ECG	Ambulatory ECG Waveform Storage	1.2.840.10008.5.1.4.1.1.9.1.3
STANDARD_WAVEFORM_BASIC_VOICE_AU	Basic Voice Audio Waveform Storage	1.2.840.10008.5.1.4.1.1.9.4.1
STANDARD_WAVEFORM_CARDIAC_EP	Cardiac Electrophysiology Waveform Storage	1.2.840.10008.5.1.4.1.1.9.3.1
STANDARD_WAVEFORM_GENERAL_ECG	General ECG Waveform Storage	1.2.840.10008.5.1.4.1.1.9.1.2

Merge DICOM Toolkit Service Name	SOP Class Name	SOP Class UID
STANDARD_WAVEFORM_HEMODYNAMIC	Hemodynamic Waveform Storage	1.2.840.10008.5.1.4.1.1.9.2.1
STANDARD_GRAYSCALE _SOFTCOPY_PS	Grayscale Softcopy Presentation State Storage	1.2.840.10008.5.1.4.1.1.11.1
STANDARD_BLENDING_SOFTCOPY_PS	Blending Softcopy Presentation State Storage	1.2.840.10008.5.1.4.1.1.11.4
STANDARD_COLOR_SOFTCOPY_PS	Color Softcopy Presentation State Storage	1.2.840.10008.5.1.4.1.1.11.2
STANDARD_PSEUDOCOLOR_SOFTCOPY_PS	Pseudo-Color Softcopy Presentation State Storage	1.2.840.10008.5.1.4.1.1.11.3
CHEST_CAD_SR	Chest CAD SR Storage	1.2.840.10008.5.1.4.1.1.88.65
KEY_OBJECT_SELECTION_DOC	Key Object Selection Document	1.2.840.10008.5.1.4.1.1.88.59
MAMMOGRAPHY_CAD_SR	Mammography CAD SR SOP Class	1.2.840.10008.5.1.4.1.1.88.50
XRAY_RADATION_DOSE_SR	X-Ray Radiation Dose SR	1.2.840.10008.5.1.4.1.1.88.67
PROCEDURE_LOG	Procedure Log Storage	1.2.840.10008.5.1.4.1.1.88.40
STANDARD_BASIC_TEXT_SR	Basic Text Structured Reporting SOP Class	1.2.840.10008.5.1.4.1.1.88.11
STANDARD_COMPREHENSIVE_SR	Comprehensive Structured Reporting SOP Class	1.2.840.10008.5.1.4.1.1.88.33
STANDARD_ENHANCED_SR	Enhanced Structured Reporting SOP Class	1.2.840.10008.5.1.4.1.1.88.22
STANDARD_ENCAPSULATED_PDF	Encapsulated PDF Storage	1.2.840.10008.5.1.4.1.1.104.1
REAL_WORLD_VALUE_MAPPING	Real World Value Mapping Storage	1.2.840.10008.5.1.4.1.1.67
SEGMENTATION	Segmentation Storage	1.2.840.10008.5.1.4.1.1.66.4

Commands Supported

When an SCU or SCP implementing the DICOM Storage Service Class sends or receives a message, the following Merge DICOM Toolkit defined commands will be encoded within the message:

C_STORE_RQ

An SCU will encode store request messages with the C_STORE_RQ command. An SCP will receive store request messages encoded with the C_STORE_RQ command.

C_STORE_RSP

An SCP will encode store response messages with the C_STORE_RSP command. An SCU will receive store response messages encoded with the C_STORE_RSP command.

Valid Messages

Valid DICOM messages are defined in terms of a composite service and command. The file "message.txt", which is included with your Merge DICOM Toolkit, contains the DICOM message formats. Below is an excerpt from the "message.txt" file for the STANDARD_CR composite service, C_STORE_RQ command. For instance, the example shows that attribute 0008,0020 representing STUDY_DATE, with a DA value representation, is present in this message.

```
#####
STANDARD_CR - C_STORE_RQ
#####
0008,0005    Specific Character set                CS    1C
Condition: EXPANDED_OR_REPLACEMENT_CHARACTER_SET_USED
Defined Terms:    ISO-IR 100, ISO-IR 101, ISO-IR 109, ISO-IR 110,
ISO-IR144, ISO-IR 127, ISO-IR 126, ISO-IR 138, ISO-IR 148
0008,0008    Image Type                            CS    3
Defined Terms:    (ORIGINAL, DERIVED) (PRIMARY, SECONDARY)
0008,0012    Instance Creation Date                DA    3
0008,0013    Instance Creation Time                TM    3
0008,0014    Instance Creator UID                  UI    3
0008,0016    SOP Class UID                          UI    1
0008,0018    SOP Instance UID                      UI    1
0008,0020    Study Date                            DA    2
0008,0021    Series Date                           DA    3
0008,0022    Acquisition Date                      DA    3
```

Note: The Merge DICOM Toolkit function call `MC_Validate_Message` can be used to check the contents of a message for validity. There are however minor limitations when validating messages opened for composite services, as is the case in the Storage Service Class. Please see the `MC_Validate_Message` function description in the *Merge DICOM Toolkit Reference Manual* for a discussion of these limitations.

Storage Service Class Sample Applications

The following discussions concerning the sample applications are general in nature, and deal with concepts necessary in the creation of Storage and Storage Commitment Service Class applications which uses Merge DICOM Toolkit. Please see the sample files “stor_scu.c” and “stor_scp.c” for specific coding examples.

The sample programs were designed to be simple in their functionality, thereby exposing the basic framework upon which any Storage or Storage Commitment Service Class program is built using Merge DICOM Toolkit. This framework consists of a series of Merge DICOM Toolkit function calls which constitute your interface to DICOM in general, and the Storage Service Class in particular.

Configuration

Both the SCU and the SCP sample Storage Service Class applications require configuration files which define communication parameters, levels of message logging, etc. Please see the “Configuration Parameters” appendix of the *Merge DICOM Toolkit Reference Manual* for complete descriptions of the configuration files. Some important points to remember for these sample applications are as follows:

SCU Configuration

Since the sample SCU will be opening a Storage Service Class association with the sample SCP, there is a section for the SCP in the Application Profile (“mergecom.app”).

The Application Entity Title for the sample SCP is MERGE_STORE_SCP.

Ensure that the PORT_NUMBER matches the value configured for TCPIP_LISTEN_PORT in the **SCP Configuration** section below.

You must change the HOST_NAME to be the host name of the machine on which the SCP will be running.

If utilizing the Storage Commitment Service Class, you should configure the TCPIP_LISTEN_PORT in the System Profile (“mergecom.pro”) to an unused TCP/IP port. Make sure the PORT_NUMBER in the “mergecom.app” for the MERGE_STORE_SCU matches this value. This value is used by the SCP to connect to the SCU.

SCP Configuration

You should configure the TCPIP_LISTEN_PORT in the System Profile (“mergecom.pro”) to an unused TCP/IP port. Make sure the PORT_NUMBER in “mergecom.app” for MERGE_STORE_SCP matches this value. This value is used by the SCU to connect to the SCP.

If utilizing the Storage Commitment Service Class, the sample SCP will be opening a Storage Commitment Service Class association with the sample SCU, there is a section for the SCU in the Application Profile (“mergecom.app”).

If utilizing the Storage Commitment Service Class, you must change the `HOST_NAME` to be the host name of the machine on which the SCU will be running. Also, ensure that the `PORT_NUMBER` matches the value configured for `TCPIP_LISTEN_PORT` in the **SCU Configuration** section above.

General Configuration

Don't forget to place the license number you received when you purchased the toolkit into the `[ASSOC_PARMS]` section of the System Profile ("mergecom.pro").

Set the environmental variable `MERGE_INI` to point to the "merge.ini" file.

Note: The sample Storage Service Class programs are shipped with a single set of configuration files: `merge.ini`, `mergecom.app`, `mergecom.pro`, and `mergecom.srv`. When utilizing Storage Commitment, both the SCU and SCP must listen for associations, requiring separate configuration files for both the SCU and SCP.

Sample SCU

Overview of Program Operation

The sample SCU sends a variable number of images, to a Storage Service Class SCP. A list of the images to be sent can be placed in a file, or the images must have been previously named "1.img", "2.img", etc. The sample SCU is invoked with command line arguments which determine the operation of the program.

These arguments take the form:

```
stor_scu -a local_ae -b local_port -n remote_host -p
remote_port -l service_list -f filename -c -e -v -u
username -w password -q [remote_application] [start_image]
[stop_image]
```

The arguments supported by `stor_scu` are described in the following table

Table 4: Options for the Store SCU

Parameter	Description
<code>-a local_ae</code>	This optional argument specifies the Application Entity Title of the sample application.
<code>-n remote_host</code>	This optional argument specifies the hostname of the remote SCP application. This argument overrides the hostname specified in the <code>remote_application</code> section of the <code>mergecom.app</code> file.
<code>-p remote_port</code>	This optional argument specifies the listen port of the remote SCP application. This argument overrides the listen port specified in the <code>remote_application</code> section of the <code>mergecom.app</code> file.

Parameter	Description
-l service_list	This optional argument specifies the service list to use when negotiating with the remote SCP application. This argument overrides the service list specified in the <code>remote_application</code> section of the <code>mergecom.app</code> file.
-f filename	This optional argument specifies a file containing a list of images to transfer. When this option is set, the <code>start_image</code> and <code>stop_image</code> parameters need not be specified.
-c	This optional argument specifies that the sample application may request storage commitment for the transferred files.
-b local_port	This optional argument specifies the port that the sample application will listen on during storage commitment. This argument overrides the built in default of 1115.
-e	This optional argument specifies that the sample application may send encapsulated (compressed) images. It is assumed that the “mergecom.app” file has been configured to support encapsulated transfer syntaxes in the service list.
-v	Print verbose information during application execution.
-u username	This optional argument specifies a username to negotiate as defined in DICOM Supplement 99, which added support for user identity information being included in association negotiation.
-w password	This optional argument specifies a password to negotiate as defined in DICOM Supplement 99. Note that just a username can be specified or a username and password can be specified. A password alone cannot be specified.
-q	Positive response to User Identity negotiation requested from the SCP.
Remote_application	Application Entity Title of remote SCP application. In the case of the sample programs, this would be the sample SCP “MERGE_STORE_SCP”.
Start_image	The image number which starts the sequence of images to send. For example, if you would like to send images “1.img”, “2.img”, and “3.img”, <code>start_image</code> would equal 1. This option is not required if the <code>-f</code> option is being utilized.
Stop_image	The image number which ends the sequence of images to send. For example, if you would like to send images “1.img”, “2.img”, and “3.img”, <code>stop_image</code> would equal 3. This option is not required if the <code>-f</code> option is being utilized.

The general flow of the sample Storage Service Class SCU can be charted as in *Figure 2*.

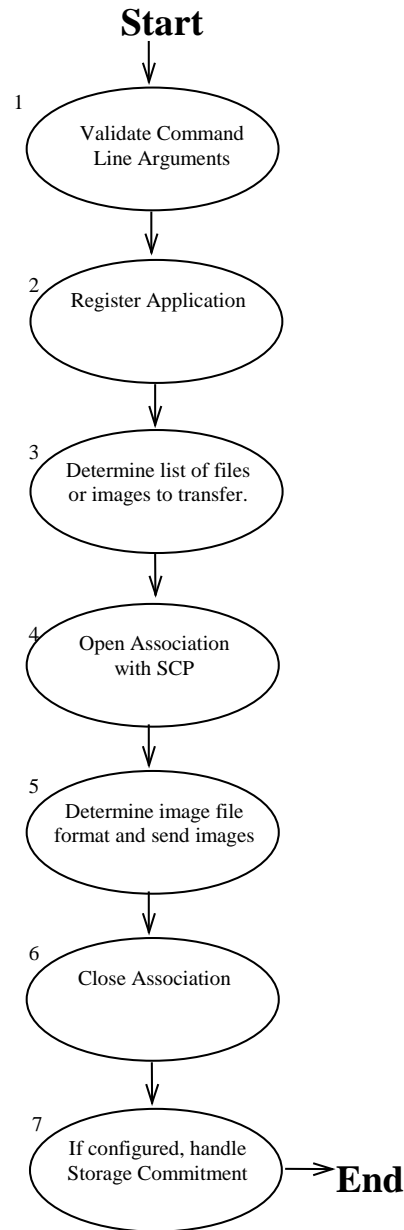


Figure 2: Sample SCU Overview

Each of the numbered steps is described below in greater detail.

1. Necessary command line arguments [`remote_application`], [`start_image`], and [`stop_image`], are verified as to their presence. Optional arguments are also read.

2. The first Merge DICOM Toolkit call is `MC_Library_Initialization` followed by `MC_Register_Application`. The former function performs general library initialization while the latter initializes data which Merge DICOM Toolkit needs for this program to function as a DICOM application entity.
3. If the `-f` option was specified, the configuration file is read in and a list is created contain all of the files to be read in. Otherwise, a list of files to be transferred is created based on the `[start_image]` and `[stop_image]` parameters.
4. `MC_Open_Association` is called to open an association with `remote_application` which was specified on the command line. There must be an entry in “mergecom.app” for `remote_application` specifying `PORT_NUMBER`, `HOST_NAME`, and `SERVICE_LIST`.
5. Each image is sent to the SCP as follows:

`CheckFileFormat` is called to determine the file format of the message.

If the file is a DICOM Part 10 format file, a file object is created with `MC_Create_Empty_File` and it is read with `MC_Open_File`. This is facilitated through the use of a callback function `MediaToFileObj` which reads blocks of data from the image file and returns this data to be added to the message. The file is then converted into a message by calling `MC_File_To_Message`.

If the file is encoded in the DICOM “stream” format, a message object is created with `MC_Open_Empty_Message`. The file is then read with `MC_Stream_To_Message` to copy the image from the disk file into the opened message. This is facilitated through the use of a callback function `StreamToMsgObj` which reads blocks of data from the image file and returns this data to be added to the message.

`MC_Get_Value_To_String` is used to get the “**SOP Class UID**” attribute for the image. This value will be used to determine the composite service.

`MC_Get_MergeCOM_Service` is called to get the composite service name by checking the “**SOP Class UID**”.

`MC_Set_Service_Command` is called to set the service name and command for the message.

`MC_Get_Value_To_String` is called to get the “**SOP instance UID**” from the message. `MC_Set_Value_From_String` is used to load this value into the “**Affected SOP Instance UID**” attribute to comply with DICOM requirements. The use of this and other “group zero elements” is defined in Part PS3.7 of the DICOM standard.

`MC_Send_Request_Message` is used to send the message containing the image to the SCP.

The application now calls `MC_Read_Message` with a timeout value of 30 seconds to wait for the response from the SCP.

When the response is received, the SCU frees the response message and returns to check for more images to send. The response status from the SCP is then checked for validity.

6. When the last image in the sequence has been sent, the application calls `MC_Close_Association` to close the association with the SCP. The loop count is then checked to determine if the sequence should be sent again.
7. When enabled with the `[-c]` option, Storage Commitment is now handled. An association is opened for Storage Commitment with the SCP. An N-ACTION Request is created and sent to the SCP with a list of all of the SOP Instances sent to the SCP. The association is then closed, and the SCU waits for an incoming association with the N-EVENT-REPORT message confirming the commitment.

Asynchronous Operations

The Storage SCU sample application has been written to take advantage of when an association allowing asynchronous DICOM operations has been negotiated. Note that by default the Storage SCU does not support asynchronous DICOM operations. The `mergecom.app` configuration file must be modified to enable this support.

The use of DICOM asynchronous operations allows an SCU to send multiple request messages to an SCP without receiving a response message. Also, the SCP can send the response messages out of order. During association negotiation, the SCU and SCP negotiate how many outstanding operations (i.e., request messages) are allowed over the association at any time.

The following aspects of the SCU's code are specific to asynchronous communications:

- The application maintains a linked list of all of the messages sent over the association. This list is used to maintain which request messages have been sent over the network and if response messages have been received for these request messages.
- The SCU polls for response messages after sending its request message. If the max operations will not be exceeded by sending another request, the SCU will simply move on to send the next request message.
- After sending all its request messages, the SCU waits for any remaining response messages until it has received responses for all of them.

Sample SCP

Overview of Program Operation

The sample SCP handles associations and receives images from Storage Service Class SCUs. The sample SCP is invoked with a command line argument which determines what is to be done with received images. These arguments take the form:

```
stor_scp [options]
```

where [options] is one of the following three values:

Table 5: Storage SCP Options

Option	Description
-a local_ae	This optional argument sets the storage SCP's local Application Entity Title.
-p listen_port	This optional argument sets the storage SCP's listen port. This value overrides the configured value in the mergecom.pro file.
-v	Execute in "verbose" mode. Additional association information will be displayed.
-F	The program will write received images to a file. The images will be named "1.img", "2.img", etc.
-M	The program will write received images to DICOM Part 10 format files. The images will be named "1.img", "2.img", etc.
-t <i1, e1, eb>	Specify the transfer syntax to store the images in where i1 = Implicit VR Little Endian, e1 = Explicit VR Little Endian, and eb = Explicit VR Big Endian. This option is used in conjunction with the -F and -M options to specify the file format.
-L	The program will list the contents received images to stdout.
-i	User Identity Negotiation information as defined in DICOM Supplement 99 will be checked for in the association request message.

The general flow of the sample Storage Service Class SCP can be charted as in *Figure 3*.

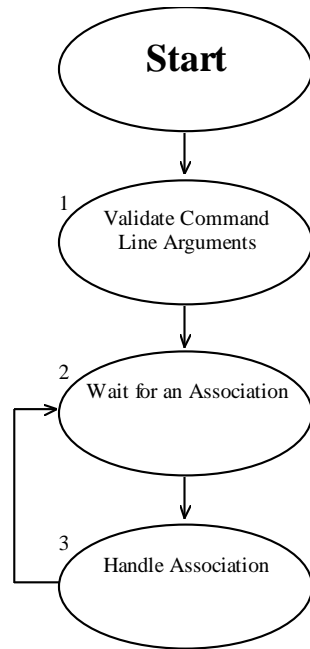


Figure 3: Sample SCP Overview

Each of the numbered steps is described below in greater detail.

1. The `options` command line arguments are checked for validity.
2. `MC_Wait_For_Association` is called as the SCP waits for an SCU to initiate an association.
3. The association is handled differently depending upon whether the operating system supports multi-tasking. In a multi-tasking environment such as UNIX, the application will create a child process to handle the association and immediately execute `MC_Wait_For_Association` to wait for the next association. In this manner the application can simultaneously process multiple associations. In a single-tasking environment such as DOS, the application must complete the processing of the association before returning to wait for another association.

Regardless of the environment, the association itself is handled as follows:

1. `MC_Accept_Association` is called to accept the association with the SCU.
2. The application next calls `MC_Read_Message` to wait for a message from the SCU.
3. At this point, the received message could be a Storage Commitment Service Class message or a Storage Service Class message. If the message is a Storage Service Class message, the following will be done:
 - a. If the "-M" option was specified, the message read will be converted into a file with `MC_Message_To_File`, and the image will be written with `MC_Write_File`. The writing of the file is facilitated through

the use of a callback function "FileObjToMedia" which is used to write blocks of data to the file.

If the "-F" option was specified, MC_Message_To_Stream is called to store the image in a file. This streaming of the message is facilitated through the use of a callback function "MsgObjToFile" which is used to write blocks of data to a file.

If the "-L" option was specified, MC_List_Message is called to list the message to stdout.

The message is freed using MC_Free_Message or MC_Free_File.

A response message is opened by calling MC_Open_Message.

MC_Send_Response_Message is called to send the response to the SCU.

MC_Free_Message is called to free the response message.

MC_Read_Message is called to wait for another message on the association until the SCU closes the association.

4. If the message received is a Storage Commitment N-ACTION message, the following will be done:

- a. A response message is opened by calling MC_Open_Message.

A list is created to store all of the storage commitment requests. A node is placed in the list for this request.

The response message is populated, and sent to the SCU by a call to MC_Send_Response_Message.

MC_Free_Message is called to free the response and the original request message.

MC_Read_Message is called to wait for another message on the association until the SCU closes the association.

Once the association has been closed, the SCP will check if any Storage Commitment requests have been received over the association. If so, for each individual Storage Commitment N-ACTION request received, the following will be done:

- An association will be opened with the SCU.
- An N-EVENT-REPORT Request message will be created with MC_Open_Message.
- The message is populated with the appropriate fields for storage commitment and sent to the SCU by a call to MC_Send_Request_Message.
- MC_Free_Message is called to free the request message.

- `MC_Read_Message` is called to wait for the response message from the SCU.
- The response message is processed and freed with `MC_Free_Message`.
- The association is closed with `MC_Close_Association`.

Please Note!

It is important in multi-tasking (not multi-threading) applications that the parent process call `MC_Release_Parent_Association` after starting a child process to handle the association, so that the parent's resources for the association are released.

Asynchronous Operations

The Storage SCP sample application does not have any specific changes implemented to support asynchronous communications. Note that by default the Storage SCP does not support asynchronous DICOM operations. The `mergecom.app` configuration file must be modified to enable this support.

The Query/Retrieve Service Class

As described in the DICOM standard, the **Query/Retrieve Service Class** is a set of related services that make up an application. These application services cooperate with each other by using specific DICOM commands to act on a specific set of data. These services allow DICOM applications to request the transfer of images between DICOM conformant applications.

Service Definition

The Query/Retrieve (Q/R) Service Class is implemented using two applications: the **Service Class Provider (SCP)** and the **Service Class User (SCU)**. The SCP accepts find requests from the SCU and performs searches using a simple search algorithm to find the images specified in the find command. The SCU then requests that the SCP move those images that were found to a specified application entity. The following tables detail the composite services supported by the SCP and the SCU.

Table 6: Composite Services Supported by the SCP

Merge DICOM Toolkit Service Name	SOP Class Name	SOP Class UID
PATIENT_STUDY_ONLY_QR_FIND	Patient/Study Only Root Query/Retrieve Information Model - Find	1.2.840.10008.5.1.4.1.2.3.1
PATIENT_STUDY_ONLY_QR_MOVE	Patient/Study Only Root Query/Retrieve Information Model - Move	1.2.840.10008.5.1.4.1.2.3.2

Merge DICOM Toolkit Service Name	SOP Class Name	SOP Class UID
STUDY_ROOT_QR_FIND	Study Root Query/Retrieve Information Model - Find	1.2.840.10008.5.1.4.1.2.2.1
STUDY_ROOT_QR_MOVE	Study Root Query/Retrieve Information Model - Move	1.2.840.10008.5.1.4.1.2.2.2
PATIENT_ROOT_QR_FIND	Patient Root Query/Retrieve Information Model - Find	1.2.840.10008.5.1.4.1.2.1.1
PATIENT_ROOT_QR_MOVE	Patient Root Query/Retrieve Information Model - Move	1.2.840.10008.5.1.4.1.2.1.2
COMPOSITE_INSTANCE_ROOT_RET_MOVE	Composite Instance Root Retrieve - Move	1.2.840.10008.5.1.4.1.2.4.2

Table 7: Composite Services Supported by the SCU

Merge DICOM Toolkit Service Name	SOP Class Name	SOP Class UID
PATIENT_STUDY_ONLY_QR_FIND	Patient/Study Only Root Query/Retrieve Information Model - Find	1.2.840.10008.5.1.4.1.2.3.1
PATIENT_STUDY_ONLY_QR_MOVE	Patient/Study Only Root Query/Retrieve Information Model - Move	1.2.840.10008.5.1.4.1.2.3.2
STUDY_ROOT_QR_FIND	Study Root Query/Retrieve Information Model - Find	1.2.840.10008.5.1.4.1.2.2.1
STUDY_ROOT_QR_MOVE	Study Root Query/Retrieve Information Model - Move	1.2.840.10008.5.1.4.1.2.2.2
PATIENT_ROOT_QR_FIND	Patient Root Query/Retrieve Information Model - Find	1.2.840.10008.5.1.4.1.2.1.1

Merge DICOM Toolkit Service Name	SOP Class Name	SOP Class UID
PATIENT_ROOT_QR_MOVE	Patient Root Query/Retrieve Information Model - Move	1.2.840.10008.5.1.4.1.2.1.2
COMPOSITE_INSTANCE_ROOT_RET_MOVE	Composite Instance Root Retrieve - Move	1.2.840.10008.5.1.4.1.2.4.2

Commands Supported

C-FIND Command

The C-FIND command is the mechanism used in DICOM to conduct a query. This implementation of the C-FIND command for the Q/R Service Class is intended as an example. Only the baseline behavior has been implemented. The full behavior has been left for the application developer.

C-MOVE Command

Application Entity Tiles should be configurable

The C-MOVE command is the mechanism used in DICOM to initiate the transfer of SOP instances between application entities. A C-MOVE-RQ instructs an SCP to move a SOP instance to a specified application entity (and not hostname and TCP/IP port). Thus, application entity titles within a DICOM network should be unique because they identify applications within the scope of a network. It is recommended that application entity titles be configurable to help solve this problem.

C-GET Command

C-GET not widely implemented!

This example does not use the DIMSE C-GET service in this implementation of the Q/R Service Class. Note that the DICOM Standard indicates that a DIMSE-C C-GET may also be used to implement the Q/R Service Class. Currently, most implementations of the Query/Retrieve Service Class use the C-MOVE command to initiate transfers of images.

Extended Negotiation for Relational Queries

Although the Query/Retrieve sample applications do not support it, Merge DICOM Toolkit supports the use of extended negotiation information for relational queries. See the functions `MC_Set_Negotiation_Info`, `MC_Get_Negotiation_Info`, and `MC_Clear_Negotiation_Info` in the *Merge DICOM Toolkit Reference Manual* for further details.

Valid Messages

Valid DICOM messages are defined in terms of a composite service and command. The file “message.txt”, which is included with your Merge DICOM Toolkit, contains DICOM message formats. Below is an excerpt from the “message.txt” file.

```
#####
#####
PATIENT_ROOT_QR_FIND - C_FIND_RQ
#####
#####
0008,0005    Specific Character Set                CS    3
Defined Terms:  ISO-IR 100, ISO-IR 101, ISO-IR 109, ISO-IR 110,
ISO-IR144, ISO-IR 127, ISO-IR 126, ISO-IR 138, ISO-IR 148
0008,0008    Image Type                            CS    3
Enumerated Values: (ORIGINAL, DERIVED)(PRIMARY,SECONDARY)(SINGLE
PLANE, BIPLANE A, BIPLANE B,BIPLANE)
0008,0012    Instance Creation Date                DA    3
0008,0013    Instance Creation Time                TM    3
0008,0014    Instance Creator UID                 UI    3
0008,0016    SOP Class UID                         UI    3
0008,0018    SOP Instance UID                     UI    1C
Condition: IMAGE_LEVEL_QUERY
0008,0020    Study Date                            DA    3
0008,0021    Series Date                           DA    3
0008,0022    Acquisition Date                     DA    3
0008,0023    Image Date                            DA    3
```

The Merge DICOM Toolkit function call `MC_Validate_Message` can be used to check the contents of a message for validity. There are some limitations when validating messages opened for composite services. Please see the `MC_Validate_Message` function description in the *Merge DICOM Toolkit Reference Manual* for a discussion of these limitations.

Query/Retrieve Service Class Sample Applications

This section describes the implementation of the Q/R Service Class. As previously mentioned, this implementation is intended to be used as an *example* of one way a user may implement the Q/R Service Class; it is not a full implementation. The sample does, however, give the user a good feel for the use of the Merge DICOM Toolkit.

The Q/R Service Class is implemented by two peer DICOM Application Entities (AE's) with one AE acting as the service class provider (SCP) and the other acting as the service class user (SCU). The SCP accepts and services the DICOM DIMSE-C commands C-FIND and C-MOVE. These commands are constructed and sent to the SCP by the SCU. The figure below outlines this interaction.

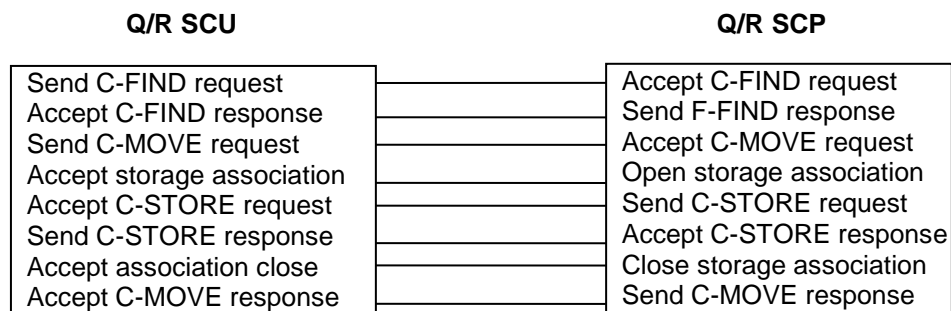


Figure 4: Q/R SCU and Q/R SCP Interaction

The basic user-provider interaction scenario of the sample Q/R application proceeds as follows. A more detailed description is given in the individual program sections.

1. The SCU requests that the SCP perform a search on the information it possesses. The SCU forms a message which contains a `C-FIND` request and information pertaining to the data it would like the SCP to match.
2. The SCP searches through the data it possesses. It then generates responses containing a unique key for each match it finds. The response messages contain a `PENDING` command status and the information which was requested by the SCU if the operation was successful. Otherwise, a blank message is formed and an error status is returned to the SCU. If no error exists, the SCP continues to send `PENDING` response messages until the final match is found. The SCP then returns `SUCCESS`.
3. The SCU receives each response message and interprets the status and data. The SCU reads `C-FIND` responses from the SCP until a status of `SUCCESS` is received.
4. The SCU then generates `C-MOVE` requests to the SCP by sending unique key values to the SCP. The SCP opens a separate association with an SCU and for each key value that the SCU sends to the SCP, the SCP generates separate `C-STORE` operations to perform the actual move from one place to another.
5. The SCP sends a response message with a status of `PENDING` to the SCU during the processing of the `C-STORE` operation.
6. When the `C-STORE` operation finishes, the SCP sends the SCU an additional `C-MOVE` response message containing the final status of the operation.
7. In addition, the SCU can generate a frame level retrieval by specifying a list of frame numbers in a multi frame image. The SCP can handle a `C-MOVE` request based on this frame level retrieval as specified by the SCU.

Configuration

Configuration information for the sample applications are kept in configuration files. These files contain initialization and startup information used by the sample applications as they execute. There are 4 different configuration files necessary for execution by any one application. They are: the Initialization file (referred to as "the dot ini file"), the Network Profile (referred to as "the dot pro file"), the Application Profile (referred to as "the dot app file") and the Service file (referred to as "the dot S-R-V file").

The configuration files follow the same format: a section starts with a label delimited with square brackets. Each item belonging to a section is then listed. The list is constructed of a variable followed by the equal sign ("=") followed by the value of the variable.

For a more detailed discussion of the configuration files distributed with the DICOM Toolkit please see the files on the distribution itself and see the DICOM Toolkit Reference Manual. Each file is fully documented and explained in detail.

Sample Client (User)

The Q/R Service Class User is an Application Entity whose purpose is to send requests for find and move services to the Q/R Service Class Provider.

The SCU is implemented in the C programming language using a program model suited for single-tasking operating systems. By using the single-tasking model, portability between single-tasking operating systems such as MS-DOS and multi-tasking operating systems such as UNIX become less of an issue.

Running the SCU

To run the SCU provided, execute the following command:

```
qr_scu remote_ae [options]
```

where `remote_ae` is the Q/R SCP and `options` are described below.

Note: The above command assumes that you are in the proper directory and that your `MERGE_INI` environment variable has been set properly.

Note: The listen port has to be changed in the `mergecom.pro` file of the Q/R SCU to match the Q/R SCP's port number found in the `mergecom.app` file.

The options supported by `qr_scu` are described in the following table:

Table 8: Options for the Q/R SCU

Option	Parameter	Description
-a	AppTitle	This option specifies the Application Title for this application. The default is "MERGE_QR_SCU." AppTitle may be any user defined string with a length of no greater than 16 characters.
-d	Dest	This option specifies the name of the remote application that will act as the destinations of moves. The Dest title may be any user defined string with a length of no greater than 16 characters. Dest defaults to this application.
-p	ListenPort	This option specifies the port that this application will listen on during retrieval from the remote application, i.e. moves to the default destination. This option overrides the built in default of 1115, and will be used when Dest is not provided.
-h	<none>	This option prints a short description on how to run qr_scp.
-o	TimeOut	This option specifies the length of time-outs on network commands. TimeOut is specified in whole integer seconds.
-t	Type	This option specifies the type of image being transferred. The default is "ct". It is used to determine the name of the image file that data will be taken from, by default "ct.img".
-1	ServList1	This option specifies the name of the service list that will be used as the Q/R service list. ServList1 may be any user defined string with a length of no greater than 16 characters. This string must be defined in the Service File.
-2	ServList2	This option specifies the name of the service list that will be used as the C-STORE service list. ServList2 may be any user defined string with a length of no greater than 16 characters. Currently this option is not implemented since the service list is obtained from the "dot app" file. The service list for the Dest parameter is used.

The SCU User Interface

The Q/R SCU has been designed to include a simple user interface. The main menu appears below.

```
[1] Begin [PATIENT_STUDY_ONLY] Query
[2] Choose Model [PATIENT_STUDY_ONLY]
[3] Show Options
[4] Instructions
[5] Exit
```

A typical scenario is:

1. Choose a Model
2. Begin the query
3. Enter data using * as a wildcard
4. Enter “d” for done editing
5. Select an instance
6. Enter a number for the instance to select
7. MOVE the instance that was selected
8. Exit

Note: The default model and level, `Patient Study Only`, is the only model and level that is handled by the sample SCP.

The Enter Data option allows data to be entered in the following manner. Type the option followed by the data (e.g. to enter a name type “2 Jones^B”).

DICOM Related Functions

main

The `main` function begins by calling `SetProgramDefaults` and `GetOptions` to configure how the program will run. It then proceeds to run a loop that will give a user a chance to proceed the way they wish. Program default values may be changed, query model may be changed, or beginning a query may occur from the main function. Exiting is an option as well, which will mean releasing the application that was registered in Merge DICOM Toolkit by calling `MC_Release_Application`. Finally, the list of query results is freed if there are any elements still in it.

CFINDOption

`CFINDOption` checks to make sure the list of query results is freed up, and then proceeds to send a C-FIND message by calling `SendCFINDMessage`. The now empty list is used to hold the data that is returned from the SCP.

SendCFINDMessage

Next, a new message is opened by calling `MC_Open_Message`. The message is filled in with the items the SCU would like the SCP to return. This includes the Q/R level. The message is sent on its way by calling `MC_Send_Request_Message`.

After the message has been sent to the SCP, the SCU waits for a response by calling `MC_Read_Message`. When a message arrives, the status is checked to determine whether more data is coming (`C_FIND_PENDING`) or whether the last data item has been returned (`C_FIND_SUCCESS`). If the status is pending, the unique ID returned to the SCU is stored in the `ReturnedData` array and `MC_Read_Message` is called again to wait for the next message. If success is returned, the final data item has been sent to the SCU and the final patient ID is stored in the patient ID array.

CMOVEOption

`CMOVEOption` establishes an association with the SCP in order to move the selected item. It establishes the connection by calling `MC_Open_Association`. If successful, `SendCMOVEMessage` is called to send the move request to the SCP.

SendCMOVEMessage

`SendCMOVEMessage` is used to handle sending of C-MOVE request messages and to accept the C-MOVE responses. This function does the following:

1. A listen socket which is done by `ProcessCSTOREAssociation`.
2. A C-MOVE request message is generated for each SOP Instance the user wants returned reported in the data by the C-FIND request. I
3. The C-MOVE message is sent to the associated application. It then checks for a response for the associated application. Notice the `MC_Read_Message` with a time-out of 0. This checks only once for a message so that `ProcessCSTOREAssociation` can be called to handle any incoming C-STORE message.
4. `ProcessCSTOREAssociation` is called to process the incoming storage association.
5. It loops doing the previous two steps until either an error response is received or a success response is received. If a success response is received, it will call `ProcessCSTOREAssociation` one more time to get any other pending responses.

Notice that `ProcessCSTOREAssociation` is called in key areas to avoid missing messages and to avoid wasting time when no messages are going to be present. In a multiprocessing environment it would be appropriate to have this function fork another process to handle the communication when an association was received. If threads are used this function would most likely be threaded in order to continue to wait for new associations.

ProcessCSTOREAssociation

`ProcessCSTOREAssociation` processes a storage association that is initiated by a C-MOVE message. It begins by posting a `MC_Wait_For_Association`. This initial “wait” gets the listen socket started, which avoids a possible problem with timing. If no association is received it just returns, if one is received it is dealt with.

While the association proceeds with the remote application (the SCP), three events can occur concurrently:

1. The association could be closed, the association could be aborted or the network could shut down. If one of these happens it will handle the association closing and return with no error.
2. There could be some sort of abnormal response which would cause a return with an error.
3. There could be a successful C-STORE message in which case we are ready to receive the information. A file is opened and given to `MC_Message_To_Stream` along with the callback function `MessageToFile`. The data is then written to the file. If there is some sort of problem with writing the data to file, `CancelCMOVERQ` is called to cancel the sending of any more information. If there were no problems, a C-STORE response is then given to the associated application for the successful receipt of the message.

To clean up, the message object is released by calling `MC_Close_Message`. Also, the second association is closed (if it was opened) by calling `MC_Close_Association`. This is continued until the function returns because of one of the above conditions.

MessageToFile

`MessageToFile` processes a C-STORE request from a remote application. After determining the modality of the service, `MessageToFile` streams the entire message into a file and sends a successful response.

BuildCFINDMessage

Once the user has input all the necessary data in `ChooseModel` and `GetLevelData`, it is passed to this function. The proper message is then built and returned so that the message can be sent by `SendCFINDMessage`.

BuildCMOVEMessage

Once the C-FIND responses have been received from the SCP the SCU can begin to generate C-MOVE requests. Once a user indicates that they want a particular Patient ID, Study Instance UID, Series Instance UID, or SOP Instance to be moved this function creates the C-MOVE request.

ReadCFINDMessage

This function decodes a C-FIND response and places the data into a data structure. This data is later used by `PrintCFINDResults` to be displayed to the user.

CancelCMOVERQ

This routine simply sends a `C-CANCEL-MOVE-RQ` to the SCP to stop the MOVE request that was started.

CancelCFINDRQ

This routine sends a `C-CANCEL-FIND-RQ` to the SCP to stop the FIND request that was started.

SetValue

Simplifies the `MC_Set_Value_From_String` call since it takes care of all error handling, default values, and required values.

GetValue

Simplifies the `MC_Get_Value_To_String` call since it takes care of all error handling and default values.

User Interface Functions

ChooseModel

`ChooseModel` sends a message to the SCP to determine what type of querying it will do. The information that is received is used to construct a menu to allow the user to choose what query model they wish to use.

EditQuery

`EditQuery` will, depending on the option you choose, allow you to edit the fields of the root query by calling `GetLevelData`. After editing of the root level information is complete, `EditQuery` will call `CFINDOption` which opens a new association to the SCP. If the association negotiation finishes properly and without any errors, `SendCFINDMessage` is called to send a DIMSE `C-FIND` message to the SCP. The user now possesses the results of the `C-FIND` command. `EditQuery` now calls `CMOVEOption` which calls `SendCMOVEMessage` to construct the `C-MOVE` message and send it to the SCP. Finally, `MC_Close_Association` is called to break the communication between the processes.

GetLevelData

`GetLevelData` is used to modify the parameters of the query. It displays a user interface that will allow the user to modify the fields of the query. The fields that the user may edit depend on the model and the level of the query. When editing is finished it returns the new information back so that it can be used in creating a `C-FIND` message.

OKtoMove

Once the SCU receives all the `C-FIND` responses for the SCP this function lists the data and allows the user to select the data to be moved.

SelectRecord

This function displays the `C-FIND` responses from a previous query and allows the user to select a specific response for continuing the query to the next level.

PrintCFINDResults

Prints the results list from a query that was received from the SCP. It will print out the fields that are relevant to the particular level that the query was done on. It may prompt to select, move, continue, or quit a query.

MainMenu

Gives a user interface similar to what is seen in the scenario mentioned earlier.

NextMenu

Gives other user interface menus to allow changing from level to level and continuing the search independently of what model is chosen.

ShowOptions

Shows the user a list of application configuration variables that may be modified at run-time.

SetOptions

Allows the user to configure the program during runtime and change communication options and time-outs.

ChangeAheadLevel

Changes the query level of the query after an instance is selected from the list that was received from the SCP. It is transparent to the user that they have changed to a different level of the query.

ChangeBackLevel

Changes the query level back to the root level of the current model. This happens when querying is complete and it is desired to start over.

AddToList

Adds an element to the list of elements that were received in a query. This is an extremely simple implementation of a list.

EmptyList

Empties the list of query results. This is an extremely simple implementation of a list.

GetOptions

Takes care of all command line arguments given to the SCU, and sets the values accordingly.

SetProgramDefaults

Takes care of the initialization of many variables that the SCU needs set before it can start running. It is called to set the defaults before the command line arguments are taken into consideration.

Sample Server (Provider)

The Q/R Service Class Provider is a program whose purpose is to provide services to the Q/R Service Class User.

The SCP, like the SCU, is implemented in the C programming language using a program model suited for single-tasking operating systems. By using the single-tasking model, portability between single-tasking operating systems such as MS-DOS and multi-tasking operating systems such as UNIX becomes less of an issue.

The SCP receives DICOM C-STORE-RQ messages, stores these incoming messages to disk, and maintains internal database with information about the C-STORE-RQ messages received. The internal database in reality is a data structure which contains Patient, Study, Series, and Instance level information. (Note that patient and study information is kept together in the same record.) The root of the data structure is a linked list of study records. Each study record contains a list of series records for the study, and each series record contains a list of instance level records. This hierarchical data structure is then used to respond to Query/Retrieve requests from an SCU.

The Query/Retrieve application can keep its state between restarts of the application. All C-STORE Request messages received are stored in a local folder, and are re-read upon startup of the application to reconstruct the internal database to its previous state. Note that due to the nature of the implementation, it can take an excessive amount of time to repopulate the database if a large number of images have been stored by the application.

Running the SCP

To run the SCP provided, execute the following command.

```
qr_scp [options]
```

where `options` are described below. Please note that the above command assumes that you are in the proper directory and that your `MERGE_INI` environment variable has been set properly (or that your `merge.ini` file is stored in the local directory from which you're running the application).

The options supported by `qr_scp` are described in the following table.

Table 9: Options for Q/R SCP

Option	Parameter	Description
-a	AppTitle	This option specifies the Application Title for this application. The default is "MERGE_QR_SCP." AppTitle may be any user defined string with a length of no greater than 16 characters.
-h	<none>	This option prints a short description on how to run <code>qr_scp</code> .

Option	Parameter	Description
-o	TimeOut	This option specifies the length of time-outs on network commands. TimeOut is specified in whole integer seconds.
-s	ServList	This option specifies the name of the service list. ServList may be any user defined string with a length of no greater than 16 characters. This string must be defined in the Service File.
-t	Type	This option specifies the type of image being transferred. The default is "CT." qr_scp searches for a file with this name and the extension ".img"
-f	Folder	This option specifies a folder to place incoming DICOM images into. The sample application receives C-STORE-RQs, stores these images, and maintains data structures with information about the stored C-STORE-RQ messages so it can respond to C-FIND and C-MOVE requests.

DICOM Related Functions

main

The `main` function begins by setting default values and parsing the command line. It does this by calling `SetProgramDefaults` and `GetOptions`. Next, the application is registered in Merge DICOM Toolkit by calling `MC_Register_Application`. This call provides Merge DICOM Toolkit with the information necessary to identify the SCP from other DICOM applications.

Next, the SCP checks the storage folder to look for any pre-existing images. The `InitDatabaseFromFolder` routine is called to read any files that were received by the application in a previous run.

The SCP now goes into a "do forever" loop. Inside the loop, the SCP performs three functions: it waits for an association to be received from the SCU, it calls `HandleAssociation` and it closes the association. The SCP waits for an association by calling `MC_Wait_For_Association`. This function is used to wait for another remote DICOM application to make a request for service. The association is "broken" or "let go" by calling `MC_Close_Association`. This function gracefully shuts down an open association and releases any resources bound to that association.

Finally, the program ends with a call to `MC_Release_Application`. This call "de-registers" the application and releases the system resources used by the application.

HandleAssociation

The `HandleAssociation` function call processes a newly opened DICOM association. It reads a message from the open association and processes it by calling the appropriate processing routine: `ProcessCFINDRQ`, `ProcessCMOVERQ`, or `ProcessCSTORERQ`. It returns `QR_SUCCESS` if the function finishes properly or `QR_FAILURE` if it detects an error.

`HandleAssociation` begins by accepting the newly opened association. This is accomplished by calling `MC_Accept_Association`. This call accepts the association posed by the remote application and signals the remote application to proceed.

Again, as in the main function, `HandleAssociation` goes into a "do forever" loop. This loop will service the calling application by reading a message and calling `ProcessCFINDRQ`, `ProcessCMOVERQ` or `ProcessCSTORERQ` based on the contents of the received message.

The message is read from the open association by calling `MC_Read_Message`. Next, `HandleAssociation` determines which processing function to call by looking at the command sent along with the message. If the command is a `C_FIND_RQ`, `HandleAssociation` calls `ProcessCFINDRQ`. If the command is `C_MOVE_RQ`, `HandleAssociation` calls `ProcessCMOVERQ`. If the command is `C_STORE_RQ`, `HandleAssociation` calls `ProcessCSTORERQ`.

ProcessCFINDRQ

The `ProcessCFINDRQ` determines the query model being used and the query level for the C-FIND-RQ it is processing. It then calls a number of routines that search the internal database of the sample application and send matching response messages back to the SCU.

`ProcessCFINDRQ` will call `SearchDBPatientLevel`, `SearchDBStudyLevel`, `SearchDBSeriesLevel`, or `SearchDBImageLevel` to search the internal database for matching information. These routines search the hierarchical data structure to find the matching records at the appropriate query level. When the routine has found a matching record, it calls `SendCFINDReply` to send the actual C-FIND-RSP message back to the SCU.

Finally, after each response message has been sent back to the SCU, a final response message with a status of `C_FIND_SUCCESS` is sent back to the SCU signaling the end of the transaction. This is done through the `SendCFINDComplete` function.

SendCFINDReply

`SendCFINDReply` is the routine that allocates, constructs and sends response messages back to the SCU.

`SendCFINDReply` begins by calling `MC_Open_Message` to obtain a new message object. An object is opened that supports the query model of the original C-FIND Request.

Next, the object is populated with data. This is done by calling `CheckReturnTag`, which checks if a tag was requested in the C-FIND Request, and sets it with the `MC_Set_Value_From_String` function if needed. This is done for all the tags supported in the database for the level of the query being performed.

Once the message has been populated, the reply message is sent to the SCU with a call to `MC_Send_Response_Message`. This process is repeated for each match found while querying the data.

ProcessCMOVERQ

The `ProcessCMOVERQ` function is responsible for processing a DIMSE C-MOVE request from the SCU. Because of implementation decisions, this routine was designed to handle two open associations at the same time. The DICOM standard states the C-STORE DIMSE service shall be implemented and executed over a separate association than that of the C-MOVE. Thus, `ProcessCMOVERQ` needs to handle two open associations at the same time.

Q/R SCP becomes Store SCU!

Please note that although the SCP is a provider, at this point in the code, the SCP turns into a C-STORE SCU.

`ProcessCMOVERQ` begins by calling `MC_Open_Association` to establish a new association on which the C-STORE will be performed. It then loops through the internal database to find matching image information to transfer. As each match is found, the location of the matching DICOM Part 10 format file is retrieved from the databases.

Once the file is known, the `MC_Create_Empty_Message` function is used to create an empty file and `MC_Open_File` is used to read the file from disk and place it into a Merge DICOM Toolkit file object. Before transferring over the network, the file must be changed into a Merge DICOM Toolkit's message format, by calling `MC_File_To_Message`. The message is then sent back to the SCU by calling `MC_Send_Request_Message`. The routine then waits for the SCU to respond by calling `MC_Read_Message` with a -1 as the time-out parameter. The -1 instructs the Merge DICOM Toolkit system to wait for a message until one arrives. When a message arrives, the routine checks to be sure it is a `C_STORE_RSP`. If it is, `ProcessCMOVERQ` makes sure the C-STORE command completed successfully. It finally frees the message object and cleans up the second association.

ProcessComplnstRootRetRQ

This function composes a response message based on frame level retrieve request specified by `COMPOSITE_INSTANCE_ROOT_RET_MOVE` service. It uses `MC_Get_Frame_To_Function` to retrieve frames from a multi frame image and constructs a new SOP instance.

ProcessCSTORERQ

The `ProcessCSTORERQ` function is responsible for processing a DIMSE C-STORE Request. The Storage SCU sample application, as described earlier in this manual, can be used to send C-STORE Request messages to the SCP. Note that the Application Profile (`mergecom.app`) may have to be updated for the

Storage SCU to communicate with the Query/Retrieve SCP. The default entry for MERGE_QR_SCP in the Application Profile does not include Storage Services in its service list. This service list may have to be updated to include storage services. Note also that a different service list can be specified on the command line of the Storage SCU to address this problem.

The `ProcessCSTORERQ` function first reads data needed for the SCP's database from the C-STORE-RQ message. This information includes tags that are needed for later responding to C-FIND Request messages. `ProcessCSTORERQ` calls the `CreateDataFolder` function for creating a folder to store the incoming image. A folder structure is created based on the Study and Series Instance UIDs contained in the C-STORE Request:

```
<Storage Folder>\<Study Instance UID>\<Series Instance UID>
```

The C-STORE Request is then stored as a DICOM Part 10 format file within the created directory. The file name is the SOP Instance UID for the file with a ".dcm" file extension. `ProcessCSTORERQ` calls `WriteToMedia` to write the DICOM Part 10 format file. If writing this file to disk is successful, `ProcessCSTORERQ` will insert information about the file into its internal data structures, and send a subsequent C-STORE Response message back to the Storage SCU application. Inserting into the internal data structures is done by calling the `InsertInstance` function.

WriteToMedia

This function is called by `ProcessCSTORERQ` to save incoming C-STORE Request messages as DICOM Part 10 format files. `WriteToMedia` calls Merge DICOM Toolkit's `MC_Message_To_File` to convert the incoming message into Merge DICOM Toolkit's file representation, it then adds the File Meta information to the file through the use of the `AddGroup2Elements` function. Finally, it writes the file to disk through the use of Merge DICOM Toolkit's `MC_Write_File` routine.

Other Functions

GetOptions

This function parses the command line options and sets the necessary program variables. It takes care of all options that are given on the command line.

SetProgramDefaults

The default values for the program are set with this function. Every needed program configuration variable is set to a default value. `GetOptions` is called after `SetProgramDefaults` in order to change from default values to those specified on the command line.

SearchDBPatientLevel

The internal database of the Query/Retrieve SCP application contains study, series, and image records. The study records contain both study and patient information. This routine performs a search through the list of study records and finds records that match any search keys. Note that for a patient level search a

filter is used to ensure that duplicate patient records are not matched. It will then call `SendCFINDReply` to send a response message for each matching record.

SearchDBStudyLevel

Performs a search of the Patient Level tags or the Study Level tags depending on what request was for. The search is done through a list of study level records stored in the application. It will then call `SendCFINDReply` to send a response message for each matching record.

SearchDBSeriesLevel

Performs a search of the Series level tags that match the search criteria. `SearchDBSeriesLevel` will search the list of Study records to find a matching study. It will then search the list of series contained within the study to find a list of records that match the search criteria in the `C-FIND-RQ`. It will then call `SendCFINDReply` to send a response message for each matching record.

SearchDBImageLevel

Performs a search of the Image level tags that match the search criteria. `SearchDBImageLevel` will search the list of Study records to find a matching study. It will then search the list of series contained within the study to find a matching series. It will then search the list of images within the found series for records that match the search criteria in the `C-FIND-RQ`. It will then call `SendCFINDReply` to send a response message for each matching record.

WildcardMatch

Performs a wildcard search as defined by DICOM. For a given tag, the search value from a `C-FIND` Request and the corresponding tag's value stored in the internal database are passed to the routine. `WildcardMatch` will then check these values to see if they match according to the wild card matching rules defined in DICOM.

DateAndTimeMatch

Performs a Date or Time range search as defined by DICOM. For a given date or time based tag, the search value from a `C-FIND` Request and the corresponding tag's value stored in the internal database are passed to the routine. `DateAndTimeMatch` will then check these values to see if they match according to the date and time range matching rules defined in DICOM.

InitDatabaseFromFolder

Performs a search of the configured storage folder looking for DICOM Part 10 format files. The `GetFileList` routine is used to traverse through the directory and return a list of the files contained within the storage folder. These files are then read into memory. `GetDataFromMessage` is used to retrieve patient, study, series, and instance level information from the file. The data returned from `GetDataFromMessage` is inserted into the application's database by calling the `InsertInstance` function.

InsertInstance

`InsertInstance` performs an insert of information about a DICOM C-STORE Request message into the internal data structures of the application. The routine is used by both the `ProcessCSTORERQ` and `InitDatabaseFromFolder` functions. The function traverses through the current internal database to see if a Study, Series, or Instance level record already exists for the C-STORE Request within the database. The routine will insert new information at the proper level into the database as required.

Sample Application Include File: qr.h

The first interesting item in `qr.h` is the `LIST_MESSAGES` entry. Un-commenting this line and rebuilding the application allows the SCU to print the contents of all incoming and outgoing messages to the screen.

The second interesting item in `qr.h` is `VALIDATE_MESSAGES`. Un-commenting this line and rebuilding the application allows the SCU to do some validation of incoming and outgoing messages. However there are limitations to the validation, as previously mentioned, see the *Merge DICOM Toolkit Reference Manual* for details.

The Query/Retrieve C-GET Service Sample Applications

This section describes the implementation of the Q/R Service Class for the C-GET service. This implementation is intended to be used as an *example* of one way a user may implement the Q/R Service Class; it is not a full implementation. The sample does, however, give the user a good feel for the use of the Merge DICOM Toolkit.

The Q/R Service Class is implemented by two peer DICOM Application Entities (AE's) with one AE acting as the service class provider (**SCP**) and the other acting as the service class user (**SCU**). The SCP accepts and services the DICOM DIMSE-C commands C-FIND and C-GET. These commands are constructed and sent to the SCP by the SCU. The following figure outlines this interaction.

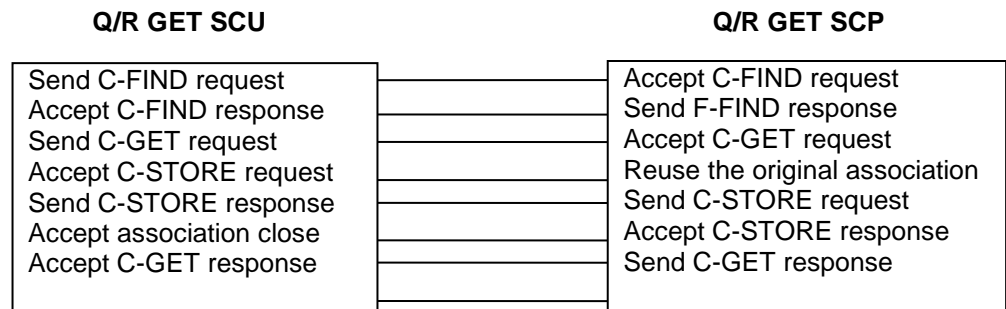


Figure 5: Q/R GET SCU and Q/R GET SCP Interaction

The basic user-provider interaction scenario of the sample Q/R GET application proceeds as follows. A more detailed description is given in the individual program sections.

1. The SCU requests that the SCP perform a search on the information it possesses. The SCU forms a message which contains a C-FIND request and information pertaining to the data it would like the SCP to match.
2. The SCP searches through the data it possesses. It then generates responses containing a unique key for each match it finds. The response messages contain a PENDING command status and the information which was requested by the SCU if the operation was successful. Otherwise, a blank message is formed and an error status is returned to the SCU. If no error exists, the SCP continues to send PENDING response messages until the final match is found. The SCP then returns SUCCESS.
3. The SCU receives each response message and interprets the status and data. The SCU reads C-FIND responses from the SCP until a status of SUCCESS is received.
4. The SCU then generates C-GET requests to the SCP by sending unique key values to the SCP. The SCP reuses the original association with an SCU and for each key value that the SCU sends to the SCP, the SCP generates separate C-STORE operations to perform the actual move from one place to another.
5. The SCP sends a response message with a status of PENDING to the SCU during the processing of the C-STORE operation
6. When the C-STORE operation finishes, the SCP sends the SCU an additional C-GET response message containing the final status of the operation.
7. In addition, the SCU can generate a frame level retrieval by specifying a list of frame numbers in a multi frame image. The SCP can handle a C-GET request based on this frame level retrieval as specified by the SCU.

Composite Services Supported by the GET SCU/SCP

Table 10: Composite Services Supported by the GET SCU/SCP

Merge DICOM Toolkit Service Name	SOP Class Name	SOP Class UID
PATIENT_STUDY_ONLY_QR_FIND	Patient/Study Only Root Query/Retrieve Information Model - Find	1.2.840.10008.5.1.4.1.2.3.1
PATIENT_STUDY_ONLY_QR_GET	Patient/Study Only Root Query/Retrieve Information Model - Get	1.2.840.10008.5.1.4.1.2.3.3

Merge DICOM Toolkit Service Name	SOP Class Name	SOP Class UID
STUDY_ROOT_QR_FIND	Study Root Query/Retrieve Information Model - Find	1.2.840.10008.5.1.4.1.2.2.1
STUDY_ROOT_QR_GET	Study Root Query/Retrieve Information Model - Get	1.2.840.10008.5.1.4.1.2.2.3
PATIENT_ROOT_QR_FIND	Patient Root Query/Retrieve Information Model - Find	1.2.840.10008.5.1.4.1.2.1.1
PATIENT_ROOT_QR_GET	Patient Root Query/Retrieve Information Model - Get	1.2.840.10008.5.1.4.1.2.1.3
COMPOSITE_INSTANCE_ROOT_RET_GET	Composite Instance Root Retrieve - Get	1.2.840.10008.5.1.4.1.2.4.3
COMPOSITE_INST_RET_NO_BULK_GET	Composite Instance Retrieve Without Bulk Data - GET	1.2.840.10008.5.1.4.1.2.5.3

Sample Client (User)

The Q/R GET Service Class User is an Application Entity whose purpose is to send requests for find and get services to the Q/R GET Service Class Provider.

Running the SCU

To run the QR GET SCU provided, execute the following command:

```
qr_get_scu.exe remote_ae [options]
```

where `remote_ae` is the Q/R GET SCP.

Note: The above command assumes that you are in the proper directory and that your `MERGE_INI` environment variable has been set properly. Also note that the listen port has to be changed in the `mergecom.pro` file of the Q/R SCU to match the Q/R SCP's port number found in the `mergecom.app` file.

The SCU User Interface

The Q/R SCU has been designed to include a simple user interface. The main menu appears below.

```
[1] Begin [PATIENT_STUDY_ONLY] Query
[2] Choose Model [PATIENT_STUDY_ONLY]
[3] Show Options
[4] Instructions
[5] Exit
```

A typical scenario would be:

1. Choose a Model
2. Begin the query
3. Enter data using * as a wildcard
4. Enter “d” for done editing
5. Select an instance
6. Enter a number for the instance to select
7. GET the instance that was selected
8. Exit

The Enter Data option allows data to be entered in the following manner. Type the option followed by the data (e.g. to enter a name type “2 Jones^B”).

The implementation details for the Q/R GET SCU sample are covered in the inline documentation of this sample.

Sample Server (Provider)

The Q/R GET Service Class Provider is a program whose purpose is to provide services to the Q/R GET Service Class User.

The SCP receives DICOM C-STORE-RQ messages, stores these incoming messages to disk, and maintains internal database with information about the C-STORE-RQ messages received. The internal database in reality is a data structure which contains Patient, Study, Series, and Instance level information. (Note that patient and study information is kept together in the same record.) The root of the data structure is a linked list of study records. Each study record contains a list of series records for the study, and each series record contains a list of instance level records. This hierarchical data structure is then used to respond to Query/Retrieve requests from an SCU.

The Query/Retrieve application can keep its state between restarts of the application. All C-STORE Request messages received are stored in a local folder, and are re-read upon startup of the application to reconstruct the internal database to its previous state. Note that due to the nature of the implementation, it can take an excessive amount of time to repopulate the database if a large number of images have been stored by the application.

Running the SCP

To run the QR GET SCP provided, execute the following command:

```
qr_get_scp [options]
```

Note: The above command assumes that you are in the proper directory and that your MERGE_INI environment variable has been set properly (or that your merge.ini file is stored in the local directory from which you're running the application).

The implementation details for the Q/R GET SCP sample are covered in the inline documentation of this sample.

The Basic Print Service Classes

The DICOM Basic Print Service Classes define the context for the printing of images using the DICOM standard. The following is an overview of the definition of the Basic Print Service Classes as they relate to an application developer using the Merge DICOM Toolkit. If you require greater detail concerning the service classes than is provided here, please refer to Part 4, Annex H of the DICOM standard.

The reader should take special notice of the use of Normalized classes in DICOM Basic Print, not the Composite classes used in the DICOM Storage Service Classes. Normalized Service Class messages work on objects and can create, delete, update or take action upon these objects.

Service Definition

The service definition can be broken down into the actions of **Service Class Users (SCU's)** and **Service Class Providers (SCP's)**. An SCU sends information and requests to an SCP. In client/server terminology, the SCU's role is that of a client; the SCP's role is that of a server. Now we will look more closely at the behavior of SCUs and SCPs.

Print Service Class User Requirements

From the point of view of an application developer using Merge DICOM Toolkit, the behavior of a DICOM Basic Print SCU is very simple. An SCU will perform the following actions to print one or more images to an SCP:

1. Open an association with a DICOM Basic Print SCP.
2. Format and transfer a DICOM message defining a Basic Film Session object.
3. Format and transfer a DICOM message defining a Basic Film Box object.
4. Format and transfer DICOM messages defining the contents of the Image Boxes to the SCP.
5. Instruct the SCP to print the images sent.
6. If the SCU supports the PRINT_JOB service class, it may wait to be notified by the SCP of the print job's status
7. At this point the SCU may format and send more messages to the SCP, or close the association.

Print Service Class Provider Requirements

The behavior of a DICOM Basic Print SCP is also straight-forward when using Merge DICOM Toolkit:

1. Receive associations from DICOM Basic Print SCUs.
2. Receive and process DICOM Basic Print messages sent from SCUs over these associations.
3. Send response messages as a result of processing DICOM Basic Print messages. Response messages will contain a status code as defined in *Table 11*, *Table 12*, and *Table 13*.

Table 11: Basic Film Session Service Class Status Response Codes

Service Status	Meaning
REFUSED	Out of Resources.
FAILURE	Invalid Attribute Value (memory allocation cannot be provided) Resource Limitation (requested memory allocation temporarily not available) Film Session SOP Instance hierarchy does not contain Film Box SOP Instances Unable to create Print Job SOP Instance; print queue is full Image position collision : multiple images assigned to single image position Image size is larger than image box size (by using the specified magnification value)
WARNING	Memory Allocation not Supported Film Session Printing (collation) is not supported Film Session SOP instance hierarchy does not contain Image Box SOP Instances (empty page)
SUCCESS	

Table 12: Basic Film Box Service Class Response Codes

Service Status	Meaning
REFUSED	Out of Resources.
FAILURE	Unable to create Print Job SOP Instance; print queue is full Image position collision : multiple images assigned to single image position Image size is larger than image box size (by using the specified

Service Status	Meaning
	magnification value)
WARNING	Film Box SOP Instance hierarchy does not contain Image Box SOP Instances (empty page)
SUCCESS	

Table 13: Image Box Service Class Response Codes

Service Status	Meaning
REFUSED	Out of Resources.
FAILURE	Insufficient Memory in printer to store the image
SUCCESS	

As can be seen from these simple descriptions of SCU and SCP behavior, Merge DICOM Toolkit transparently handles the majority of the DICOM implementation details. The sample application code described in this manual demonstrates how to use the Merge DICOM Toolkit to implement SCUs and SCPs within the Basic Print Service Classes.

Normalized Services Supported

The DICOM standard specifies a number of normalized services or “SOP Classes” which may be supported within the Basic Print Service Classes by an SCU or SCP (see *Table 14*). The DICOM standard also supports a number of “Meta SOP Classes” that can be used to refer to a group of normalized services for negotiation. An SCU or SCP may support all, or a subset of, these normalized services and be conformant to the DICOM Basic Print Service Classes.

Table 14: Basic Print Service Classes normalized services

Merge DICOM Toolkit Service Name	SOP Class Name	SOP Class UID
BASIC_GRAYSCALE_PRINT_MANAGEMENT	Basic Grayscale Print Management	1.2.840.10008.5.1.1.9
BASIC_COLOR_PRINT_MANAGEMENT	Basic Color Print Management	1.2.840.10008.5.1.1.18
REF_GRAYSCALE_PRINT_MANAGEMENT	Referenced Grayscale Print Management	1.2.840.10008.5.1.1.9.1
REF_COLOR_PRINT_MANAGEMENT	Referenced Color Print Management	1.2.840.10008.5.1.1.18.1

Merge DICOM Toolkit Service Name	SOP Class Name	SOP Class UID
BASIC_FILM_SESSION	Basic Film Session	1.2.840.10008.5.1.1.1
BASIC_FILM_BOX	Basic Film Box	1.2.840.10008.5.1.1.2
BASIC_GRAYSCALE_IMAGE_BOX	Basic Grayscale Image Box	1.2.840.10008.5.1.1.4
BASIC_COLOR_IMAGE_BOX	Basic Color Image Box	1.2.840.10008.5.1.1.4.1
REFERENCED_IMAGE_BOX	Referenced Image Box	1.2.840.10008.5.1.1.4.2
PRINTER	Printer	1.2.840.10008.5.1.1.16
PRINT_JOB	Print Job	1.2.840.10008.5.1.1.14
VOI_LUT_BOX	VOI LUT Box	1.2.840.10008.5.1.1.22
BASIC_ANNOTATION_BOX	Basic Annotation Box	1.2.840.10008.5.1.1.15
IMAGE_OVERLAY_BOX	Image Overlay Box	1.2.840.10008.5.1.1.24

Commands Supported

When an SCU or SCP implementing the DICOM Basic Print Service Classes sends or receives a message, the following Merge DICOM Toolkit defined commands will be used to encode the message:

N_CREATE_RQ

An SCU will use the N_CREATE_RQ command to create an instance of a SOP Class object. An SCP will create the object requested in a received N_CREATE_RQ message.

N_CREATE_RSP

An SCP will encode create response messages with the N_CREATE_RSP command. An SCU will receive create response messages encoded with the N_CREATE_RSP command.

N_SET_RQ

An SCU will use the N_SET_RQ command to update an instance of a SOP Class object. An SCP will update the object requested in a received N_SET_RQ message.

N_SET_RSP

An SCP will encode update response messages with the N_SET_RSP command. An SCU will receive update response messages encoded with the N_SET_RSP command.

N_GET_RQ

An SCU will use the N_GET_RQ command to get information from an instance of a SOP Class object. An SCP will get information from the object requested in a received N_GET_RQ message.

N_GET_RSP

An SCP will encode get response messages with the N_GET_RSP command. An SCU will receive get response messages encoded with the N_GET_RSP command.

N_DELETE_RQ

An SCU will use the N_DELETE_RQ command to delete an instance of a SOP Class object. An SCP will delete the object requested in a received N_DELETE_RQ message.

N_DELETE_RSP

An SCP will encode delete response messages with the N_DELETE_RSP command. An SCU will receive delete response messages encoded with the N_DELETE_RSP command.

N_ACTION_RQ

An SCU will use the N_ACTION_RQ command to create an instance of a SOP Class object. An SCP will create the object requested in a received N_ACTION_RQ message.

N_ACTION_RSP

An SCP will encode create response messages with the N_ACTION_RSP command. An SCU will receive create response messages encoded with the N_ACTION_RSP command.

N_EVENT_REPORT_RQ

An SCP will use the N_EVENT_REPORT_RQ command to report events related to an instance of a SOP Class object. An SCU will receive the event report on the object indicated in a received N_EVENT_REPORT_RQ message.

Note: The SCP is the originator of this message, not the SCU as in the other messages.

N_EVENT_REPORT_RSP

An SCU will encode event report response messages with the N_EVENT_REPORT_RSP command. An SCP will receive event report response messages encoded with the N_EVENT_REPORT_RSP command.

Basic Grayscale Print Management Meta SOP Class

The Basic Grayscale Print Management Meta SOP Class is defined by the following set of supported SOP classes: Basic Film Session SOP Class, Basic Film Box SOP Class, Basic Grayscale Image Box SOP Class and Printer SOP Class. *Table 15* offers brief descriptions of each attribute for Basic Film Box SOP Class and *Table 16* offers brief descriptions of each attribute for Image Box SOP class. For additional information please refer to DICOM standard PS 3.3 C13.

Table 15: Basic Film Box attributes

Attribute Name	Tag	Description
Image Display Format	(2010,0010)	Enumerated values for the type of image display format.
Referenced Film Session Sequence	(2010,0500)	References a film session class instance previously created.
Referenced Image Box Sequence	(2010,0510)	References a film box class instance created as a response.
Referenced Basic Annotation Box Sequence	(2010,0520)	References an annotation class instance created as a response.
Film Orientation	(2010,0040)	Enumerated values: PORTRAIT, LANDSCAPE.
Film Size ID	(2010,0050)	Defined terms for size of film.
Magnification Type	(2010,0060)	Magnification algorithm to apply to the image in order to fit the image on the film.
Max Density	(2010,0130)	Maximum density of the image, expressed in hundredths.
Configuration Information	(2010,0150)	Character string that contains printer specific print parameters. The use of this attribute is defined in the printer's conformance statement.
Annotation Display Format ID	(2010,0030)	The use of this attribute is defined in the printer's conformance statement.
Smoothing Type	(2010,0080)	Specifies type of CUBIC magnification. The values of this attribute are defined in the conformance statement.
Border Density	(2010,0100)	Specifies density of areas around and between images.
Empty Image Density	(2010,0110)	Specifies the density of the empty image box area.

Attribute Name	Tag	Description
Min Density	(2010,0120)	Minimum density of the image, expressed in hundredths.
Trim	(2010,0140)	Specifies whether a trim box be printed around each image on a film.

Table 16: Basic Grayscale Image Box N-SET attribute list

Attribute Name	Tag	Description
Image Position	(2020,0010)	The position of the image on the film.
Preformatted Grayscale Image Sequence	(2020,0110)	Sequence of Preformatted Grayscale Image Pixel Attributes.
>Samples Per Pixel	(0028,0002)	Enumerated value: 1.
>Photometric Interpretation	(0028,0004)	Enumerated values: MONOCHROME1, MONOCHROME2
>Rows	(0028,0010)	Specifies the number of rows contained within the image.
>Columns	(0028,0011)	Specifies the number of columns contained within the image.
>Pixel Aspect Ratio	(0028,0034)	Any pair of positive values (i.e. 1/1).
>Bits Allocated	(0028,0100)	Enumerated values: 8 or 16.
>Bits Stored	(0028,0101)	Enumerated values: 8 or 12.
>High Bit	(0028,0102)	Enumerated values: 7 or 11.
>Pixel Representation	(0028,0103)	Enumerated value: 0000
>Pixel Data	(7FE0,0010)	Pixel data.
Polarity	(2020,0020)	Enumerated values: NORMAL, REVERSE If this attribute is undefined, the SCP should default to NORMAL
Referenced Overlay Sequence	(0008,1130)	See DICOM standard PS3.3 C.13-6
Magnification Type	(2010,0060)	Magnification interpolation to apply to the image. Defining this attribute, overrides the magnification type specified for the image box.

Attribute Name	Tag	Description
Smoothing Type	(2010,0080)	Specifies type of CUBIC magnification. The values of this attribute are defined in the conformance statement.
Requested Image Size	(2020,0030)	Length of image rows in mm.

Note: The symbol '>' precedes the attribute name of the members of a sequence of Items.

Note: Although Window Center and Window Width are optional in storage objects, they are not included in the Preformatted Grayscale Image Sequence.

Note: Although Pixel Aspect Ratio is not mandatory in storage objects, it is mandatory in print objects.

Valid Messages

Valid DICOM messages are defined in terms of a normalized services and commands. The file "message.txt", which is included with your Merge DICOM Toolkit, contains the DICOM message formats. Below is an excerpt from the "message.txt" file for the BASIC_FILM_SESSION normalized service, N_CREATE_RQ command. For instance, the example shows that attribute (2000,0020) representing PRIORITY, with a CS value representation, is present in this message and has three enumerated values: HIGH, MEDIUM, LOW.

```

=====
=
Service Name: BASIC_FILM_SESSION
=====
=

#####
#
BASIC_FILM_SESSION - N_CREATE_RQ
#####
#

2000,0010   Number of Copies      IS    3
2000,0020   Print Priority        CS    3
Enumerated Values:                HIGH, MED, LOW
2000,0030   Medium Type          CS    3
Defined Terms:                    PAPER, CLEAR FILM, BLUE FILM
2000,0040   Film Destination     CS    3
Enumerated Values:                MAGAZINE, PROCESSOR

```

2000,0050	Film Session Label	LO	3
2000,0060	Memory Allocation	IS	3

Note: The Merge DICOM Toolkit function call `MC_Validate_Message` can be used to check the contents of a message for validity. There are, however, minor limitations when validating messages opened for normalized services, as is the case in the Basic Print Service Classes. Please see the `MC_Validate_Message` function description in the *Merge DICOM Toolkit Reference Manual* for a discussion of these limitations.

Performance Tuning

It should be noted that the use of `MC_Validate_Message` adversely affects the application's performance. To improve application performance, the use of the `MC_Validate_Message` function call should be limited in or omitted from the application.

Print Service Class Sample Applications

The following discussions concerning the sample applications are general in nature, and deal with concepts necessary in the creation of DICOM Basic Print applications which use Merge DICOM Toolkit. Please see the sample files “`prnt_scu.c`” and “`prnt_scp.c`” for specific coding examples.

The sample programs were designed to be simple in their functionality, thereby exposing the basic framework upon which any DICOM Basic Print program is built using Merge DICOM Toolkit. This framework consists of a series of Merge DICOM Toolkit function calls which constitute your interface to DICOM in general, and the Basic Print Service Classes in particular.

Configuration

Both the SCU and the SCP sample Basic Print Service Classes applications require configuration files which define communication parameters, levels of message logging, etc. Please see the “Configuration” section of the *Merge DICOM Toolkit User's Manual* for complete descriptions of the configuration files. Some important points to remember for these sample applications are as follows:

SCU Configuration

- Since the sample SCU will be opening an association with the sample SCP, there is a section for the SCP in the Application Profile (“`mergecom.app`”).
- The Application Entity Title for the sample SCP is `MERGE_PRINT_SCP`.
- Ensure that the `PORT_NUMBER` matches the value configured for `TCPIP_LISTEN_PORT` in the **SCP Configuration** section below.
- You must change the `HOST_NAME` to be the host name of the machine on which the SCP will be running.

SCP Configuration

- You should configure the TCPIP_LISTEN_PORT in the System Profile (“mergecom.pro”) to an unused TCP/IP port. Make sure the PORT_NUMBER in “mergecom.app” for MERGE_PRINT_SCP matches this value.
- “MAX_PENDING_CONNECTIONS”, also in the System Profile, can be set to a number greater than 1 if your operating system supports multi-tasking or multi-threading; in which case the SCP can process multiple concurrent associations.

General Configuration

- Don’t forget to place the license number you received when you purchased the toolkit into the [ASSOC_PARMS] section of the System Profile (“mergecom.pro”).
- Set the environmental variable MERGE_INI to point to the “merge.ini” file.

Note: The sample Print Service Class programs are shipped with a single set of configuration files: merge.ini, mergecom.app, mergecom.pro, and mergecom.srv. After making the above changes, the configuration files will be correct for use by either the SCU or SCP Print Service Class sample programs.

Sample SCU

Overview of Program Operation

The sample SCU sends a variable number of images, a variable number of times to a DICOM Basic Print SCP. These images must have been previously named “1.img”, “2.img”, etc. The sample SCU is invoked with command line arguments which determine the operation of the program. These arguments take the form:

```
prnt_scu -c [copies] -p [priority] -l [level]
[remote_application] [format] [start_image] [stop_image]
[loop_count]
```

Table 17: Print SCU Options

Option	Parameter	Description
-c	copies	Number of copies to print of the Film Boxes created
-p	priority	Priority for the Print Job(s). L(ow) M(edium) and H(igh)
-l	level	The object to send the N-Action command to: S(ession) or B(ox).
Remote_app lication	<none>	This option specifies the TCP listening port that the remote application is waiting for associations on.

Option	Parameter	Description
Format	<none>	The image layout in the image box. For example, if format would equal 2, two images would be printed next to each other in the image box
start_image	<none>	The image number which starts the sequence of images to send. For example, if you would like to send images "1.img", "2.img", and "3.img", start_image would equal 1.
stop_image	<none>	The image number which ends the sequence of images to send. For example, if you would like to send images "1.img", "2.img", and "3.img", stop_image would equal 3.
loop_count	<none>	This optional argument specifies the number of times you would like to send the image sequence. For example, if you would like to send images "1.img", "2.img", and "3.img" five times, loop_count would equal 5.

Example:

```
prnt_scu -c 1 -p L -l B MERGE_PRINT_SCP 1 1 6 2
```

The general flow of the sample DICOM Basic Print SCU can be charted as in *Figure 6*.

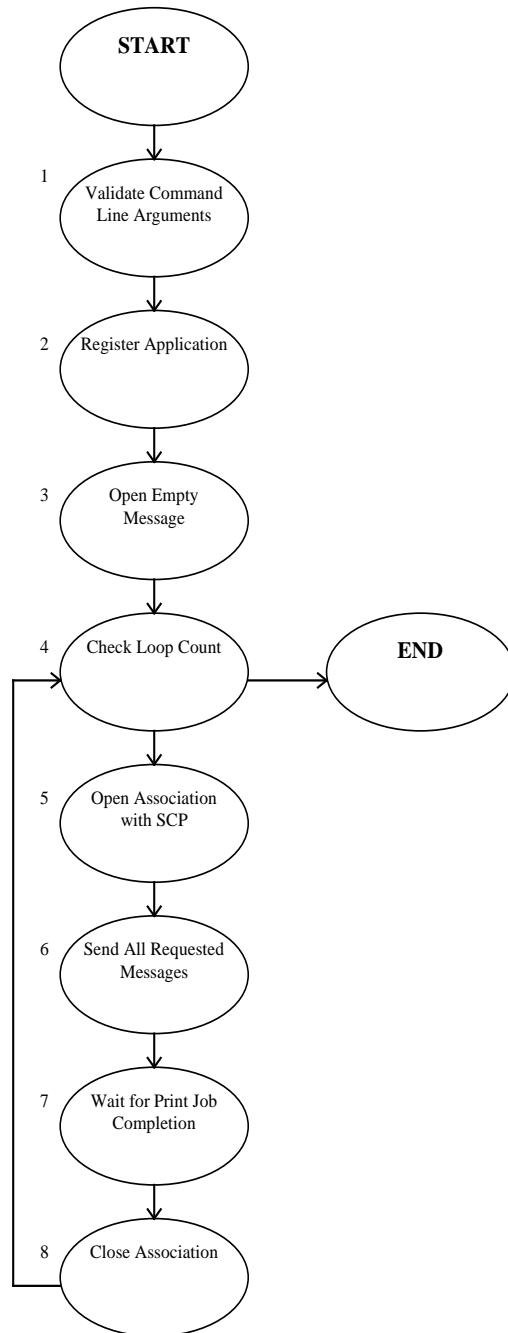


Figure 6: Sample Print SCU Overview

Each of the numbered steps is described below in greater detail:

1. Necessary command line arguments [`remote_application`], [`format`], [`start_image`], and [`stop_image`], are verified as to their presence. If [`loop_count`] has been specified on the command line, it's value is checked for validity.

Performance Tuning

2. The first Merge DICOM Toolkit call is `MC_Library_Initialization` followed by `MC_Register_Application`. The former function performs general library initialization while the latter initializes data which Merge DICOM Toolkit needs for this program to function as a DICOM application entity.
3. `MC_Open_Empty_Message` is called to open an empty message which has no service name or command associated with it so that the message can be reused for images of differing normalized services. Reusing messages in this manner can quite dramatically improve performance.
4. If `loop_count` was specified on the command line, it is checked to determine if the sequence of images should be sent.
5. `MC_Open_Association` is called to open an association with **remote_application** which was specified on the command line. There must be an entry in “mergecom.app” for **remote_application** specifying `PORT_NUMBER`, `HOST_NAME`, and `SERVICE_LIST`.
6. Each image is sent to the SCP as follows:
 - `MC_Empty_Message` is called to clear the empty message created in Step 3.
 - `MC_Stream_To_Message` is called to copy the image from the disk file into the opened message. This is facilitated through the use of a callback function `StreamToMessageFunction` which reads blocks of data from the image file and returns this data to be added to the message.
 - `MC_Get_Value_To_String` is used to get the “**SOP class UID**” attribute for the image. This value will be used to determine the normalized service.
 - `Get_Service_Name` is called to get the normalized service name by checking the “**SOP class UID**”.
 - `MC_Set_Service_Command` is called to set the service name and command for the message.
 - `MC_Get_Value_To_String` is called to get the “**SOP instance UID**” from the message. `MC_Set_Value_From_String` is used to load this value into the “**affected SOP instance UID**” attribute to comply with DICOM requirements.
 - `MC_Send_Request_Message` is used to send the message containing the image to the SCP.
 - The application now calls `MC_Read_Message` with a timeout value of 30 seconds to wait for the response from the SCP. At this time, the SCU may receive a response message or an `N_EVENT_REPORT_RQ` message from the SCP.

- When the response is received, the SCU frees the response message and returns to check for more images to send. This sample application does not check the value of the response from the SCP. Normally the response should be checked to determine if the SCP successfully printed the image. When an `N_EVENT_REPORT_RQ` message is received, the SCU will check the “**affected SOP instance UID**” and the print job status.
7. If there are outstanding print job instances, the application will wait for `N_EVENT_REPORT_RQ` messages confirming the completion of all print jobs before closing the application.
 8. When the last image in the sequence has been sent, the application calls `MC_Close_Association` to close the association with the SCP. The loop count is then checked to determine if the sequence should be sent again.

Sample SCP

Overview of Program Operation

The sample SCP handles associations and receives images from DICOM Basic Print SCUs. The sample SCP is invoked as follows:

```
print_scp
```

The general flow of the sample DICOM Basic Print SCP can be charted as in *Figure 7*.

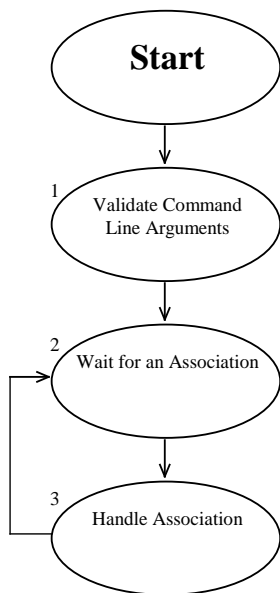


Figure 7: Sample SCP Overview

Each of the numbered steps is described below in greater detail:

1. `MC_Wait_For_Association` is called as the SCP waits for an SCU to initiate an association.
2. The association is handled differently depending upon whether the operating system supports multi-tasking. In a multi-tasking environment such as UNIX, the application will create a child process to handle the association and immediately execute `MC_Wait_For_Association` to wait for the next association. In this manner the application can simultaneously process multiple associations. In a single-tasking environment such as DOS, the application must complete the processing of the association before returning to wait for another association.
3. Regardless of the environment, the association itself is handled as follows:
 - a. `MC_Accept_Association` is called to accept the association with the SCU.

The application next calls `MC_Read_Message` to wait for a message from the SCU.

The SCP will process each message accordingly.

The message is freed using `MC_Free_Message`.

A response message is opened by calling `MC_Open_Message`.

For `N_ACTION_RQ` messages, the application will call `MC_Send_Response_Message` to send the response to the SCU.

`MC_Free_Message` is called to free the response message.

`MC_Read_Message` is called to wait for another message on the association until the SCU closes the association.

`MC_Send_Request_Message` is called to notify the SCU that a print job has completed.

`MC_Read_Message` is called to wait for the `N_EVENT_REPORT_RSP` message from the SCU.

Note: It is important in multi-tasking applications that the parent process call `MC_Release_Parent_Association` after starting a child process to handle the association, so that the parent's resources for the association are released.

The Modality Worklist Service Class

As described in the DICOM standard, the **Modality Worklist Service Class** is a set of related services that make up an application. These application services cooperate with each other by using specific DICOM commands to act on a specific set of data. These services allow DICOM applications to request the transfer of data between DICOM conformant applications.

Service Definition

The Modality Worklist Service Class is implemented using two applications: the **Service Class Provider (SCP)** and the **Service Class User (SCU)**. The SCP accepts find requests from the SCU and performs searches using a simple search algorithm to find the data specified in the find command. The SCP then forms a response message that is then sent back to the SCU. The SCU then receives the data that was found by the SCP.

Composite Services Supported by the SCU and SCP

Table 18: Composite Service Supported by the SCU and SCP

Merge DICOM Toolkit Service Name	SOP Class Name	SOP Class UID
MODALITY_WORKLIST_FIND	Modality Worklist Information Model - Find	1.2.840.10008.5.1.4.31
PERFORMED_PROCEDURE_STEP	Modality Performed Procedure Step	1.2.840.10008.3.1.2.3.3

Commands Supported

The SCP and SCU are implemented using the C-FIND DIMSE-C Service.

Also, this implementation of the Modality Worklist Service Class is intended as an example. Only the baseline behavior has been implemented. The full behavior has been left for the application developer.

Valid Messages

Valid DICOM messages are defined in terms of a composite service and command. The file “message.txt”, which is included with your Merge DICOM Toolkit, contains DICOM message formats. Below are excerpts from the “message.txt” file.

```
#####
MODALITY_WORKLIST_FIND - C_FIND_RQ
#####
0008,0005 Specific Character Set CS 3
Defined Terms: ISO_IR 100, ISO_IR 101, ISO_IR 109, ISO_IR 110,
ISO_IR144, ISO_IR 127, ISO_IR 126, ISO_IR 138, ISO_IR 148
0008,0050 Accession Number SH 3
0008,0080 Institution Name LO 3
0008,0081 Institution Address ST 3
0008,0082 Institution Code Sequence SQ 3
Item Name(s): INSTITUTION_CODE
0008,0090 Referring Physician's Name PN 3
0008,0092 Referring Physician's Address ST 3
0008,0094 Referring Physician's Telephone Numbers SH 3
0008,1080 Admitting Diagnoses Description LO 3
0008,1084 Admitting Diagnosis Code Sequence SQ 3
Item Name(s): ADMITTING_DIAGNOSIS_CODE
0008,1110 Referenced Study Sequence SQ 3
Item Name(s): REF_STUDY
0008,1120 Referenced Patient Sequence SQ 3
Item Name(s): REF_PATIENT
0008,1125 Referenced Visit Sequence SQ 3
Item Name(s): REF_VISIT
0010,0010 Patient's Name PN 3
0010,0020 Patient ID LO 3
0010,0021 Issuer of Patient ID LO 3
0010,0030 Patient's Birth Date DA 3
0010,0032 Patient's Birth Time TM 3
0010,0040 Patient's Sex CS 3
Enumerated Values: M, F, O
```

```
#####
MODALITY_WORKLIST_FIND - C_FIND_RSP
#####
0008,0005 Specific Character Set CS 3
Defined Terms: ISO_IR 100, ISO_IR 101, ISO_IR 109, ISO_IR 110,
ISO_IR144,ISO_IR 127, ISO_IR 126, ISO_IR 138, ISO_IR 148
0008,0050 Accession Number SH 3
0008,0080 Institution Name LO 3
0008,0081 Institution Address ST 3
0008,0082 Institution Code Sequence SQ 3
Item Name(s): INSTITUTION_CODE
0008,0090 Referring Physician's Name PN 3
0008,0092 Referring Physician's Address ST 3
0008,0094 Referring Physician's Telephone Numbers SH 3
0008,1080 Admitting Diagnoses Description LO 3
0008,1084 Admitting Diagnosis Code Sequence SQ 3
Item Name(s): ADMITTING_DIAGNOSIS_CODE
0008,1110 Referenced Study Sequence SQ 3
Item Name(s): REF_STUDY
0008,1120 Referenced Patient Sequence SQ 3
Item Name(s): REF_PATIENT
0008,1125 Referenced Visit Sequence SQ 3
Item Name(s): REF_VISIT
0010,0010 Patient's Name PN 3
0010,0020 Patient ID LO 3
0010,0021 Issuer of Patient ID LO 3
```

0010,0030	Patient's Birth Date	DA	3
0010,0032	Patient's Birth Time	TM	3
0010,0040	Patient's Sex	CS	3

Enumerated Values: M, F, O

The Merge DICOM Toolkit function call `MC_Validate_Message` can be used to check the contents of a message for validity. There are minor limitations when validating messages opened for composite services. Please see the `MC_Validate_Message` function description in the *Merge DICOM Toolkit Reference Manual* for a discussion of these limitations.

Modality Worklist and Modality Performed Procedure Step Sample Applications

This section describes the implementation of the Modality Worklist Service Class. As previously mentioned, this implementation is intended to be used as an *example* of one way a user may implement the Modality Worklist Service Class; it is not a full implementation. The sample does, however, give the user a good feel for the use of the Merge DICOM Toolkit.

These sample applications support both the Modality Worklist and Modality Performed Procedure Step Service Classes. However, this documentation currently only describes the support for the Modality Worklist Service Class.

The Modality Worklist Service Class is implemented by two peer DICOM Application Entities (AE's) with one AE acting as the service class provider (**SCP**) and the other acting as the service class user (**SCU**). The SCP accepts and services the DICOM DIMSE-C command, C-FIND. Commands are constructed and sent to the SCP by the SCU. The following figure outlines this interaction.

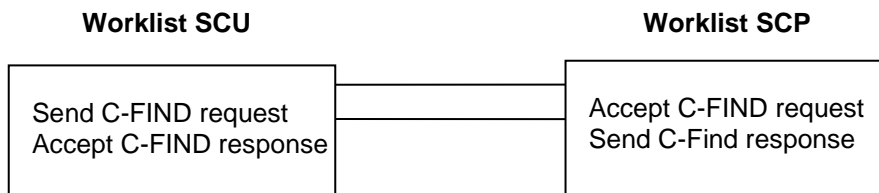


Figure 8: Modality Worklist SCU and SCP Interaction

The basic user-provider interaction scenario of the sample Modality Worklist application proceeds as follows. A more detailed description is given in the individual program sections.

1. The SCU requests that the SCP perform a search on the information it possesses. The SCU forms a message which contains a C-FIND request and information pertaining to the data it would like the SCP to match.
2. The SCP searches through the data it possesses. It then generates a response containing a unique key for each match it finds. The response message contains a PENDING command status and the information which was requested by the SCU if the operation was successful.

Otherwise, a blank message is formed and an error status is returned to the SCU. If no error exists, the SCP continues to send `PENDING` response messages until the final match is found. The SCP then returns `SUCCESS`.

3. The SCU receives each response message and interprets the status and data. The SCU reads `C-FIND` responses from the SCP until a status of `SUCCESS` is received, or the SCU receives more responses than it is configured to handle. Should it receive more responses than it can handle, the SCU sends a `C_CANCEL` request back to the SCU, and continues with attempting to read responses.
4. The SCU then generates `C-FIND` requests to the SCP by sending unique key values to the SCP. The SCP again formats response `PENDING` and ultimately, `SUCCESS` messages, and sends them to the SCU.

Configuration

Merge DICOM Toolkit Configuration information for the sample applications are kept in configuration files. These files contain initialization and startup information used by the sample applications as they execute. There are 4 different configuration files necessary for execution by any one application. They are: the Initialization file (referred to as "the dot ini file"), the Network Profile (referred to as "the dot pro file"), the Application Profile (referred to as "the dot app file") and the Service file (referred to as "the dot S-R-V file").

The configuration files follow the same format: a section starts with a label delimited with square brackets. Each item belonging to a section is then listed. The list is constructed of a variable followed by the equal sign (=) followed by the value of the variable.

For a more detailed discussion of the configuration files distributed with the DICOM Toolkit please see the files on the distribution itself and see the *Merge DICOM Toolkit Reference Manual*. Each file is fully documented and explains each item in detail.

Sample Client (User)

The Modality Worklist Service Class User is an Application Entity whose purpose is to send requests for find services to the Modality Worklist Service Class Provider.

The SCU is implemented in the C programming language using a program model suited for single-tasking operating systems. By using the single-tasking model, portability between single-tasking operating systems such as MS-DOS and multi-tasking operating systems such as UNIX become less of an issue.

Running the SCU

To run the SCU provided, execute the following command:

```
work_scu [options] REMOTE_AE_TITLE
```


where `options` are described below and `REMOTE_AE_TITLE` is the application entity title of the SCP that `work_scu` is attempting to connect with.

Please note that the above command assumes that you are in the proper directory and that your `MERGE_INI` environment variable has been set properly. Also note that the listen port has to be changed in the `mergecom.pro` file for the Modality Worklist SCU to match the Modality Worklist SCP's port number.

The command line options supported by `work_scu` are described in the following table.

Table 19: Optional command line parameters for the Modality Worklist SCU

Option	Parameter	Description
-a	AppTitle	This option specifies the Application Title for this application. The default is "MERGE_WORK_SCU." AppTitle may be any user defined string with a length of no greater than 16 characters.
-h	<none>	This option prints a short description on how to run <code>work_scu</code> .
-t	TimeOut	This option specifies the length of time-outs on network commands. TimeOut is specified in whole integer seconds. The default is 3000.
-p	PortNum	This option specifies the TCP listening port that the remote application is waiting for associations on.
-n	Hostname	This option specifies the host name of the worklist service class provider. That is, the remote's host name.

The SCU User Interface

The Modality Worklist SCU has been designed to include a simple user interface. The main menu appears below.

```

** MODALITY WORKLIST SEARCH **
Enter the number of the value to modify,
followed by the value that you wish to set.
e.g.: 1 CT
[1] Modality []
[2] Station Application Title []
[3] Procedure Start Date []
[4] Reply message cut-off threshold [15]
[Q] Perform a query
[X] Exit

==>
```

A typical scenario would be:

1. Enter the Modality, Station Application Title and Start Date.
2. Perform the query (do the C-FIND request)
3. Select a patient that was returned from the FIND.
4. Examine the Patient Data for correctness.
5. Exit

The Enter Data option allows data to be entered in the following manner. Type the option followed by the data (e.g. to enter a modality type "1 CT").

main

The `main` program begins by calling `GetOptions` to parse any command line options. Next, the application is registered in Merge DICOM Toolkit by calling `PerformInitialization`, which uses the Merge DICOM Toolkit library functions to perform this task.

The main menu is displayed to the user. The user now enters the query data. When finished with the data entry, the user can choose to perform a C-FIND. Once the user chooses to construct the C-Find command, `main` calls `OpenAssociation` in order to negotiate an association with the SCP. `main` then calls `SetAndSendMsg` to construct the C-FIND message and send it to the SCP. After the C-FIND message is constructed and sent, `main` calls `ProcessReplyMsg` to wait for the response messages from the SCP. After any response messages are received and processed the initial association with the SCP is closed. `GatherMoreData` is called to construct another C-FIND message on a single patient, while opening another association with the SCP, and closing that association when the second C-FIND has finished. Finally, `MC_Release_Application` is called to release the application from the library, once the user chooses to exit the program.

OpenAssociation

`OpenAssociation` is used to form an association with the Service Class Provider. Its only purpose is to call `MC_Open_Association`.

`OpenAssociation` does, however, open the association using the passed in command line parameters, if there were any specified.

SetAndSendMsg

`SetAndSendMsg` begins by opening a message. The message is constructed from the data that the user entered via the user interface (main menu) and by setting some of the other message parameters to the `NULL` value. Note that the message sent to the SCP also contains a sequence of items. The sequence is constructed by opening an item.

After the message has been created, it is sent to the SCP via a call to `MC_Send_Request_Message`.

`SetAndSendMsg` then returns to its caller.

ProcessReplyMsg

`ProcessReplyMessage` processes a C-FIND response message. It begins by attempting to read a message from the SCP. Once a message is obtained from the SCP, it is examined to see if it is a PENDING message, or a C_FIND_SUCCESS message. If the message is a C_FIND_SUCCESS message, all of the “data” from the messages (if any) has been processed, and `ProcessReplyMsg` can return to its caller.

If the message is a PENDING message, the message’s data is obtained from Merge DICOM Toolkit by calling the different `MC_Get_Value_To_...` functions. These functions are used to obtain the different fields from a message.

Once all of the desired fields of the message have been obtained from Merge DICOM Toolkit, the data is stored on a linked list. The linked list is used to provide a means of storing the data received from the SCP, until the user is able to act on this data. The operation of the linked list functions isn’t important as these functions would, most likely, be replaced by an “in-house” processing mechanism.

While the messages received from the SCP are C_FIND_PENDING messages, the data is obtained from the message, and stored onto the linked list. This process is repeated until a C_FIND_SUCCESS message is received.

SendCancelRQ

The `SendCancelRQ` function attempts to send a C_CANCEL_RQ message to the SCP. This function is called when the SCU receives response messages, greater in number than it is configured to accept. The threshold of messages is configurable through the main menu of the SCU. Keep in mind the fact that the SCP may still send a few messages after the SCU has sent the cancel request. This may occur when the SCP isn’t able to process the cancel request in a timely manner.

GatherMoreData

The `GatherMoreData` function works in a similar manner as the main menu. It displays a list of patients and the patient ID that was obtained from the original C-FIND request sent to the SCP. `GatherMoreData` starts by asking the user to select a single patient from this list and then calls `SetAndSendMsg` and `ProcessReplyMsg` again to perform another C-FIND request, followed by a wait for a C-FIND reply from the server. This time, the SCP should provide a single patient back to the SCU. The single patient data is displayed, and the user is then able to return to the main function of the program.

Sample Server (Provider)

The Modality Worklist Service Class Provider is a program whose purpose is to provide services to the Modality Worklist Service Class User.

The SCP, like the SCU, is implemented in the C programming language using a program model suited for single-tasking operating systems. By using the single-tasking model, portability between single-tasking operating systems such as MS-DOS and multi-tasking operating systems such as UNIX becomes less of an issue.

Running the SCP

To run the SCP provided, execute the following command:

```
work_scp [options]
```

where `options` are described below. Please note that the above command assumes that you are in the proper directory and that your `MERGE_INI` environment variable has been set properly.

The options supported by `work_scp` are described in the following table.

Table 20: Optional command line arguments for Modality Worklist SCP

Option	Parameter	Description
-a	AppTitle	This option specifies the Application Title for this application. The default is "MERGE_WORK_SCP." AppTitle may be any user defined string with a length of no greater than 16 characters.
-d	FileName	The -d option specifies the name of the file which contains the SCP data. See the DESCRIPTION section for details of the format of the data file. The default name is <code>work.dat</code> .
-h	<none>	This option prints a short description on how to run <code>work_scp</code> .
-t	TimeOut	This option specifies the length of time-outs on network commands. TimeOut is specified in whole integer seconds. The default is 3000.

main

The `main` program begins by setting all global options, parsing the command line and printing a header message. Next, the application is registered in Merge DICOM Toolkit by calling `MC_Register_Application`. This call provides Merge DICOM Toolkit with the information necessary to identify the SCP from other DICOM applications.

The SCP now goes into an "infinite" loop. Inside the loop, the SCP performs two functions: it waits for an association to be received from the SCU, and while the association is open, it waits for and replies to C-FIND request messages. The SCP waits for an association by calling `MC_Wait_For_Association`. This function is used to wait for another remote DICOM application to make a request for service. The association is "broken" or "let go" by calling `MC_Close_Association`. This function gracefully shuts down an open association and releases any resources bound to that association. Note that in multitasking environments, a "thread", or a new process could be created to handle the association activity.

When the program ends it calls `MC_Release_Application`. This call "de-registers" the application and releases the system resources used by the application.

ProcessCFINDRQ

The `ProcessCFINDRQ` is the function which performs most of the work for the C-FIND command. `ProcessCFINDRQ` reads the query fields from the message by calling `MC_Get_Value_To_...` for each of the fields.

After obtaining the information from the message, the function determines if there is a match with any of the data the SCP possesses by calling `PerformWorkSearch`. `PerformWorkSearch` is called in the following manner:

```
status = PerformWorkSearch ( SAE, station_ae, bIDarrayPS,  
global_data_file );
```

It is not important to understand how the search is performed because this routine will most likely be replaced by an "in-house" searching routine that queries a database or flat file instead of the unsophisticated linear search used here. The important thing to notice here is that this search is performed for each item requested.

If there were no errors up to this point, `ProcessCFINDRQ` determines if there were any complete matches and calls `SendCFINDReply` if there were.

SendCFINDReply

`SendCFINDReply` is the routine that allocates, constructs and sends the response message back to the SCU.

`SendCFINDReply` begins by calling `MC_Open_Message` to obtain a new message object. The object is opened to support `MODALITY_WORKLIST_FIND` within the C-FIND Response object.

Next, the object is populated with data. This is done by calling the `MC_Set_Value...` functions for each item the user has requested from the SCP.

Once the message has been populated, the reply message is sent off with a call to `MC_Send_Response_Message`. The reply message is marked as a PENDING reply. This process is repeated for each match found while querying the data.

After each reply message is sent to the SCU, the SCU is polled to see if it has sent any messages back to this SCP. In this implementation, the expected message is a `C_CANCEL_RQ` message. As mentioned before, should the SCU receive more responses than it can handle, it may send a `C_CANCEL_RQ` message to this SCP that needs to be handled. If the cancel message is waiting, the processing loop that sends all of the reply messages back to the SCU is exited.

Finally, after each response message has been sent back to the SCU or the cancel message has been processed, a final response message with a status of `C_FIND_CANCEL_REQUEST_RECEIVED` is sent back to the SCU signaling the end of the transaction.

qr_util.c

This module contains the `LogError` function, as well as the linked list manipulation functions. `LogError` is a general error logging function that is used to report the Merge DICOM Toolkit error messages in a uniform manner to the user. It can be used to report where the error has occurred, as well as define a severity of the error that had occurred.

The linked list functions are a very simple implementation of a generic linked list. By generic, it is meant that any data type, even user defined structures, can be inserted into the linked list. The list is meant for only one purpose. The purpose is to hold data in a sequential manner. There are currently eight functions that are used to create, manipulate, and destroy the linked list objects.

This module also contains functions which are used by the Query / Retrieve sample applications. Information on the use of any of these functions can be obtained by examining the sample application manual that pertains to that sample application.

In the case of the `LogError` and linked list functions, both of these will most likely be replaced with the user's own "in-house" functions. The purpose of these functions is not to take the place of a user's error logging and storage functions.

The Media Storage Service Class

The DICOM Media Storage Service Class defines the context for the transfer of images stored on media from one DICOM application entity to another. Following is an overview of the definition of the Media Storage Service Class as it relates to an application developer using the Merge DICOM Toolkit. If you require greater detail concerning the service class than is provided here, please refer to Parts 10, 11, and 12 of the DICOM standard.

Service Definition

The service definition can be broken down into the operations summarized in the following table.

Table 21: Media storage operations

Media Storage Operation	Definition	FSC	FSR	FSU
M-WRITE	Create new files and assign them a File ID.	ü		ü
M-READ	Read existing files based on their File ID.		ü	ü
M-DELETE	Delete existing files based on their File ID.			ü
M-INQUIRE FILE-SET	Inquire the free space available for creating new files within the File-set.	ü		ü

Media Storage Operation	Definition	FSC	FSR	FSU
M-INQUIRE FILE	Inquire the date and time of creation (or last update if applicable) for any files within the File-set based on their File ID.		ü	ü

A DICOM Application Entity will implement a set of these operations based on which of three roles it takes: **File-set Creator (FSC)**, **File-set Reader (FSR)**, or **File-set Updater (FSU)**. An FSC must support the M-WRITE and M-INQUIRE FILE-SET operations. An FSR supports the M-READ and M-INQUIRE FILE operations. Finally, an FSU supports all of the operations in Table 21. An Application Entity can choose to take any combination of these roles.

Merge DICOM Toolkit directly supports the M-WRITE and M-READ media operations. It aids the developer in formatting DICOM files for writing to media, and parses incoming DICOM files read from media. Merge DICOM Toolkit also aids in manipulation of the attributes in file objects. The remaining media operations are left for the developer to implement.

As required by the DICOM standard, any File-set must include a directory file with the File ID "DICOMDIR". This file contains general File-set information. It also includes a hierarchical directory structure which facilitates the access of files in the File-set based on key medical information. The structure is based on a hierarchy of Directory Entities. Each Directory Entity includes one or more Directory Records which in turn, may each reference a lower level Directory Entity. The Directory Records contained in this hierarchy reference each of the files contained in the File-set. Merge DICOM Toolkit includes functions for manipulating Directory Records and Directory Entities within a "DICOMDIR" directory file.

We will now look more closely at the behavior of FSCs, FSRs and FSUs.

File-set Creator Requirements

An FSC will perform the following actions involving creating a File-set on media:

1. Format media according to Part 12 of the standard.
2. Create a DICOMDIR file to contain the organization of the File-set.
3. Write zero or more DICOM files to media.
4. Handle requests for the amount of free space available in the File-set.

File-set Reader Requirements

A Media Storage Service Class FSR will perform the following actions:

1. Read DICOM files contained in a File-set.
2. Handle requests for the date and time of file creation (or update if applicable) of any file in a File-set.

File-set Updator Requirements

A Media Storage Service Class FSU will perform the following actions:

1. Read DICOM files contained in a File-set.
2. Add zero or more DICOM files to a File-set.
3. Update the DICOMDIR file in a File-set.
4. Delete DICOM files contained in a File-set.
5. Handle requests for the date and time of file creation (or update if applicable) of any file in the File-set.
6. Handle requests for the amount of free space available in the File-set.

Merge DICOM Toolkit handles the manipulation and data encoding of DICOM files (including DICOMDIRs). The developer is responsible for media specific functions such as inquiring the amount of space available on a file set, the date and time of file creation or update, and the actual reading from media and writing to media of encoded DICOM file objects. The sample application code described in this manual demonstrates how to use the Merge DICOM Toolkit to implement FSCs, FSRs and FSUs within the Media Storage Service Class.

Composite Services Supported

The DICOM standard specifies a number of composite services or “SOP Classes” which may be supported within the Media Storage Service Class by an FSC, FSR, or FSU (see *Table 22*). An FSC, FSR, or FSU may support all, or a subset of, these composite services and be conformant to the DICOM Media Storage Service Class.

Table 22: Media Storage Standard SOP Classes

Merge DICOM Toolkit Service Command Pair	SOP Class Name	SOP Class UID
DICOMDIR, C_STORE_RQ	Media Storage Directory Storage	1.2.840.10008.1.3.10
STANDARD_CR, C_STORE_RQ	Computed Radiography Image Storage	1.2.840.10008.5.1.4.1.1.1
STANDARD_CT, C_STORE_RQ	CT Image Storage	1.2.840.10008.5.1.4.1.1.2
STANDARD_US_MF, C_STORE_RQ	Ultrasound Multi-frame Image Storage	1.2.840.10008.5.1.4.1.1.3.1
STANDARD_MR, C_STORE_RQ	MR Image Storage	1.2.840.10008.5.1.4.1.1.4
STANDARD_NM, C_STORE_RQ	Nuclear Medicine Image Storage	1.2.840.10008.5.1.4.1.1.20
STANDARD_US, C_STORE_RQ	Ultrasound Image Storage	1.2.840.10008.5.1.4.1.1.6.1
STANDARD_SEC_CAPTURE, C_STORE_RQ	Secondary Capture Image Storage	1.2.840.10008.5.1.4.1.1.7

Merge DICOM Toolkit Service Command Pair	SOP Class Name	SOP Class UID
STANDARD_OVERLAY, C_STORE_RQ	Stand-alone Overlay Storage	1.2.840.10008.5.1.4.1.1.8
STANDARD_CURVE, C_STORE_RQ	Stand-alone Curve Storage	1.2.840.10008.5.1.4.1.1.9
STANDARD_MODALITY_LUT, C_STORE_RQ	Stand-alone Modality LUT Storage	1.2.840.10008.5.1.4.1.1.10
STANDARD_VOI_LUT, C_STORE_RQ	Stand-alone VOI LUT Storage	1.2.840.10008.5.1.4.1.1.11
STANDARD_XRAY_ANGIO, C_STORE_RQ	X-ray Angiographic Image Storage	1.2.840.10008.5.1.4.1.1.12.1
STANDARD_XRAY_RF, C_STORE_RQ	X-ray RadioFluoroscopic Image Storage	1.2.840.10008.5.1.4.1.1.12.2
STANDARD_XRAY_ANGIO_BIPLANE, C_STORE_RQ	X-ray Angiographic Bi-plane Image Storage	1.2.840.10008.5.1.4.1.1.12.3
STANDARD_PET, C_STORE_RQ	Positron Emission Tomography Image Storage	1.2.840.10008.5.1.4.1.1.128
STANDARD_PET_CURVE, C_STORE_RQ	Positron Emission Tomography Curve Storage	1.2.840.10008.5.1.4.1.1.129
STANDARD_RT_BEAMS_TREAT, C_STORE_RQ	RT Beams Treatment Record Storage	1.2.840.10008.5.1.4.1.1.481.4
STANDARD_RT_BRACHY_TREAT, C_STORE_RQ	RT Brachy Treatment Record Storage	1.2.840.10008.5.1.4.1.1.481.6
STANDARD_RT_IMAGE, C_STORE_RQ	RT Image Storage	1.2.840.10008.5.1.4.1.1.481.1
STANDARD_RT_DOSE, C_STORE_RQ	RT Dose Storage	1.2.840.10008.5.1.4.1.1.481.2
STANDARD_RT_STRUCTURE_SET, C_STORE_RQ	RT Structure Set Storage	1.2.840.10008.5.1.4.1.1.481.3
STANDARD_RT_PLAN, C_STORE_RQ	RT Plan Storage	1.2.840.10008.5.1.4.1.1.481.5
STANDARD_RT_TREAT_SUM, C_STORE_RQ	RT Treatment Summary Record Storage	1.2.840.10008.5.1.4.1.1.481.7
STANDARD_DX_PRESENT, C_STORE_RQ	Digital X-Ray Image Storage - For Presentation	1.2.840.10008.5.1.4.1.1.1.1
STANDARD_DX_PROCESS, C_STORE_RQ	Digital X-Ray Image Storage - For Processing	1.2.840.10008.5.1.4.1.1.1.1.1
STANDARD_IO_PRESENT, C_STORE_RQ	Digital Intra-oral X-Ray Image Storage - For Presentation	1.2.840.10008.5.1.4.1.1.1.3

Merge DICOM Toolkit Service Command Pair	SOP Class Name	SOP Class UID
STANDARD_IO_PPROCESS, C_STORE_RQ	Digital Intra-oral X-Ray Image Storage - For Processing	1.2.840.10008.5.1.4.1.1.1.3.1
STANDARD_MG_PRESENT, C_STORE_RQ	Digital Mammography Image Storage - For Presentation	1.2.840.10008.5.1.4.1.1.1.2
STANDARD_MG_PPROCESS, C_STORE_RQ	Digital Mammography Image Storage - For Processing	1.2.840.10008.5.1.4.1.1.1.2.1
STANDARD_HARDCOPY_COLOR, C_STORE_RQ	Hardcopy Color Image Storage	1.2.840.10008.5.1.1.30
STANDARD_HARDCOPY_GRAYSCALE, C_STORE_RQ	Hardcopy Grayscale Image Storage	1.2.840.10008.5.1.1.29
STANDARD_PRINT_STORAGE, C_STORE_RQ	Stored Print Storage	1.2.840.10008.5.1.1.27
STANDARD_VL_ENDOSCOPIC, C_STORE_RQ	VL Endoscopic Image Storage	1.2.840.10008.5.1.4.1.1.77.1.1
STANDARD_VL_MICROSCOPIC, C_STORE_RQ	VL Microscopic Image Storage	1.2.840.10008.5.1.4.1.1.77.1.2
STANDARD_VL_PHOTOGRAPHIC, C_STORE_RQ	VL Photographic Image Storage	1.2.840.10008.5.1.4.1.1.77.1.4
STANDARD_VL_SLIDE_MICROSCOPIC	VL Slide-Coordinates Microscopic Image Storage	1.2.840.10008.5.1.4.1.1.77.1.3
DETACHED_PATIENT_MANAGEMENT, N_GET_RQ	Detached Patient Management Storage	1.2.840.10008.3.1.2.1.1
DETACHED_VISIT_MANAGEMENT, N_GET_RQ	Detached Visit Management Storage	1.2.840.10008.3.1.2.2.1
DETACHED_STUDY_MANAGEMENT, N_GET_RQ	Detached Study Management Storage	1.2.840.10008.3.1.2.3.1
STUDY_COMPONENT_MANAGEMENT, N_GET_RQ	Detached Study Component Management Storage	1.2.840.10008.3.1.2.3.2
DETACHED_RESULTS_MANAGEMENT, N_GET_RQ	Detached Results Management Storage	1.2.840.10008.3.1.2.5.1
DETACHED_INTERP_MANAGEMENT, N_GET_RQ	Detached Interpretation Management Storage	1.2.840.10008.3.1.2.6.1
BASIC_FILM_SESSION, N_CREATE_RQ	Basic Film Session Storage	1.2.840.10008.5.1.1.1
BASIC_FILM_BOX, N_CREATE_RQ	Basic Film Box Storage	1.2.840.10008.5.1.1.2

Merge DICOM Toolkit Service Command Pair	SOP Class Name	SOP Class UID
BASIC_GRAYSCALE_IMAGE_BOX, N_CREATE_RQ	Basic Greyscale Image Box Storage	1.2.840.10008.5.1.1.4
BASIC_COLOR_IMAGE_BOX, N_CREATE_RQ	Basic Color Image Box Storage	1.2.840.10008.5.1.1.4.1

Valid Messages

Valid DICOM messages are defined in terms of a composite service and command. The file “message.txt”, which is included with your Merge DICOM Toolkit, contains the DICOM message formats. Below is an excerpt from the “message.txt” file for the STANDARD_CR composite service, C_STORE_RQ command. For instance, the example shows that attribute 0008,0020 representing STUDY_DATE, with a DA value representation, is present in this message.

```
#####
STANDARD_CR - C_STORE_RQ
#####
0008,0005 Specific Character set CS 1C
Condition: EXPANDED_OR_REPLACEMENT_CHARACTER_SET_USED
Defined Terms: ISO-IR 100, ISO-IR 101, ISO-IR 109, ISO-IR 110,
ISO-IR144, ISO-IR 127, ISO-IR 126, ISO-IR 138, ISO-IR 148
0008,0008 Image Type CS 3
Defined Terms: (ORIGINAL, DERIVED) (PRIMARY, SECONDARY)
0008,0012 Instance Creation Date DA 3
0008,0013 Instance Creation Time TM 3
0008,0014 Instance Creator UID UI 3
0008,0016 SOP Class UID UI 1
0008,0018 SOP Instance UID UI 1
0008,0020 Study Date DA 2
0008,0021 Series Date DA 3
0008,0022 Acquisition Date DA 3
```

Note: The Merge DICOM Toolkit function call `MC_Validate_Message` can be used to check the contents of a message for validity. There are however minor limitations when validating messages opened for composite services, as is the case in the Storage Service Class. Please see the `MC_Validate_Message` function description in the *Merge DICOM Toolkit Reference Manual* for a discussion of these limitations.

Media Storage Sample Application

The following discussion concerning the sample application is general in nature, and deals with concepts necessary in the creation of Media Storage Service Class applications which use Merge DICOM Toolkit. Please see the sample file “med_fsu.c” for specific coding examples.

The sample program was designed to be simple in its functionality, thereby exposing the basic framework upon which any Media Storage Service Class program is built using Merge DICOM Toolkit. This framework consists of a series of Merge DICOM Toolkit function calls which constitute your interface to DICOM in general, and the Media Storage Service Class in particular.

Configuration

The FSU sample Media Storage Service Class application requires configuration files which define communication parameters, levels of message logging, etc. Please see the “Configuration” section of the *Merge DICOM Toolkit User and Reference Manuals* for complete descriptions of the configuration files. Some important points to remember for this sample application are as follows:

FSU Configuration

The Application Entity Title for the sample FSU is MERGE_MEDIA_FSU. This Application Entity Title is hard programmed into the application itself.

You should configure the TCPIP_LISTEN_PORT in the System Profile (“mergecom.pro”) to an unused TCP/IP port. This port number is used by the Storage SCU application.

General Configuration

Don't forget to place the license number you received when you purchased the toolkit into the [ASSOC_PARMS] section of the System Profile (“mergecom.pro”).

Set the environmental variable MERGE_INI to point to the “merge.ini” file.

Note: The sample Storage Service Class and Media Storage Service Class programs are shipped with a single set of configuration files: merge.ini, mergecom.app, mergecom.pro, and mergecom.srv.

Sample FSU

Overview of Program Operation

The sample FSU handles associations and receives images from Storage Service Class SCUs. It will then place records of the images received within a DICOMDIR. It also will read in an already existing DICOMDIR. The sample FSU is invoked with command line arguments which determine where the DICOMDIR file (File-set) is located. These arguments take the form:

```
med_fsu [-d <directory>]
```

-d is an optional parameter that specifies the directory in which to place the File-set. Note that this directory can be the mount point of a file system.

The program also facilitates traversing the DICOMDIR and viewing directory records based on the current record within the DICOMDIR tree structure. The general flow of the sample Media Storage Service Class FSU can be charted as in *Figure 9*. Each of the numbered steps is described below in greater detail:

1. The command line argument is checked for validity. The existence of a DICOMDIR file within the current directory or in the directory specified by the argument -d is checked.
2. If a DICOMDIR was found in the previous step, it is read into memory here with the `MC_DDH_Open ()` function. If not, a new DICOMDIR object is created using `MC_DDH_Create ()`.
3. The main process loops while waiting for a menu selection from the user.
4. Based on the user input, various actions are performed.
5. If the user selected to wait for an association, an association listener is started.
6. When an association is received the code processes the C-STORE requests.
7. For each C-STORE request the received instance is saved in a Part 10 file and new records that reference that instance are added to the DICOMDIR file.

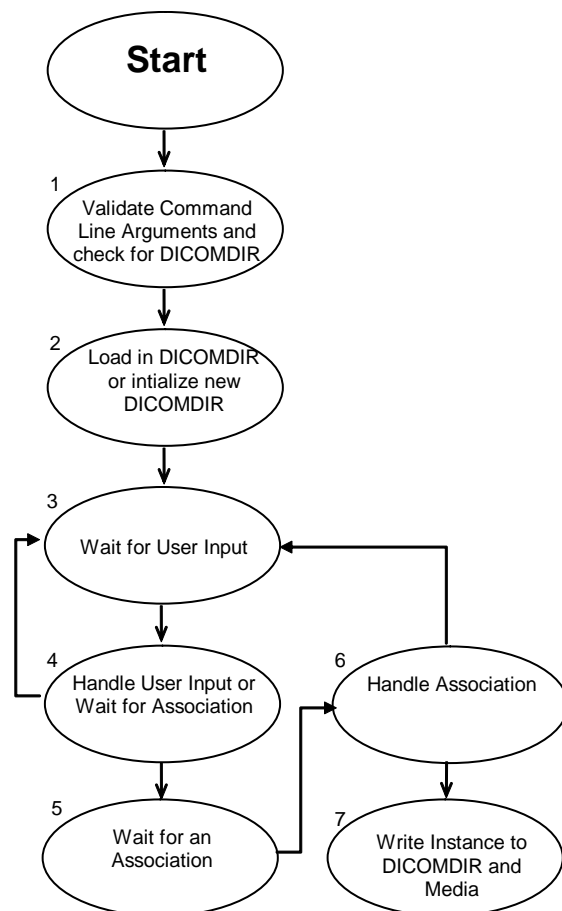


Figure 9: Sample Media FSU Overview

Duplicate Sample Application

Please see the sample file “duplicate.c” for specific coding examples.

The sample program was designed to be simple in its functionality, thereby exposing the basic framework upon which any program could be built using Merge DICOM Toolkit. “duplicate.c” shows the proper use of `MC_Duplicate_Message` when duplicating and changing the transfer syntax of a message is desired. This is mainly for when the source or destination is encapsulated. Changing transfer syntaxes between `IMPLICIT_LITTLE_ENDIAN`, `EXPLICIT_LITTLE_ENDIAN`, and `EXPLICIT_BIG_ENDIAN` are handled automatically by the toolkit on a transfer and would not require a duplication.

Note that although the sample application is included on all Merge DICOM Toolkit platforms, the complete compression functionality does not work on all platforms. An RLE compressor and decompressor is included on all platforms, however, the JPEG and JPEG2000 compressor and decompressor utilizes libraries from Pegasus Imaging and is not supported on all platforms. Please see the platform notes for your platform to determine if the Pegasus libraries are supported.

Configuration

The Duplicate Sample Application only requires general configuration since there is no network communication.

General Configuration

Don't forget to place the license number you received when you purchased the toolkit into the [ASSOC_PARMS] section of the System Profile ("mergecom.pro").

Set the environmental variable MERGE_INI to point to the "merge.ini" file.

Note: The sample applications are shipped with a single set of configuration files: merge.ini, mergecom.app, mergecom.pro, and mergecom.srv.

Sample Duplicate

Overview of Program Operation

The sample duplicate handles files or streams as input and creates a new file or stream(<originalFileName>_out.<originalExtension>) of a different transfer syntax. It can also "break out" multiple image files into separate image files. The sample duplicate is invoked with command line arguments which determine what is to be done with the input and output. These arguments take the form:

```
duplicate -f <filename> [-s <source format> -d <dest format> -v -b]
```

-f filename is a required parameter. It specifies the path/filename of the input file which is either part 10 DICOM, or in the "stream" format. The file/stream must be IMPLICIT_LITTLE_ENDIAN, EXPLICIT_LITTLE_ENDIAN, IMPLICIT_BIG_ENDIAN, EXPLICIT_BIG_ENDIAN, DEFLATED_EXPLICIT_LITTLE_ENDIAN, RLE, JPEG_BASELINE, JPEG_LOSSLESS_HIER_14, or JPEG_EXTENDED_2_4.

-s is an optional parameter that defaults to IMPLICIT_LITTLE_ENDIAN if not present. If the default is not correct, please use this option to correctly specify the transfer syntax of the stream. If this is a part 10 DICOM file, this option is not used because the transfer syntax is stored in the file.

-d is an optional parameter that defaults to JPEG_BASELINE if not present. If the default is not desired, please use this option to specify the destination's transfer syntax. This may be IMPLICIT_LITTLE_ENDIAN, EXPLICIT_LITTLE_ENDIAN, IMPLICIT_BIG_ENDIAN, EXPLICIT_BIG_ENDIAN, DEFLATED_EXPLICIT_LITTLE_ENDIAN, RLE, JPEG_BASELINE, JPEG_LOSSLESS_HIER_14, or JPEG_EXTENDED_2_4. If the destination format is the same as the source, no _out file is created. However, if you use "-b" in this instance, and the source is encapsulated, the image(s) will be "broken out" into separate file(s).

-v is an optional parameter for verbose. If you would like more information on what the program is doing, please use this flag.

-b is an optional parameter for "breaking out" single or multi-frame images contained in the destination file into one or more files, each containing a single

frame. The new file does not contain ANY DICOM tags or delimiters. It is strictly the pixel data. Note, the full pixel data is still stored in the _out file. The "break out" is in addition to the creation of the output. If the source format is the same as dest format, this option will work on the source file.

The general flow of the sample duplicate can be charted as in *Figure 10*. Each of the numbered steps is described below in greater detail:

1. The command line arguments are checked for validity. The source format is checked for validity now, unless part 10 DICOM file.
2. A stream file is read into a message. A part 10 DICOM file is read from file, transfer syntax checked for validity, and then is converted to a message.
3. Check if the source format is the same as the destination. If it is not, then proceed to step 4, otherwise go to step 7.
4. Callbacks must be registered for the source if the source message is encapsulated. `MC_Duplicate_Message` will complain if we do not to this. We will pass callbacks to `MC_Duplicate_Message` only if the destination will be encapsulated. `MC_Duplicate_Message` automatically will register these callbacks.
5. If we just created a new message with a lossy compression, the message requires a new SOP Instance UID. A new SOP Instance UID is REQUIRED when any part of a duplicated message is altered.
6. If the input was part 10 DICOM, then create the output file as part 10 DICOM. Otherwise, create a "stream" file.
7. If "break out" was selected as an option, get each image from the message and store it in a separate file. If the transfer syntax is `JPEG_BASELINE` or `JPEG_EXTENDED_2_4`, then the output will be of the format `<originalFileName>_frameX.jpg`. If `JPEG_LOSSLESS_HIER_14`, the extension is `.ljp`. These files will not contain DICOM information or delimiters.
8. Clean up messages/files, and release library.

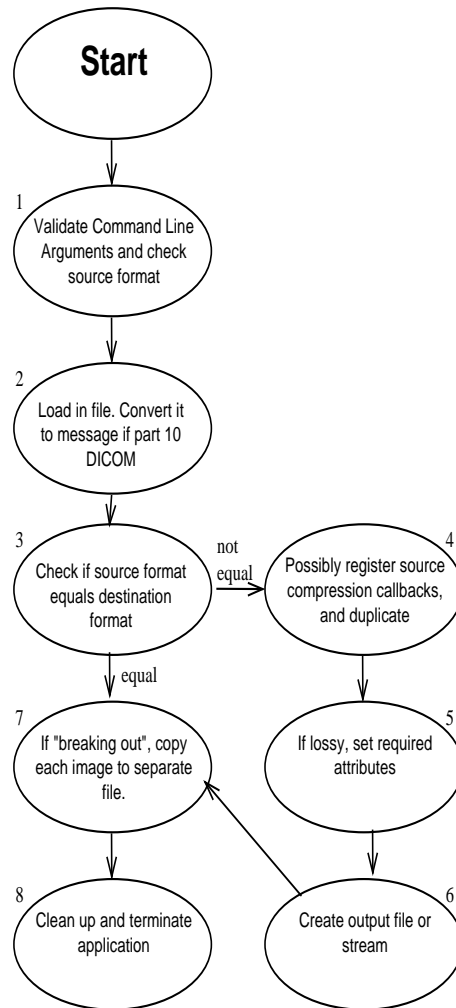


Figure 10: Sample Duplicate Overview

MPEG Conversion Sample Application

Please see the sample file "mpeg2Dicom.c" for specific coding examples.

The sample program was designed to be simple in its functionality, thereby exposing the basic framework upon which any program could be built using Merge DICOM Toolkit. "mpeg2Dicom.c" shows the proper use of following functions for manipulating the pixel data:

- MC_Set_Encapsulated_Value_From_Function
- MC_Get_Encapsulated_Value_To_Function

The sample code provides two basic functions:

- Wrapping a standard MPEG2 file into DICOM file.
- Extracting an MPEG2 stream from the DICOM file.

Note: This sample code uses hardcoded values for the MPEG image size as 512x512@fps to avoid parsing the MPEG header. The DICOM header data is hardcoded as well.

Configuration

The MPEG Conversion Sample Application only requires general configuration since there is no network communication.

General Configuration

Don't forget to place the license number you received when you purchased the toolkit into the [ASSOC_PARMS] section of the System Profile ("mergecom.pro").

Set the environmental variable MERGE_INI to point to the "merge.ini" file.

Note: The sample applications are shipped with a single set of configuration files: merge.ini, mergecom.app, mergecom.pro, and mergecom.srv.

MPEG Conversion

Overview of Program Operation

The MPEG Conversion sample is invoked with command line arguments which determine what is to be done with the input and output. These arguments take the form:

```
mpeg2Dicom -p | -u <inputFile> <outputFile>
```

-p Pack MPEG2 stream into DICOM file

-u Unpack MPEG2 stream from the DICOM file

COMP Sample Application

The COMP sample was designed to expose in a simple manner the functionality of multi-frame pixel data access using the callback mechanisms. The sample reads an input file from the data storage, creates compressed output files, then reads pixel data frames from them.

Configuration

The COMP Sample requires general standard *Merge DICOM Toolkit* configuration and by default uses an evaluation copy of Accusoft's AIMTools™ compression/decompression toolkit.

Note: The sample applications are shipped with a single set of configuration files: merge.ini, mergecom.app, mergecom.pro, and mergecom.srv.

Overview of Program Operation

COMP sample is invoked with command line arguments which specify the data folder and input file in the form:

```
comp -f 1.3.6.1.4.dcm -d ./data
```

As a result the sample will create in data folder a list of output compressed files with various Transfer Syntax attributes.

Internally, the sample demonstrates how to use a set of callback function to access a multi-frame pixel data using a callback technique.

1. The command line argument is checked for validity and displays a help message in case of an error.
2. If a multi-frame input DICOM file was found, the sample loops through various Transfer Syntax list and creates compressed DICOM files in a data folder using `MC_Duplicate_Message` API.
3. Registers a callback function for "on demand" reading of Pixel Data attribute.
4. Loops through all new files and reads DICOM attributes bypassing the Pixel Data using `MC_Open_File_Upto_Tag_Bypass_Value` API.
5. Retrieves the offset table and multi-frame pixel data using API:

```
MC_Get_Encapsulated_Value_To_Function
```

```
MC_Get_Next_Encapsulated_Value_To_Function
```

```
MC_Get_Frame_To_Function
```

```
MC_Get_Offset_Table_To_Function
```

Appendix A: Storage SCU Conformance Statement

0. Introduction

This is a conformance statement for the Merge Healthcare sample program (MERGE_STORE_SCU) which supports DICOM Storage Services as a Service Class User (SCU).

DICOM has been implemented by Merge Healthcare and is called Merge DICOM Toolkit. Therefore, Merge DICOM Toolkit and DICOM can and are used synonymously within this document.

1. Implementation model

MERGE_STORE_SCU with Merge DICOM Toolkit input and output is, very basically, an implementation of a DICOM Storage Service Class user (SCU) which can send DICOM images to a DICOM Storage Service Class provider (SCP).

1.1 Application data flow diagram

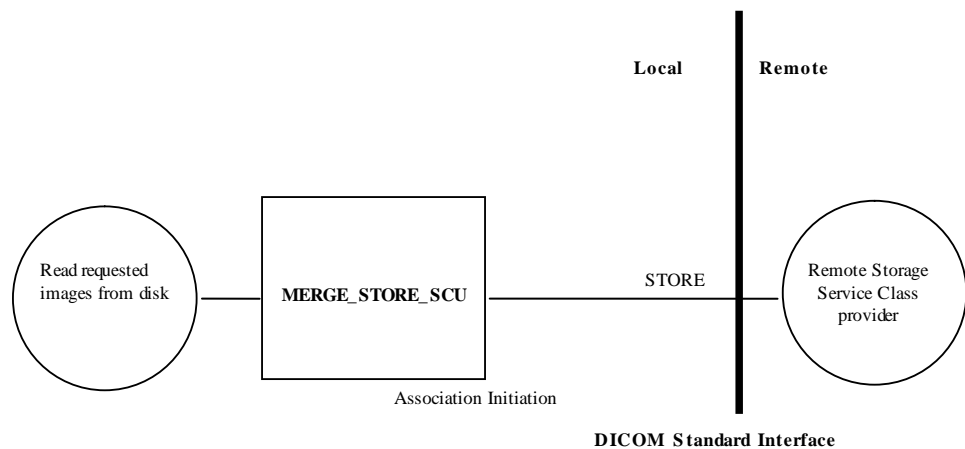


Figure A.1: MERGE_STORE_SCU application data flow diagram

1.2 Functional definition of Application Entity (AE)

All communications and image transfer with the remote application is accomplished utilizing the DICOM protocol over a network using the TCP/IP protocol stack. It establishes an association with a user selected remote AE just prior to sending a Store request to that AE.

1.3 Sequencing of real-world activities

Not applicable.

2. AE specifications

MERGE_STORE_SCU, in conjunction with Merge DICOM Toolkit, provides Standard Conformance to the following DICOM V3.0 Service Object Pair (SOP) Classes as a Storage Service Class User (SCU).

Table A.1: Valid SCU Storage SOP Classes for MERGE_STORE_SCU

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.1	Computed Radiography Image Storage
1.2.840.10008.5.1.4.1.1.2	CT Image Storage
1.2.840.10008.5.1.4.1.1.3	Ultrasound Multi-frame Image Storage (Retired)
1.2.840.10008.5.1.4.1.1.3.1	Ultrasound Multi-frame Image Storage
1.2.840.10008.5.1.4.1.1.4	MR Image Storage
1.2.840.10008.5.1.4.1.1.20	Nuclear Medicine Image Storage
1.2.840.10008.5.1.4.1.1.5	Nuclear Medicine Image Storage (Retired)
1.2.840.10008.5.1.4.1.1.6	Ultrasound Image Storage (Retired)
1.2.840.10008.5.1.4.1.1.6.1	Ultrasound Image Storage
1.2.840.10008.5.1.4.1.1.7	Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.7.2	Multi-frame Grayscale Byte Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.7.3	Multi-frame Grayscale Word Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.7.1	Multi-frame Single Bit Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.7.4	Mulfi-frame True Color Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.8	Standalone Overlay Storage
1.2.840.10008.5.1.4.1.1.9	Standalone Curve Storage
1.2.840.10008.5.1.4.1.1.10	Standalone Modality LUT Storage
1.2.840.10008.5.1.4.1.1.11	Standalone VOI LUT Storage
1.2.840.10008.5.1.4.1.1.12.1	Standard Xray Angio
1.2.840.10008.5.1.4.1.1.12.2	Standard Xray RF
1.2.840.10008.5.1.4.1.1.12.2	Standard Xray Angio Biplane

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.128	Standard Positron Emission Tomography Image Storage
1.2.840.10008.5.1.4.1.1.129	Standard Positron Emission Tomography Curve Storage
1.2.840.10008.5.1.4.1.1.481.4	Standard RT Beams Treatment Record Storage
1.2.840.10008.5.1.4.1.1.481.6	Standard RT Brachy Treatment Record Storage
1.2.840.10008.5.1.4.1.1.481.1	Standard RT Image Storage
1.2.840.10008.5.1.4.1.1.481.2	Standard RT Dose Storage
1.2.840.10008.5.1.4.1.1.481.3	Standard RT Structure Set Storage
1.2.840.10008.5.1.4.1.1.481.9	RT Ion Beams Treatment Record Storage
1.2.840.10008.5.1.4.1.1.481.8	RT Ion Plan Storage
1.2.840.10008.5.1.4.1.1.481.5	Standard RT Plan Storage
1.2.840.10008.5.1.4.1.1.481.7	Standard RT Treatment Summary Record Storage
1.2.840.10008.5.1.1.30	Hardcopy Color Image Storage
1.2.840.10008.5.1.1.29	Hardcopy Grayscale Image Storage
1.2.840.10008.5.1.1.27	Stored Print Storage
1.2.840.10008.5.1.4.1.1.1.1	Digital X-Ray Image Storage - For Presentation
1.2.840.10008.5.1.4.1.1.1.1.1	Digital X-Ray Image Storage - For Processing
1.2.840.10008.5.1.4.1.1.1.3	Digital Intra-oral X-Ray Image Storage - For Presentation
1.2.840.10008.5.1.4.1.1.1.3.1	Digital Intra-oral X-Ray Image Storage - For Processing
1.2.840.10008.5.1.4.1.1.1.2	Digital Mammography Image Storage - For Presentation
1.2.840.10008.5.1.4.1.1.1.2.1	Digital Mammography Image Storage - For Processing
1.2.840.10008.5.1.4.1.1.77.1.1	VL Endoscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.2	VL Microscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.4	VL Photographic Image Storage

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.77.1.3	VL Slide-Coordinates Microscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.1.1	Video Endoscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.2.1	Video Microscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.4.1	Video Photographic Image Storage
1.2.840.10008.5.1.4.1.1.88.65	Chest CAD SR Storage
1.2.840.10008.5.1.4.1.1.88.59	Key Object Selection Document
1.2.840.10008.5.1.4.1.1.88.50	Mammography CAD SR
1.2.840.10008.5.1.4.1.1.88.40	Procedure Log Storage
1.2.840.10008.5.1.4.1.1.88.11	Basic Text Structured Reporting
1.2.840.10008.5.1.4.1.1.88.33	Comprehensive Structured Reporting
1.2.840.10008.5.1.4.1.1.88.22	Enhanced Structured Reporting
1.2.840.10008.5.1.4.1.1.104.1	Encapsulated PDF Storage
1.2.840.10008.5.1.4.1.1.88.67	X-Ray Radiation Dose SR
1.2.840.10008.5.1.4.1.1.11.1	Grayscale Softcopy Presentation State Storage
1.2.840.10008.5.1.4.1.1.11.3	Pseudo-Color Softcopy Presentation State Storage
1.2.840.10008.5.1.4.1.1.11.2	Color Softcopy Presentation State Storage
1.2.840.10008.5.1.4.1.1.11.4	Blending Softcopy Presentation State Storage
1.2.840.10008.5.1.4.1.1.2.1	Enhanced CT Image Storage
1.2.840.10008.5.1.4.1.1.4.1	Enhanced MR Image Storage
1.2.840.10008.5.1.4.1.1.12.1.1	Enhanced XA Image Storage
1.2.840.10008.5.1.4.1.1.12.2.1	Enhanced XRF Image Storage
1.2.840.10008.5.1.4.1.1.4.2	MR Spectroscopy Storage
1.2.840.10008.5.1.4.1.1.66	Raw Data Storage
1.2.840.10008.5.1.4.1.1.66.1	Spatial Registration Storage
1.2.840.10008.5.1.4.1.1.66.2	Spatial Fiducials Storage
1.2.840.10008.5.1.4.1.1.77.1.5.3	Stereometric Relationship Storage

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.67	Real World Value Mapping Storage
1.2.840.10008.5.1.4.1.1.66.3	Deformable Spatial Registration
1.2.840.10008.5.1.4.1.1.66.4	Segmentation Storage
1.2.840.10008.5.1.4.1.1.77.1.5.2	Ophthalmic 16 bit Photography Image Storage
1.2.840.10008.5.1.4.1.1.77.1.5.1	Ophthalmic 8 bit Photography Image
1.2.840.10008.5.1.4.1.1.9.1.1	12-lead ECG Waveform Storage
1.2.840.10008.5.1.4.1.1.9.1.3	Ambulatory ECG Waveform Storage
1.2.840.10008.5.1.4.1.1.9.4.1	Basic Voice Audio Waveform Storage
1.2.840.10008.5.1.4.1.1.9.3.1	Cardiac Electrophysiology Waveform Storage
1.2.840.10008.5.1.4.1.1.9.1.2	General ECG Waveform Storage
1.2.840.10008.5.1.4.1.1.9.2.1	Hemodynamic Waveform Storage

2.1.1 Association establishment policies

2.1.1.1 General

The MERGE_STORE_SCU AE will initiate an association as an SCU of Storage Services when a local operator requests to send images over the network to a remote Storage Service Class provider. The maximum PDU size is configurable from a minimum of 4,096 bytes.

2.1.1.2 Number of associations

The MERGE_STORE_SCU AE only opens 1 Store association at a time. The operator may select that the sequence of images be sent multiple times; in which case, the MERGE_STORE_SCU AE will open multiple non-simultaneous associations with the remote AE.

2.1.1.3 Asynchronous nature

The MERGE_STORE_SCU AE supports asynchronous communication (multiple outstanding transactions over a single association).

2.1.1.4 Implementation identifying information

The Implementation Class Unique Identifier (UID) for the MERGE_STORE_SCU Application Entity is:

2.16.840.1.113669.2.1.1 (place your Implementation Class UID here).

The Implementation Version Name for the MERGE_STORE_SCU Application Entity is:

MergeCOM3_370 (place your Implementation Version Name here).

2.1.2 Association initiation by real-world activity

The MERGE_STORE_SCU AE initiates an association for the appropriate Storage Services Class that corresponds to the set of images requested to be transferred. The association is closed when all images have been sent to the remote DICOM network node. The client is also able to abort the association when an error occurs.

2.1.2.1 Real-world activity for Send Image operations

The MERGE_STORE_SCU AE initiates associations for the transfer of images to a DICOM Image Storage Server. The types of images that can be transferred are specified in the mergecom.app configuration file.

2.1.2.1.1 Associated real-world activity for Send Image operations

Once the Store Image association has been established, an image Store message is sent by MERGE_STORE_SCU.

2.1.2.1.2 Proposed presentation contexts for Send Image operations

The presentation contexts that are proposed by MERGE_STORE_SCU AE for the Send Image operation are specified in the following table.

Table A.2: Send Image Presentation Contexts of MERGE_STORE_SCU

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
All Services Listed in Table A.1	All Services Listed in Table A.1	Explicit VR Little Endian	1.2.840.10008.1.2.1	SCU	None
		Implicit VR Little Endian	1.2.840.10008.1.2		
		Explicit VR Big Endian	1.2.840.10008.1.2.2		

All these SOP classes conform to the standard Storage Services as specified in the DICOM Standard.

3. Communication profiles

3.1 Supported Communication Stacks

MERGE_STORE_SCU, in conjunction with Merge DICOM Toolkit provides DICOM V3.0 TCP/IP Network Communication Support as defined in PS 3.8.

3.2 TCP/IP Stack

MERGE_STORE_SCU uses the Merge DICOM Toolkit to communicate over the TCP/IP protocol stack on any physical interconnection media supporting the TCP/IP stack. The toolkit inherits the TCP/IP stack from the host operating system upon which it executes. The Toolkit has been implemented on almost every major operating system platform.

3.2.1 Physical Media Support

The MERGE_STORE_SCU AE is indifferent to the physical medium over which TCP/IP executes; it inherits this from the operating system on which it exists.

4. Extensions/specializations/privatizations

4.1 Standard extended/specialized/private SOPs

None supported.

4.2 Private Transfer Syntaxes

None supported.

5. Configuration

The MERGE_STORE_SCU application references four configuration files. The first, merge.ini, is found through the MERGE_INI environment variable. They are as follows:

merge.ini	Specifies the names of the other three configuration files and also contains message logging parameters.
mergecom.pro	Specifies run-time parameters for the MERGE_STORE_SCU application.
mergecom.app	Defines applications on other network nodes, to which connections are possible.
mergecom.srv	Service and sequence definitions.

5.1 AE title/presentation address mapping

Presentation address mapping is configured in the mergecom.app file. This is where the Host Name, Port Number, and Application Title map an Application Entity (AE) Title to a Presentation Address in TCP/IP for the provider to which you wish to connect.

Note: The host name maps to an IP address as specified by your host table.

5.2 Configurable parameters

The mergecom.pro configuration file can be used to set or modify other lower-level communication parameters. This includes time-outs and other parameters. Some information about supported SOP classes is also stored here. **Most parameters in this file should NEVER be changed. Doing so could break DICOM conformance.** Before modifying any parameters, such as time-out, be sure to have a backup of the originally supplied mergecom.pro file. Also, before modifying other parameters, you should consider contacting Merge Healthcare for advice.

6. Support of extended character sets

Not supported.

Appendix B: Storage SCP Conformance Statement

0. Introduction

This is a conformance statement for the Merge Healthcare sample program (MERGE_STORE_SCP) which supports DICOM Storage Services as a Service Class Provider (SCP).

DICOM has been implemented by Merge Healthcare and is called Merge DICOM Toolkit. Therefore, Merge DICOM Toolkit and DICOM can and are used synonymously within this document.

1. Implementation model

MERGE_STORE_SCP with Merge DICOM Toolkit input and output is, very basically, an implementation of a DICOM Storage Service Class provider (SCP) which can receive DICOM images from a DICOM Storage Service Class user (SCU).

1.1 Application data flow diagram

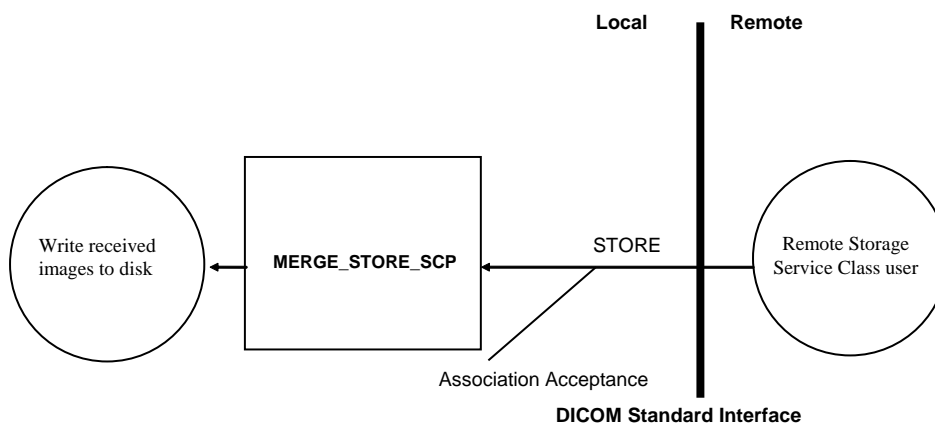


Figure B.1: MERGE_STORE_SCP application data flow diagram

1.2 Functional definition of Application Entity (AE)

All communications and image transfer with the remote application is accomplished utilizing the DICOM protocol over a network using the TCP/IP protocol stack. MERGE_STORE_SCP will respond, if asked, with the Verification SOP Class UID as an SCP for one of its implemented SOP classes. MERGE_STORE_SCP waits for an association to accept at the TCP/IP port number that is configured at the time this application is initiated. When an association request is received with valid connections criteria, MERGE_STORE_SCP responds with a list of SOP class UIDs that it will accept. It then waits for a Store request. If a Store is received, then all incoming images that are conformant to the association are either written to files on disk, listed, or discarded depending on the command-line arguments used when the application was initiated.

1.3 Sequencing of real-world activities

Not applicable.

2. AE specifications

2.1 AE Specification for MERGE_STORE_SCP

MERGE_STORE_SCP with Merge DICOM Toolkit, provides Standard Conformance to the following DICOM V3.0 Service Object Pair (SOP) Class as a Verification Service Class Provider (SCP). As an SCP it sends out an Echo response after it receives an Echo request from a remote AE.

Table B.1: Valid SCP Verification SOP Class for MERGE_STORE_SCP AE

SOP Class UID	SOP Class Name
1.2.840.10008.1.1	Verification SOP Class

MERGE_STORE_SCP, in conjunction with Merge DICOM Toolkit, provides Standard Conformance to the following DICOM V3.0 Service Object Pair (SOP) Classes as a Storage Service Class Provider (SCP).

Table B.2: Valid SCP Storage SOP Classes for MERGE_STORE_SCP

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.1	Computed Radiography Image Storage
1.2.840.10008.5.1.4.1.1.2	CT Image Storage
1.2.840.10008.5.1.4.1.1.3	Ultrasound Multi-frame Image Storage (Retired)
1.2.840.10008.5.1.4.1.1.3.1	Ultrasound Multi-frame Image Storage
1.2.840.10008.5.1.4.1.1.4	MR Image Storage
1.2.840.10008.5.1.4.1.1.20	Nuclear Medicine Image Storage
1.2.840.10008.5.1.4.1.1.5	Nuclear Medicine Image Storage (Retired)
1.2.840.10008.5.1.4.1.1.6	Ultrasound Image Storage (Retired)
1.2.840.10008.5.1.4.1.1.6.1	Ultrasound Image Storage
1.2.840.10008.5.1.4.1.1.7	Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.7.2	Multi-frame Grayscale Byte Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.7.3	Multi-frame Grayscale Word Secondary Capture Image Storage

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.7.1	Multi-frame Single Bit Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.7.4	Mulfi-frame True Color Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.8	Standalone Overlay Storage
1.2.840.10008.5.1.4.1.1.9	Standalone Curve Storage
1.2.840.10008.5.1.4.1.1.10	Standalone Modality LUT Storage
1.2.840.10008.5.1.4.1.1.11	Standalone VOI LUT Storage
1.2.840.10008.5.1.4.1.1.12.1	Standard Xray Angio
1.2.840.10008.5.1.4.1.1.12.2	Standard Xray RF
1.2.840.10008.5.1.4.1.1.12.2	Standard Xray Angio Biplane
1.2.840.10008.5.1.4.1.1.128	Standard Positron Emission Tomography Image Storage
1.2.840.10008.5.1.4.1.1.129	Standard Positron Emission Tomography Curve Storage
1.2.840.10008.5.1.4.1.1.481.4	Standard RT Beams Treatment Record Storage
1.2.840.10008.5.1.4.1.1.481.6	Standard RT Brachy Treatment Record Storage
1.2.840.10008.5.1.4.1.1.481.1	Standard RT Image Storage
1.2.840.10008.5.1.4.1.1.481.2	Standard RT Dose Storage
1.2.840.10008.5.1.4.1.1.481.3	Standard RT Structure Set Storage
1.2.840.10008.5.1.4.1.1.481.9	RT Ion Beams Treatment Record Storage
1.2.840.10008.5.1.4.1.1.481.8	RT Ion Plan Storage
1.2.840.10008.5.1.4.1.1.481.5	Standard RT Plan Storage
1.2.840.10008.5.1.4.1.1.481.7	Standard RT Treatment Summary Record Storage
1.2.840.10008.5.1.1.30	Hardcopy Color Image Storage
1.2.840.10008.5.1.1.29	Hardcopy Grayscale Image Storage
1.2.840.10008.5.1.1.27	Stored Print Storage
1.2.840.10008.5.1.4.1.1.1.1	Digital X-Ray Image Storage - For Presentation

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.1.1.1	Digital X-Ray Image Storage - For Processing
1.2.840.10008.5.1.4.1.1.1.3	Digital Intra-oral X-Ray Image Storage - For Presentation
1.2.840.10008.5.1.4.1.1.1.3.1	Digital Intra-oral X-Ray Image Storage - For Processing
1.2.840.10008.5.1.4.1.1.1.2	Digital Mammography Image Storage - For Presentation
1.2.840.10008.5.1.4.1.1.1.2.1	Digital Mammography Image Storage - For Processing
1.2.840.10008.5.1.4.1.1.77.1.1	VL Endoscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.2	VL Microscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.4	VL Photographic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.3	VL Slide-Coordinates Microscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.1.1	Video Endoscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.2.1	Video Microscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.4.1	Video Photographic Image Storage
1.2.840.10008.5.1.4.1.1.88.65	Chest CAD SR Storage
1.2.840.10008.5.1.4.1.1.88.59	Key Object Selection Document
1.2.840.10008.5.1.4.1.1.88.50	Mammography CAD SR
1.2.840.10008.5.1.4.1.1.88.40	Procedure Log Storage
1.2.840.10008.5.1.4.1.1.88.11	Basic Text Structured Reporting
1.2.840.10008.5.1.4.1.1.88.33	Comprehensive Structured Reporting
1.2.840.10008.5.1.4.1.1.88.22	Enhanced Structured Reporting
1.2.840.10008.5.1.4.1.1.104.1	Encapsulated PDF Storage
1.2.840.10008.5.1.4.1.1.88.67	X-Ray Radiation Dose SR
1.2.840.10008.5.1.4.1.1.11.1	Grayscale Softcopy Presentation State Storage
1.2.840.10008.5.1.4.1.1.11.3	Pseudo-Color Softcopy Presentation State Storage
1.2.840.10008.5.1.4.1.1.11.2	Color Softcopy Presentation State Storage

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.11.4	Blending Softcopy Presentation State Storage
1.2.840.10008.5.1.4.1.1.2.1	Enhanced CT Image Storage
1.2.840.10008.5.1.4.1.1.4.1	Enhanced MR Image Storage
1.2.840.10008.5.1.4.1.1.12.1.1	Enhanced XA Image Storage
1.2.840.10008.5.1.4.1.1.12.2.1	Enhanced XRF Image Storage
1.2.840.10008.5.1.4.1.1.4.2	MR Spectroscopy Storage
1.2.840.10008.5.1.4.1.1.66	Raw Data Storage
1.2.840.10008.5.1.4.1.1.66.1	Spatial Registration Storage
1.2.840.10008.5.1.4.1.1.66.2	Spatial Fiducials Storage
1.2.840.10008.5.1.4.1.1.77.1.5.3	Stereometric Relationship Storage
1.2.840.10008.5.1.4.1.1.67	Real World Value Mapping Storage
1.2.840.10008.5.1.4.1.1.66.3	Deformable Spatial Registration
1.2.840.10008.5.1.4.1.1.66.4	Segmentation Storage
1.2.840.10008.5.1.4.1.1.77.1.5.2	Ophthalmic 16 bit Photography Image Storage
1.2.840.10008.5.1.4.1.1.77.1.5.1	Ophthalmic 8 bit Photography Image
1.2.840.10008.5.1.4.1.1.9.1.1	12-lead ECG Waveform Storage
1.2.840.10008.5.1.4.1.1.9.1.3	Ambulatory ECG Waveform Storage
1.2.840.10008.5.1.4.1.1.9.4.1	Basic Voice Audio Waveform Storage
1.2.840.10008.5.1.4.1.1.9.3.1	Cardiac Electrophysiology Waveform Storage
1.2.840.10008.5.1.4.1.1.9.1.2	General ECG Waveform Storage
1.2.840.10008.5.1.4.1.1.9.2.1	Hemodynamic Waveform Storage

2.1.1 Association establishment policies

2.1.1.1 General

The MERGE_STORE_SCP application will wait for an association as an SCP of Storage Services. When a Store request is received, the corresponding images are either saved to files on disk, listed, or discarded. The maximum PDU size is configurable from a minimum of 4,096 bytes.

2.1.1.2 Number of associations

The MERGE_STORE_SCP AE allows multiple simultaneous Store associations. There is no maximum number of associations.

2.1.1.3 Asynchronous nature

The MERGE_STORE_SCP AE supports asynchronous communication (multiple outstanding transactions over a single association).

2.1.1.4 Implementation identifying information

The Implementation Class Unique Identifier (UID) for the MERGE_STORE_SCP Application Entity is:

2.16.840.1.113669.2.1.1 (place your Implementation Class UID here).

The Implementation Version Name for the MERGE_STORE_SCP Application Entity is:

MergeCOM3_222 (place your Implementation Version Name here).

2.1.2 Association initiation for MERGE_STORE_SCP

The MERGE_STORE_SCP does not initiate any associations.

2.1.3 Association acceptance policy for MERGE_STORE_SCP AE

The MERGE_STORE_SCP client application accepts an association for the Verification and Storage Service Class.

MERGE_STORE_SCP is able to abort the association when an error occurs.

2.1.3.1 Real-world activity for Echo Response

The MERGE_STORE_SCP Application Entity waits for an association request and accepts associations to do, among other things, the Verification Service. The association is closed after an error or when the initiator requests that it be closed.

2.1.3.1.1 Presentation context table for Echo Response operation

Only the presentation context listed in the following table will be accepted by the MERGE_STORE_SCP for the Verification Service Class.

Table B.3: Echo Check Presentation Contexts of MERGE_STORE_SCP

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
Verification Service Class	1.2.840.10008.1.1	Explicit VR Little Endian Implicit VR Little Endian Explicit VR Big Endian	1.2.840.10008.1.2.1 1.2.840.10008.1.2 1.2.840.10008.1.2.2	SCP	None

2.1.3.2 Real-world activity for Receive Image operations

MERGE_STORE_SCP waits for an association and offers to do the Image Storage service. The association is closed after an error or when the initiator requests that it be closed.

2.1.3.2.1 Associated real-world activity for Receive Image operations

Once the association has been established, the MERGE_STORE_SCP waits for transmission of conformant Storage Service messages.

2.1.3.2.2 Presentation context table for Receive Image operations

Table B.4: Receive Image Presentation Contexts of MERGE_STORE_SCP

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
All Services in Table B.2	All Services in Table B.2	Explicit VR Little Endian Implicit VR Little Endian Explicit VR Big Endian	1.2.840.10008.1.2.1 1.2.840.10008.1.2 1.2.840.10008.1.2.2	SCP	None

2.1.3.2.2.1 SOP specific conformance for all storage SOP Classes

The MERGE_STORE_SCP AE responds to a C-STORE request with one of these response codes:

Table B.5: SOP specific conformance

Service Status	Status Description	Status Code (0000,0900)	Related Fields
REFUSED	Out of Resources - There were insufficient resources to process the request. The request was not processed	A765	(0000,0902) contains a short description of the condition.
ERROR	Data Set does not match SOP Class - A required attribute is not present in the message. The request was not processed.	A965	(0000,0901) contains a listing of attribute tags missing. (0000,0902) contains a short description of the condition.
	Cannot understand - The message was not properly DICOM-encoded. The request was not processed.	C065	(0000,0902) contains a short description of the condition.
	Processing failure - A condition arose which prevented the processing of the request	0111	None
SUCCESS		0000	None

2.1.3.2.2.2 Presentation context acceptance criterion for Receive Image operations

Not applicable since only a single presentation context for each Storage Service Class is supported.

2.1.3.2.2.3 Transfer syntax selection policies for Receive Image operations

When executing on a Little Endian machine, transfer syntaxes are accepted in the following order:

Transfer Syntax UID	Transfer Syntax Name
1.2.840.10008.1.2	Explicit Little Endian Syntax
1.2.840.10008.1.2.1	Implicit Little Endian Syntax
1.2.840.10008.1.2.2	Explicit Big Endian Syntax

When executing on a Big Endian machine, transfer syntaxes are accepted in the following order:

Transfer Syntax UID	Transfer Syntax Name
1.2.840.10008.1.2.2	Explicit Big Endian Syntax
1.2.840.10008.1.2.1	Explicit Little Endian Syntax
1.2.840.10008.1.2	Implicit Little Endian Syntax

Note: This acceptance criteria can be overridden by the use of a “Transfer Syntax List” in the mergecom.app configuration file.

3. Profiles

3.1 Supported Communication Stacks

MERGE_STORE_SCP, in conjunction with Merge DICOM Toolkit provides DICOM V3.0 TCP/IP Network Communication Support as defined in PS 3.8.

3.2 TCP/IP Stack

MERGE_STORE_SCP uses the Merge DICOM Toolkit to communicate over the TCP/IP protocol stack on any physical interconnection media supporting the TCP/IP stack. The Toolkit inherits the TCP/IP stack from the host operating system upon which it executes. The Toolkit has been implemented on almost every major operating system platform.

3.2.1 Physical Media Support

The MERGE_STORE_SCP AE is indifferent to the physical medium over which TCP/IP executes; it inherits this from the operating system on which it exists.

4. Extensions/specializations/privatizations

4.1 Standard extended/specialized/private SOPs

None supported.

4.2 Private Transfer Syntaxes

None supported.

5. Configuration

5.1 MERGE_STORE_SCP Configuration files

The MERGE_STORE_SCP application references four configuration files. The first, merge.ini, is found through the MERGE_INI environment variable. They are as follows:

merge.ini	Specifies the names of the other three configuration files and also contains message logging parameters.
mergecom.pro	Specifies run-time parameters for the MERGE_STORE_SCP application.
mergecom.app	Defines applications on other network nodes, to which connections are possible.
mergecom.srv	Service and sequence definitions.

5.1.1 AE title/presentation address mapping

Presentation address mapping is configured in the mergecom.app file. The Presentation Address of an SCP application as a provider is specified by configuring the Listen Port in the mergecom.pro file, and specifying the AE title for the SCP within the application itself.

5.1.2 Configurable parameters

The mergecom.pro configuration file can be used to set or modify other lower-level communication parameters. This includes time-outs and other parameters. Some information about supported SOP classes is also stored here. **Most parameters in this file should NEVER be changed. Doing so could break DICOM conformance.** Before modifying any parameters, such as time-out, be sure to have a backup of the originally supplied mergecom.pro file. Also, before modifying other parameters, you should consider contacting Merge Healthcare for advice.

6. Support of extended character sets

Not supported.

Appendix C: Query/Retrieve SCU Conformance Statement

0. Introduction

This is a conformance statement for the Merge Healthcare sample program (MERGE_QR_SCU) which supports DICOM Query/Retrieve Services as a Service Class User(SCU).

DICOM has been implemented by Merge Healthcare and is called Merge DICOM Toolkit. Therefore, Merge DICOM Toolkit and DICOM can and are used synonymously within this document.

1. Implementation Model

MERGE_QR_SCU with Merge DICOM Toolkit input and output is, very basically, an implementation of a DICOM Query/Retrieve Service Class User (SCU) which can send DICOM queries and move requests to a DICOM Storage Service Class provider (SCP).

1.1 Application data flow diagram

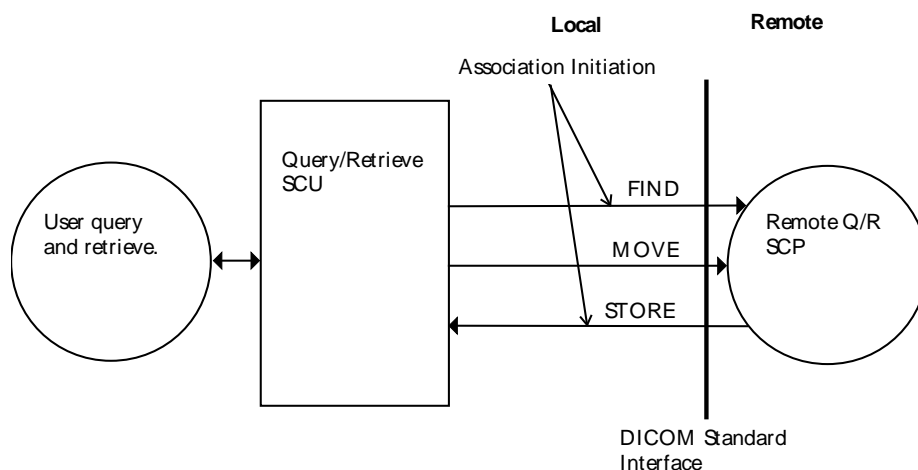


Figure C.1: Application data flow diagram form MERGE_QR_SCU

1.2 Functional definition of Application Entity (AE)

All communications and image transfer with the remote application is accomplished utilizing the DICOM protocol over a network using the TCP/IP protocol stack.

The MERGE_QR_SCU establishes an association with a user selected TCP/IP port number that is configured at the time this application is initiated. When an association is requested with a SCP, MERGE_QR_SCU responds with a list of SOP Class UIDs that it will accept. If a Find request is sent then it will wait for Find responses. If a Move request is sent, it will wait for a Move response.

2. AE Specifications

2.1 AE specification for MERGE_QR_SCU

MERGE_QR_SCU, in conjunction with Merge DICOM Toolkit, provides Standard Conformance to the following DICOM V3.0 Service Object Pair (SOP) Class as a Query Retrieve Service Class User (SCU).

Table C.1: Valid SCP STORE SOP Classes for MERGE_QR_SCU AE

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.1	Computed Radiography Image Storage
1.2.840.10008.5.1.4.1.1.2	CT Image Storage
1.2.840.10008.5.1.4.1.1.3	Ultrasound Multi-frame Image Storage (Retired)
1.2.840.10008.5.1.4.1.1.3.1	Ultrasound Multi-frame Image Storage
1.2.840.10008.5.1.4.1.1.4	MR Image Storage
1.2.840.10008.5.1.4.1.1.20	Nuclear Medicine Image Storage
1.2.840.10008.5.1.4.1.1.5	Nuclear Medicine Image Storage (Retired)
1.2.840.10008.5.1.4.1.1.6	Ultrasound Image Storage (Retired)
1.2.840.10008.5.1.4.1.1.6.1	Ultrasound Image Storage
1.2.840.10008.5.1.4.1.1.7	Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.7.2	Multi-frame Grayscale Byte Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.7.3	Multi-frame Grayscale Word Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.7.1	Multi-frame Single Bit Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.7.4	Multi-frame True Color Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.8	Standalone Overlay Storage
1.2.840.10008.5.1.4.1.1.9	Standalone Curve Storage
1.2.840.10008.5.1.4.1.1.10	Standalone Modality LUT Storage
1.2.840.10008.5.1.4.1.1.11	Standalone VOI LUT Storage
1.2.840.10008.5.1.4.1.1.12.1	Standard Xray Angio
1.2.840.10008.5.1.4.1.1.12.2	Standard Xray RF

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.12.2	Standard Xray Angio Biplane
1.2.840.10008.5.1.4.1.1.128	Standard Positron Emission Tomography Image Storage
1.2.840.10008.5.1.4.1.1.129	Standard Positron Emission Tomography Curve Storage
1.2.840.10008.5.1.4.1.1.481.4	Standard RT Beams Treatment Record Storage
1.2.840.10008.5.1.4.1.1.481.6	Standard RT Brachy Treatment Record Storage
1.2.840.10008.5.1.4.1.1.481.1	Standard RT Image Storage
1.2.840.10008.5.1.4.1.1.481.2	Standard RT Dose Storage
1.2.840.10008.5.1.4.1.1.481.3	Standard RT Structure Set Storage
1.2.840.10008.5.1.4.1.1.481.9	RT Ion Beams Treatment Record Storage
1.2.840.10008.5.1.4.1.1.481.8	RT Ion Plan Storage
1.2.840.10008.5.1.4.1.1.481.5	Standard RT Plan Storage
1.2.840.10008.5.1.4.1.1.481.7	Standard RT Treatment Summary Record Storage
1.2.840.10008.5.1.1.30	Hardcopy Color Image Storage
1.2.840.10008.5.1.1.29	Hardcopy Grayscale Image Storage
1.2.840.10008.5.1.1.27	Stored Print Storage
1.2.840.10008.5.1.4.1.1.1.1	Digital X-Ray Image Storage - For Presentation
1.2.840.10008.5.1.4.1.1.1.1.1	Digital X-Ray Image Storage - For Processing
1.2.840.10008.5.1.4.1.1.1.3	Digital Intra-oral X-Ray Image Storage - For Presentation
1.2.840.10008.5.1.4.1.1.1.3.1	Digital Intra-oral X-Ray Image Storage - For Processing
1.2.840.10008.5.1.4.1.1.1.2	Digital Mammography Image Storage - For Presentation
1.2.840.10008.5.1.4.1.1.1.2.1	Digital Mammography Image Storage - For Processing
1.2.840.10008.5.1.4.1.1.77.1.1	VL Endoscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.2	VL Microscopic Image Storage

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.77.1.4	VL Photographic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.3	VL Slide-Coordinates Microscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.1.1	Video Endoscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.2.1	Video Microscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.4.1	Video Photographic Image Storage
1.2.840.10008.5.1.4.1.1.88.65	Chest CAD SR Storage
1.2.840.10008.5.1.4.1.1.88.59	Key Object Selection Document
1.2.840.10008.5.1.4.1.1.88.50	Mammography CAD SR
1.2.840.10008.5.1.4.1.1.88.40	Procedure Log Storage
1.2.840.10008.5.1.4.1.1.88.11	Basic Text Structured Reporting
1.2.840.10008.5.1.4.1.1.88.33	Comprehensive Structured Reporting
1.2.840.10008.5.1.4.1.1.88.22	Enhanced Structured Reporting
1.2.840.10008.5.1.4.1.1.104.1	Encapsulated PDF Storage
1.2.840.10008.5.1.4.1.1.88.67	X-Ray Radiation Dose SR
1.2.840.10008.5.1.4.1.1.11.1	Grayscale Softcopy Presentation State Storage
1.2.840.10008.5.1.4.1.1.11.3	Pseudo-Color Softcopy Presentation State Storage
1.2.840.10008.5.1.4.1.1.11.2	Color Softcopy Presentation State Storage
1.2.840.10008.5.1.4.1.1.11.4	Blending Softcopy Presentation State Storage
1.2.840.10008.5.1.4.1.1.2.1	Enhanced CT Image Storage
1.2.840.10008.5.1.4.1.1.4.1	Enhanced MR Image Storage
1.2.840.10008.5.1.4.1.1.12.1.1	Enhanced XA Image Storage
1.2.840.10008.5.1.4.1.1.12.2.1	Enhanced XRF Image Storage
1.2.840.10008.5.1.4.1.1.4.2	MR Spectroscopy Storage
1.2.840.10008.5.1.4.1.1.66	Raw Data Storage
1.2.840.10008.5.1.4.1.1.66.1	Spatial Registration Storage
1.2.840.10008.5.1.4.1.1.66.2	Spatial Fiducials Storage

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.77.1.5.3	Stereometric Relationship Storage
1.2.840.10008.5.1.4.1.1.67	Real World Value Mapping Storage
1.2.840.10008.5.1.4.1.1.66.3	Deformable Spatial Registration
1.2.840.10008.5.1.4.1.1.66.4	Segmentation Storage
1.2.840.10008.5.1.4.1.1.77.1.5.2	Ophthalmic 16 bit Photography Image Storage
1.2.840.10008.5.1.4.1.1.77.1.5.1	Ophthalmic 8 bit Photography Image
1.2.840.10008.5.1.4.1.1.9.1.1	12-lead ECG Waveform Storage
1.2.840.10008.5.1.4.1.1.9.1.3	Ambulatory ECG Waveform Storage
1.2.840.10008.5.1.4.1.1.9.4.1	Basic Voice Audio Waveform Storage
1.2.840.10008.5.1.4.1.1.9.3.1	Cardiac Electrophysiology Waveform Storage
1.2.840.10008.5.1.4.1.1.9.1.2	General ECG Waveform Storage
1.2.840.10008.5.1.4.1.1.9.2.1	Hemodynamic Waveform Storage

MERGE_QR_SCU, in conjunction with Merge DICOM Toolkit, provides Standard Conformance to the following DICOM V3.0 Service Object Pair (SOP) Classes as a Query/Retrieve Service Class User (SCU), when providing the function of an archival data base for finding and moving diagnostic images.

Table C.2: Valid SCP Query/Retrieve SOP Classes for MERGE_QR_SCU AE

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.2.2.1	Study Root Query/Retrieve Information Model - Find
1.2.840.10008.5.1.4.1.2.2.2	Study Root Query/Retrieve Information Model - Move
1.2.840.10008.5.1.4.1.2.1.1	Patient Root Query/Retrieve Information Model - Find
1.2.840.10008.5.1.4.1.2.1.2	Patient Root Query/Retrieve Information Model - Move
1.2.840.10008.5.1.4.1.2.3.1	Patient/Study Only Root Query/Retrieve Information Model - Find
1.2.840.10008.5.1.4.1.2.3.2	Patient/Study Only Root Query/Retrieve Information Model - Move

2.1.1 Association establishment policies for MERGE_QR_SCU AE

2.1.1.1 General

The MERGE_QR_SCU application will initiate an association as a Query/Retrieve Service Class User requesting data for images and images themselves.

The MERGE_QR_SCU application will wait for an association as a Store.

2.1.1.2 Number of associations

The MERGE_QR_SCU AE allows a single association as a Query/Retrieve SCU and a single association as a Query/Retrieve SCP.

2.1.1.3 Asynchronous nature

The MERGE_QR_SCU AE does not support asynchronous communication (multiple outstanding transactions over a single association).

2.1.1.4 Implementation identifying information

The Implementation Class Unique Identifier (UID) for the MERGE_QR_SCU Application Entity is:

2.16.840.1.113669.2.1.1 (place your Implementation Class UID here).

The Implementation Version Name for the MERGE_QR_SCU Application Entity is:

MergeCOM3_222 (place your Implementation Version Name here).

2.1.2 Association initiation by real-world activity for MERGE_QR_SCU AE

The MERGE_QR_SCU application initiates an association for the appropriate Query/Retrieve Services Class that corresponds to the set of images requested to be transferred. The association is closed when all queries or moves have been sent to the remote DICOM network node. MERGE_QR_SCU is also able to abort the association through an operator requested abort or when an error occurs.

2.1.2.1 Real-world activity for Find and Move Execution operations of MERGE_QR_SCU

A MERGE_QR_SCU opens an association and to do C-MOVEs or C-FINDs. The association is closed after an error or when the initiator requests that it be closed.

2.1.2.1.1 Associated real-world activity for Find and Move Execution operations

Once the association has been established, the MERGE_QR_SCU waits for transmission of conformant Query/Retrieve Service messages. If a valid Find is received, then the local archival data base is searched and the requested information is returned to the requester. If a valid Move is received, then the local archival data base is searched for the requested images and they are sent to the requested remote network node.

2.1.2.1.2 Presentation context table for Find and Move Execution operations

The following table lists acceptable find and move presentation contexts.

Table C.3: Find and Move Execution Presentation Contexts of MERGE_QR_SCU

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
Patient Root Query/Retrieve Information Model - Find	1.2.840.10008.5.1.4.1.2.1.1	Explicit VR Little Endian Implicit VR Little Endian Explicit VR Big Endian	1.2.840.10008.1.2.1 1.2.840.10008.1.2 1.2.840.10008.1.2.2	SCU	None
Patient Root Query/Retrieve Information Model-Move	1.2.840.10008.5.1.4.1.2.1.2	Explicit VR Little Endian Implicit VR Little Endian Explicit VR Big Endian	1.2.840.10008.1.2.1 1.2.840.10008.1.2 1.2.840.10008.1.2.2	SCU	None
Study Root Query/Retrieve Information Model - Find	1.2.840.10008.5.1.4.1.2.2.1	Explicit VR Little Endian Implicit VR Little Endian Explicit VR Big Endian	1.2.840.10008.1.2.1 1.2.840.10008.1.2 1.2.840.10008.1.2.2	SCU	None
Study Root Query/Retrieve Information Model - Move	1.2.840.10008.5.1.4.1.2.2.2	Explicit VR Little Endian Implicit VR Little Endian Explicit VR Big Endian	1.2.840.10008.1.2.1 1.2.840.10008.1.2 1.2.840.10008.1.2.2	SCU	None
Patient/Study Only Query/Retrieve Information Model - Move	1.2.840.10008.5.1.4.1.2.3.1	Explicit VR Little Endian Implicit VR Little Endian Explicit VR Big Endian	1.2.840.10008.1.2.1 1.2.840.10008.1.2 1.2.840.10008.1.2.2	SCU	None

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
Patient/Study Only Query/Retrieve Information Model - Move	1.2.840.10008.5.1.4.1.2.3.2	Explicit VR Little Endian Implicit VR Little Endian Explicit VR Big Endian	1.2.840.10008.1.2.1 1.2.840.10008.1.2 1.2.840.10008.1.2.2	SCU	None

3. Profiles

3.1 Supported Communication Stacks

MERGE_QR_SCU, in conjunction with Merge DICOM Toolkit provides DICOM V3.0 TCP/IP Network Communication Support as defined in PS 3.8.

3.2 TCP/IP Stack

MERGE_QR_SCU uses the Merge DICOM Toolkit to communicate over the TCP/IP protocol stack on any physical interconnection media supporting the TCP/IP stack. The Toolkit inherits the TCP/IP stack from the host operating system upon which it executes. The Toolkit has been implemented on almost every major operating system platform.

3.2.1 Physical Media Support

The MERGE_QR_SCU AE is indifferent to the physical medium over which TCP/IP executes; it inherits this from the operating system on which it exists.

4. Extensions/specializations/privatizations

4.1 Standard extended/specialized/private SOPs

None supported.

4.2 Private Transfer Syntaxes

None supported.

5. Configuration

5.1 MERGE_QR_SCU Configuration Files

The MERGE_QR_SCU applications references four configuration files. The first, merge.ini, is found through the MERGE_INI environment variable. They are as follows:

merge.ini	Specifies the names of the other three configuration files and also contains message logging parameters.
mergecom.pro	Specifies run-time parameters for MERGE_QR_SCU applications.
mergecom.app	Defines applications on other network nodes, to which connections are possible.
mergecom.srv	Service and sequence definitions.

5.1.1 AE title/presentation address mapping

Presentation address mapping is configured in the mergecom.app file. This is where the Host Name, Port Number, and Application Title map an Application Entity (AE) Title to a Presentation Address in TCP/IP for the provider to which you wish to connect. Similarly, the Presentation Address of your SCP as a provider is specified by configuring the Listen Port in the mergecom.pro file, and specifying the AE title for your SCP within the application itself.

Note: The host name maps to an IP address as specified by your host table. Also, port 104 should always be used for standard connectivity; since this is the well-defined port for a DICOM server.

5.1.2 Configurable parameters

The mergecom.pro configuration file can be used to set or modify other lower-level communication parameters. This includes time-outs and other parameters. Some information about supported SOP classes is also stored here. **Most parameters in this file should NEVER be changed. Doing so could break DICOM conformance.** Before modifying any parameters, such as time-out, be sure to have a backup of the originally supplied mergecom.pro file. Also, before modifying other parameters, you should consider contacting Merge Healthcare for advice.

6. Support of extended character sets

Not supported.

Appendix D: Query/Retrieve SCP Conformance Statement

Introduction

This is a conformance statement for the Merge sample program (MERGE_QR_SCP) which supports DICOM Query/Retrieve and Storage Services as a Service Class Provider.

DICOM has been implemented by Merge Healthcare and is called Merge DICOM Toolkit. Therefore, Merge DICOM Toolkit and DICOM can and are used synonymously within this document.

1. Implementation Model

MERGE_QR_SCP with Merge DICOM Toolkit input and output is, very basically, an implementation of a DICOM Query/Retrieve Service Class Provider (SCP) which can receive DICOM queries and move requests from a DICOM Storage Service Class user (SCU). MERGE_QR_SCP responds to DICOM queries and move requests based on DICOM Storage Service Class messages it has received.

1.1 Application data flow diagram

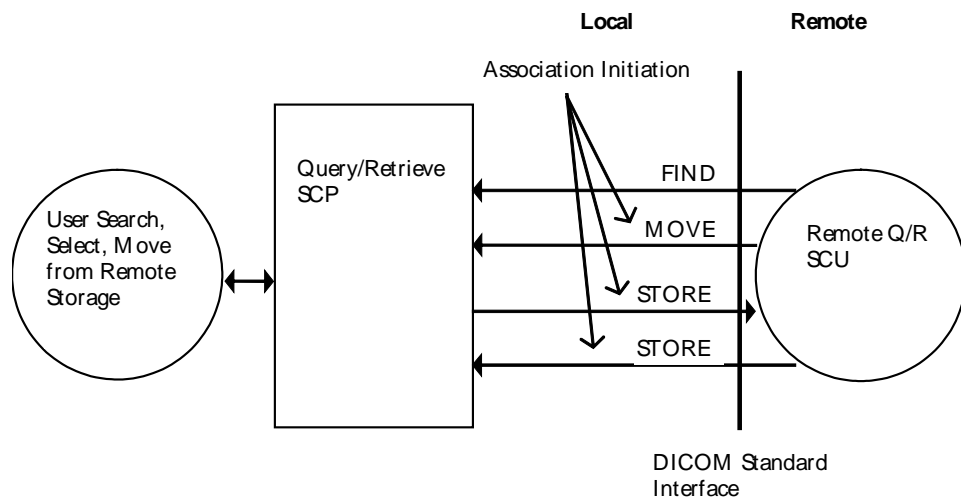


Figure D.1: Application Flow Diagram for Q/R SCP

1.2 Functional definition of Application Entity (AE)

All communications and image transfer with the remote application is accomplished utilizing the DICOM protocol over a network using the TCP/IP protocol stack. MERGE_QR_SCP will respond, if asked, with the Verification SOP Class UID as an SCP for one of its implemented SOP Classes.

The MERGE_QR_SCP waits for an association to accept at the TCP/IP port number that is configured at the time this application is initiated. When an association request is received with valid connection criteria, MERGE_QR_SCP

responds with a list of SOP Class UIDs that it will accept. It then waits for an Echo, Store, Find or Move request to determine what specific function it has to perform. If an Echo is received, then an appropriate Echo response is sent back to the initiator. If a Find request is received, then the archive is searched for the requested information and a Find response is returned with all the found information. If a Move request is received, it will initiate a transfer request of the requested set of images to the desired location. If a Store request is received, it will archive the Store request.

2. AE Specifications

2.1 AE specification for MERGE_QR_SCP

MERGE_QR_SCP, in conjunction with Merge DICOM Toolkit, provides Standard Conformance to the following DICOM V3.0 Service Object Pair (SOP) Class as a Verification Service Class User and Provider (SCU & SCP). As an SCP it sends out an Echo response after it receives an Echo request from a remote AE.

Table D.1: Valid SCU/SCP Verification SOP Class for MERGE_QR_SCP AE

SOP Class UID	SOP Class Name
1.2.840.10008.1.1	Verification SOP Class

MERGE_QR_SCP in conjunction with Merge DICOM Toolkit provides Standard Conformance to the following DICOM V3.0 Service Object Pair (SOP) Class as a Storage Service Class User and Provider (SCU & SCP).

Table D.2: Valid SCU/SCP STORE SOP Classes for MERGE_QR_SCP AE

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.1	Computed Radiography Image Storage
1.2.840.10008.5.1.4.1.1.2	CT Image Storage
1.2.840.10008.5.1.4.1.1.3	Ultrasound Multi-frame Image Storage (Retired)
1.2.840.10008.5.1.4.1.1.3.1	Ultrasound Multi-frame Image Storage
1.2.840.10008.5.1.4.1.1.4	MR Image Storage
1.2.840.10008.5.1.4.1.1.20	Nuclear Medicine Image Storage
1.2.840.10008.5.1.4.1.1.5	Nuclear Medicine Image Storage (Retired)
1.2.840.10008.5.1.4.1.1.6	Ultrasound Image Storage (Retired)
1.2.840.10008.5.1.4.1.1.6.1	Ultrasound Image Storage
1.2.840.10008.5.1.4.1.1.7	Secondary Capture Image Storage

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.7.2	Multi-frame Grayscale Byte Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.7.3	Multi-frame Grayscale Word Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.7.1	Multi-frame Single Bit Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.7.4	Multi-frame True Color Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.8	Standalone Overlay Storage
1.2.840.10008.5.1.4.1.1.9	Standalone Curve Storage
1.2.840.10008.5.1.4.1.1.10	Standalone Modality LUT Storage
1.2.840.10008.5.1.4.1.1.11	Standalone VOI LUT Storage
1.2.840.10008.5.1.4.1.1.12.1	Standard Xray Angio
1.2.840.10008.5.1.4.1.1.12.2	Standard Xray RF
1.2.840.10008.5.1.4.1.1.12.2	Standard Xray Angio Biplane
1.2.840.10008.5.1.4.1.1.128	Standard Positron Emission Tomography Image Storage
1.2.840.10008.5.1.4.1.1.129	Standard Positron Emission Tomography Curve Storage
1.2.840.10008.5.1.4.1.1.481.4	Standard RT Beams Treatment Record Storage
1.2.840.10008.5.1.4.1.1.481.6	Standard RT Brachy Treatment Record Storage
1.2.840.10008.5.1.4.1.1.481.1	Standard RT Image Storage
1.2.840.10008.5.1.4.1.1.481.2	Standard RT Dose Storage
1.2.840.10008.5.1.4.1.1.481.3	Standard RT Structure Set Storage
1.2.840.10008.5.1.4.1.1.481.9	RT Ion Beams Treatment Record Storage
1.2.840.10008.5.1.4.1.1.481.8	RT Ion Plan Storage
1.2.840.10008.5.1.4.1.1.481.5	Standard RT Plan Storage
1.2.840.10008.5.1.4.1.1.481.7	Standard RT Treatment Summary Record Storage
1.2.840.10008.5.1.1.30	Hardcopy Color Image Storage
1.2.840.10008.5.1.1.29	Hardcopy Grayscale Image Storage

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.1.27	Stored Print Storage
1.2.840.10008.5.1.4.1.1.1.1	Digital X-Ray Image Storage - For Presentation
1.2.840.10008.5.1.4.1.1.1.1.1	Digital X-Ray Image Storage - For Processing
1.2.840.10008.5.1.4.1.1.1.3	Digital Intra-oral X-Ray Image Storage - For Presentation
1.2.840.10008.5.1.4.1.1.1.3.1	Digital Intra-oral X-Ray Image Storage - For Processing
1.2.840.10008.5.1.4.1.1.1.2	Digital Mammography Image Storage - For Presentation
1.2.840.10008.5.1.4.1.1.1.2.1	Digital Mammography Image Storage - For Processing
1.2.840.10008.5.1.4.1.1.77.1.1	VL Endoscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.2	VL Microscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.4	VL Photographic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.3	VL Slide-Coordinates Microscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.1.1	Video Endoscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.2.1	Video Microscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.4.1	Video Photographic Image Storage
1.2.840.10008.5.1.4.1.1.88.65	Chest CAD SR Storage
1.2.840.10008.5.1.4.1.1.88.59	Key Object Selection Document
1.2.840.10008.5.1.4.1.1.88.50	Mammography CAD SR
1.2.840.10008.5.1.4.1.1.88.40	Procedure Log Storage
1.2.840.10008.5.1.4.1.1.88.11	Basic Text Structured Reporting
1.2.840.10008.5.1.4.1.1.88.33	Comprehensive Structured Reporting
1.2.840.10008.5.1.4.1.1.88.22	Enhanced Structured Reporting
1.2.840.10008.5.1.4.1.1.104.1	Encapsulated PDF Storage
1.2.840.10008.5.1.4.1.1.88.67	X-Ray Radiation Dose SR
1.2.840.10008.5.1.4.1.1.11.1	Grayscale Softcopy Presentation State Storage
1.2.840.10008.5.1.4.1.1.11.3	Pseudo-Color Softcopy Presentation State Storage

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.11.2	Color Softcopy Presentation State Storage
1.2.840.10008.5.1.4.1.1.11.4	Blending Softcopy Presentation State Storage
1.2.840.10008.5.1.4.1.1.2.1	Enhanced CT Image Storage
1.2.840.10008.5.1.4.1.1.4.1	Enhanced MR Image Storage
1.2.840.10008.5.1.4.1.1.12.1.1	Enhanced XA Image Storage
1.2.840.10008.5.1.4.1.1.12.2.1	Enhanced XRF Image Storage
1.2.840.10008.5.1.4.1.1.4.2	MR Spectroscopy Storage
1.2.840.10008.5.1.4.1.1.66	Raw Data Storage
1.2.840.10008.5.1.4.1.1.66.1	Spatial Registration Storage
1.2.840.10008.5.1.4.1.1.66.2	Spatial Fiducials Storage
1.2.840.10008.5.1.4.1.1.77.1.5.3	Stereometric Relationship Storage
1.2.840.10008.5.1.4.1.1.67	Real World Value Mapping Storage
1.2.840.10008.5.1.4.1.1.66.3	Deformable Spatial Registration
1.2.840.10008.5.1.4.1.1.66.4	Segmentation Storage
1.2.840.10008.5.1.4.1.1.77.1.5.2	Ophthalmic 16 bit Photography Image Storage
1.2.840.10008.5.1.4.1.1.77.1.5.1	Ophthalmic 8 bit Photography Image
1.2.840.10008.5.1.4.1.1.9.1.1	12-lead ECG Waveform Storage
1.2.840.10008.5.1.4.1.1.9.1.3	Ambulatory ECG Waveform Storage
1.2.840.10008.5.1.4.1.1.9.4.1	Basic Voice Audio Waveform Storage
1.2.840.10008.5.1.4.1.1.9.3.1	Cardiac Electrophysiology Waveform Storage
1.2.840.10008.5.1.4.1.1.9.1.2	General ECG Waveform Storage
1.2.840.10008.5.1.4.1.1.9.2.1	Hemodynamic Waveform Storage

MERGE_QR_SCP, in conjunction with Merge DICOM Toolkit, provides Standard Conformance to the following DICOM V3.0 Service Object Pair (SOP) Classes as a Query/Retrieve Service Class Provider (SCP), when providing the function of a permanent archival data base for finding and moving diagnostic images.

Table D.3: Valid SCP Query/Retrieve SOP Classes for MERGE_QR_SCP AE

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.2.2.1	Study Root Query/Retrieve Information Model - Find
1.2.840.10008.5.1.4.1.2.2.2	Study Root Query/Retrieve Information Model - Move
1.2.840.10008.5.1.4.1.2.1.1	Patient Root Query/Retrieve Information Model - Find
1.2.840.10008.5.1.4.1.2.1.2	Patient Root Query/Retrieve Information Model - Move
1.2.840.10008.5.1.4.1.2.3.1	Patient/Study Only Root Query/Retrieve Information Model - Find
1.2.840.10008.5.1.4.1.2.3.2	Patient/Study Only Root Query/Retrieve Information Model - Move

2.1.1 Association establishment policies for MERGE_QR_SCP AE

2.1.1.1 General

The MERGE_QR_SCP AE will initiate an association as a Storage Service Class User when images have been requested to be transferred to a remote DICOM network node with storage by either a local operator or by a remote operator via a Find/Move request.

The MERGE_QR_SCP AE application will wait for an association as an SCP the Query/Retrieve and Storage Service Classes. When a Find request is received, a search is done of the archival data base for the images with the requested attributes and list of found attributes is returned to the remote requester. When a Move request is received, the information identifying the set of images to be transferred is given internally to the SCU of Storage Service Class, which then transfers the image set to an SCP of Storage Service across the network. When a Store request is received, the images are stored locally and the information is stored about images so Find and Move requests can be processed for the images.

2.1.1.2 Number of associations

The MERGE_QR_SCP AE allows a variable number of associations both for association acceptance and association initiation, which are completely configurable prior to run-time. The normal default is five (5) for each, running simultaneously.

2.1.1.3 Asynchronous nature

The MERGE_QR_SCP AE does not support asynchronous communication (multiple outstanding transactions over a single association).

2.1.1.4 Implementation identifying information

The Implementation Class Unique Identifier (UID) for the MERGE_QR_SCP Application Entity is:

2.16.840.1.113669.2.1.1 (place your Implementation Class UID here).

The Implementation Version Name for the MERGE_QR_SCP Application Entity is:

MergeCOM3_370 (place your Implementation Version Name here).

2.1.2 Association initiation by real-world activity for MERGE_QR_SCP AE

The MERGE_QR_SCP client application initiates an association for the appropriate Storage Services Class that corresponds to the set of images requested to be transferred. The association is closed when all images have been sent to the remote DICOM network node. The client is also able to abort the association through an operator requested abort or when an error occurs.

2.1.2.1 Real-world activity for Echo Check operation of MERGE_QR_SCP

A MERGE_QR_SCP client application initiates associations for the echo service. The association is closed either when a correct response is received or when a time-out occurs.

2.1.2.1.1 Associated real-world activity for Echo Check operation

An echo is performed by a MERGE_QR_SCP client application by using the MC_Wait_For_Association or MC_Read_Message.

2.1.2.1.2 Proposed presentation contexts for Echo Check operation

MERGE_QR_SCP supports the Verification SOP Class fully as specified in the DICOM Standard.

The presentation context proposed by a MERGE_QR_SCP client for the Echo Check operation is specified in the following table.

Table D.4: Echo Check Presentation Contexts of MERGE_QR_SCP

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended
Name	UID	Name List	UID List		Negotiation
Verification Service Class	1.2.840.10008.1.1	DICOM Implicit VR Little Endian	1.2.840.10008.1.2	SCU	None

2.1.2.1.2.1 SOP specific conformance for Verification SOP Class

No known SOP specific conformance issues.

2.1.3.1 Real-world activity for Find and Move Execution operations of MERGE_QR_SCP

A MERGE_QR_SCP application waits for an association and offers to do C-MOVEs or C-FINDs. The association is closed after an error or when the initiator requests that it be closed.

2.1.3.1.1 Associated real-world activity for Find and Move Execution operations

Once the association has been established, the MERGE_QR_SCP waits for transmission of conferment Query/Retrieve Service messages. If a valid Find is received, then the local archival data base is searched and the requested information is returned to the requester. If a valid Move is received, then the local archival data base is searched for the requested images and they are sent to the requested remote network node.

2.1.3.1.2 Presentation context table for Find and Move Execution operations

The presentation contexts that are proposed by MERGE_QR_SCP AE for the Find and Move Execution operations are specified in the following table.

Table D.5: Find and Move Execution Presentation Contexts of MERGE_QR_SCP

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended
Name	UID	Name List	UID List		Negotiation
Patient Root Query/Retrieve Information Model - Find	1.2.840.10008.5.1.4.1.2.1.1	Explicit VR Little Endian Implicit VR Little Endian Explicit VR Big Endian	1.2.840.10008.1.2.1 1.2.840.10008.1.2 1.2.840.10008.1.2.2	SCP	None
Patient Root Query/Retrieve Information Model-Move	1.2.840.10008.5.1.4.1.2.1.2	Explicit VR Little Endian Implicit VR Little Endian Explicit VR Big Endian	1.2.840.10008.1.2.1 1.2.840.10008.1.2 1.2.840.10008.1.2.2	SCP	None
Study Root Query/Retrieve Information Model - Find	1.2.840.10008.5.1.4.1.2.2.1	Explicit VR Little Endian Implicit VR Little Endian Explicit VR Big Endian	1.2.840.10008.1.2.1 1.2.840.10008.1.2 1.2.840.10008.1.2.2	SCP	None

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended
Name	UID	Name List	UID List		Negotiation
Study Root Query/Retrieve Information Model - Move	1.2.840.10008.5.1.4.1.2.2.2	Explicit VR Little Endian Implicit VR Little Endian Explicit VR Big Endian	1.2.840.10008.1.2.1 1.2.840.10008.1.2 1.2.840.10008.1.2.2	SCP	None
Patient/Study Only Query/Retrieve Information Model - Move	1.2.840.10008.5.1.4.1.2.3.1	Explicit VR Little Endian Implicit VR Little Endian Explicit VR Big Endian	1.2.840.10008.1.2.1 1.2.840.10008.1.2 1.2.840.10008.1.2.2	SCP	None
Patient/Study Only Query/Retrieve Information Model - Move	1.2.840.10008.5.1.4.1.2.3.2	Explicit VR Little Endian Implicit VR Little Endian Explicit VR Big Endian	1.2.840.10008.1.2.1 1.2.840.10008.1.2 1.2.840.10008.1.2.2	SCP	None

2.1.3.1.3 Presentation context acceptance criterion for Find and Move Execution operations

Not applicable since only a single presentation context for each Query/Retrieve Service Class is supported.

2.1.3.1.4 Transfer syntax selection policies for Find and Move Execution operations

When executing on a Little Endian machine, transfer syntaxes are accepted in the following order:

Transfer Syntax UID	Transfer Syntax Name
1.2.840.10008.1.2	Explicit Little Endian Syntax
1.2.840.10008.1.2.1	Implicit Little Endian Syntax
1.2.840.10008.1.2.2	Explicit Big Endian Syntax

When executing on a Big Endian machine, transfer syntaxes are accepted in the following order:

Transfer Syntax UID	Transfer Syntax Name
1.2.840.10008.1.2.2	Explicit Big Endian Syntax
1.2.840.10008.1.2.1	Explicit Little Endian Syntax
1.2.840.10008.1.2	Implicit Little Endian Syntax

Note: This acceptance criteria can be overridden by the use of a “Transfer Syntax List” in the mergecom.app configuration file.

2.1.3.2 Real-world activity for Receive Image operations

MERGE_QR_SCP waits for an association and offers to do the Image Storage service. The association is closed after an error or when the initiator requests that it be closed.

2.1.3.2.1 Associated real-world activity for Receive Image operations

Once the association has been established, the MERGE_QR_SCP waits for transmission of conformant Storage Service messages.

2.1.3.2.2 Presentation context table for Receive Image operations

The presentation contexts that are proposed by MERGE_STORE_SCP AE for the Receive Image Presentation operations are specified in the following table.

Table D.6: Receive Image Presentation Contexts of MERGE_STORE_SCP

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
All Services in Table D.2	All Services in Table D.2	Explicit VR Little Endian Implicit VR Little Endian Explicit VR Big Endian	1.2.840.10008.1.2.1 1.2.840.10008.1.2 1.2.840.10008.1.2.2	SCP	None

2.1.2.2.2.2 Presentation context acceptance criterion for Receive Image operations

Not applicable since only a single presentation context for each Storage Service Class is supported.

2.1.2.2.3 Transfer syntax selection policies for Receive Image operations

When executing on a Little Endian machine, transfer syntaxes are accepted in the following order:

Transfer Syntax UID	Transfer Syntax Name
1.2.840.10008.1.2	Explicit Little Endian Syntax
1.2.840.10008.1.2.1	Implicit Little Endian Syntax
1.2.840.10008.1.2.2	Explicit Big Endian Syntax

When executing on a Big Endian machine, transfer syntaxes are accepted in the following order:

Transfer Syntax UID	Transfer Syntax Name
1.2.840.10008.1.2.2	Explicit Big Endian Syntax
1.2.840.10008.1.2.1	Explicit Little Endian Syntax
1.2.840.10008.1.2	Implicit Little Endian Syntax

Note: This acceptance criteria can be overridden by the use of a “Transfer Syntax List” in the mergecom.app configuration file.

3. Profiles

3.1 Supported Communication Stacks

MERGE_QR_SCP, in conjunction with Merge DICOM Toolkit provides DICOM V3.0 TCP/IP Network Communication Support as defined in PS 3.8.

3.2 TCP/IP Stack

MERGE_QR_SCP uses the Merge DICOM Toolkit to communicate over the TCP/IP protocol stack on any physical interconnection media supporting the TCP/IP stack. The Toolkit inherits the TCP/IP stack from the host operating system upon which it executes. The Toolkit has been implemented on almost every major operating system platform.

3.2.1 Physical Media Support

The MERGE_QR_SCP AE is indifferent to the physical medium over which TCP/IP executes; it inherits this from the operating system on which it exists.

4. Extensions/specializations/privatizations

4.1 Standard extended/specialized/private SOPs

None supported.

4.2 Private Transfer Syntaxes

None supported.

5. Configuration

5.1 MERGE_QR_SCP Configuration Files

The MERGE_QR_SCP applications references four configuration files. The first, merge.ini, is found through the MERGE_INI environment variable. They are as follows:

merge.ini	Specifies the names of the other three configuration files and also contains message logging parameters.
mergecom.pro	Specifies run-time parameters for the MERGE_QR_SCP applications.
mergecom.app	Defines applications on other network nodes, to which connections are possible.
mergecom.srv	Service and sequence definitions.

5.1.1 AE title/presentation address mapping

Presentation address mapping is configured in the mergecom.app file. This is where the Host Name, Port Number, and Application Title map an Application Entity (AE) Title to a Presentation Address in TCP/IP for the provider to which you wish to connect. Similarly, the Presentation Address of your SCP as a provider is specified by configuring the Listen Port in the mergecom.pro file, and specifying the AE title for your SCP within the application itself.

Note: The host name maps to an IP address as specified by your host table. Also, port 104 should always be used for standard connectivity; since this is the well-defined port for a DICOM server.

5.1.2 Configurable parameters

The mergecom.pro configuration file can be used to set or modify other lower-level communication parameters. This includes time-outs and other parameters. Some information about supported SOP classes is also stored here. **Most parameters in this file should NEVER be changed. Doing so could break DICOM conformance.** Before modifying any parameters, such as time-out, be sure to have a backup of the originally supplied mergecom.pro file. Also, before modifying other parameters, you should consider contacting Merge Healthcare for advice.

6. Support of extended character sets

Not supported.

Appendix E: Print SCU Conformance Statement

0. Introduction

This document provides the DICOM conformance statement for the Merge Healthcare sample print application. This application implements the DICOM print management service class to compile information necessary to display print related data on screen. Standard DICOM protocols are used to manage Film Sessions and Print Job Queues. This sample application does not fully utilize the functionality of the Merge DICOM Toolkit. Its purpose was to provide a basic example of a DICOM conformant SCU.

DICOM has been implemented by Merge Healthcare and is called Merge DICOM Toolkit. Therefore, Merge DICOM Toolkit and DICOM can and are used synonymously within this document.

1. Implementation model

MERGE_PRINT_SCU with Merge DICOM Toolkit input and output is, very basically, an implementation of a DICOM Basic Print user (SCU) which can send DICOM images to a DICOM Basic Print provider (SCP).

1.1 Application data flow diagram

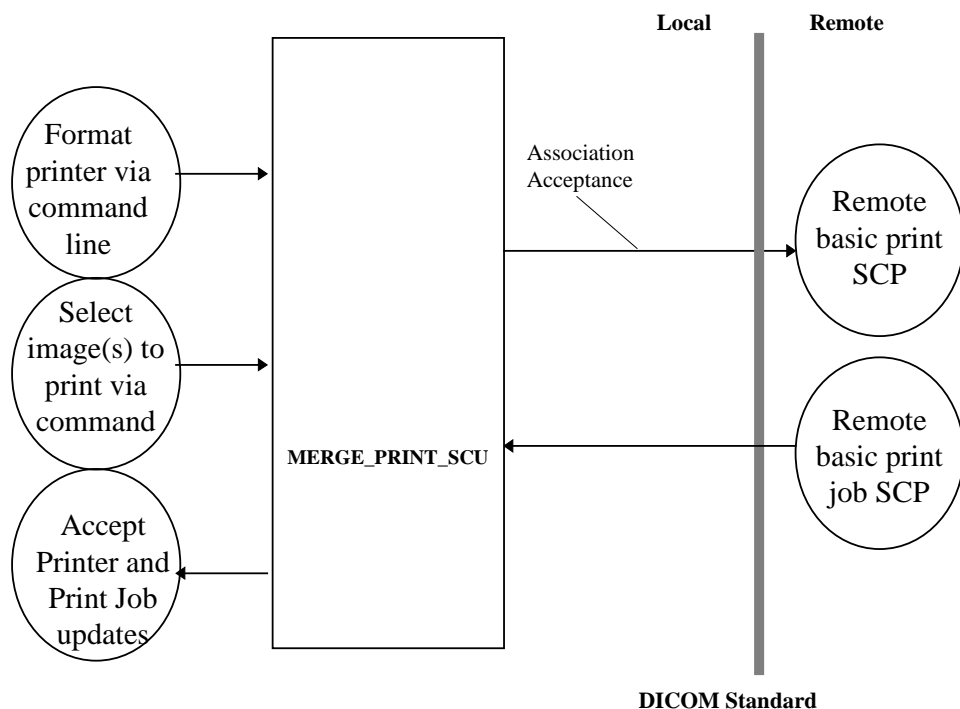


Figure E.1: MERGE_PRINT_SCU application data flow diagram

1.2 Functional definition of Application Entity (AE)

All communications with the remote application is accomplished utilizing the DICOM protocol. Using command line options, the SCU determines the printer format and the images to print. At any time, the SCU can receive printer and print job updates and will display them upon the screen. It establishes an association with a user selected remote AE just prior to sending a Print request to that AE.

1.3 Sequencing of real-world activities

Not applicable.

2. AE specifications

2.1 AE Specification for MERGE_PRINT_SCU

MERGE_PRINT_SCU, in conjunction with Merge DICOM Toolkit, provides Standard Conformance to the DICOM Basic Grayscale Print Management Meta SOP class and Print Job SOP class as a DICOM Basic Print User (SCU).

Table E.1: Valid SOP Classes for MERGE_PRINT_SCU

SOP Class Name	SOP Class UID
Basic Grayscale Print Management (META)	1.2.840.10008.5.1.1.9
Basic Film Session	1.2.840.10008.5.1.1.1
Basic Film Box	1.2.840.10008.5.1.1.2
Basic Grayscale Image Box	1.2.840.10008.5.1.1.4
Printer	1.2.840.10008.5.1.1.16
Print Job	1.2.840.10008.5.1.1.14

2.1.1 Association establishment policies

2.1.1.1 General

The MERGE_PRINT_SCU AE will initiate an association as an SCU of Print Services when a local operator requests to print images over the network to a remote DICOM Basic Print provider. The maximum PDU size is configurable from a minimum of 4,096 bytes.

2.1.1.2 Number of Associations

The MERGE_PRINT_SCU AE only opens 1 association at a time. The operator may select that the sequence of images be printed multiple times; in which case, the MERGE_PRINT_SCU AE will open multiple non-simultaneous associations with the remote AE.

2.1.1.3 Asynchronous Nature

The MERGE_PRINT_SCU AE does not support asynchronous communication (multiple outstanding transactions over a single association).

2.1.1.4 Implementation Identifying Information

The Implementation Class Unique Identifier (UID) for the MERGE_PRINT_SCU Application Entity is:

2.16.840.1.113669.2.2.1 (place your Implementation Class UID here).

The Implementation Version Name for the MERGE_PRINT_SCU Application Entity is:

MergeCOM_222 (place your Implementation Version Name here).

2.1.2 Association initiation by real-world activity

The MERGE_PRINT_SCU AE initiates an association for the appropriate Print Services Class that corresponds to the set of images requested to be printed. The association is closed when all images have been printed and all print jobs have completed. The MERGE_PRINT_SCU is able to abort an association when a time-out occurs. The following table describes the time-outs and their values. The client is also able to abort the association when an error occurs.

Table E.2: Time Out Values for MERGE_PRINT_SCU

Variable	Seconds	Description
ARTIM_TIMEOUT	30	The number of seconds to use as a time-out waiting for association request or waiting for the peer to shut down an association.
ASSOC_REPLY_TIMEOUT	15	The number of seconds to wait for reply to associate request.
RELEASE_TIMEOUT	15	The number of seconds to wait for reply to associate release.
WRITE_TIMEOUT	15	The number of seconds to wait for a network write to be accepted.

2.1.2.1 Real-World Activity for Print Image Operations

The MERGE_PRINT_SCU AE initiates associations for the printing of images to a Basic Print SCP. The application will accept any preformatted images..

2.1.2.1.1 Associated Real-World Activity for Print Image Operations

Once the Print Image association has been established, MERGE_PRINT_SCU sends a Basic Film Session, N_CREATE message to the Basic Print SCP. This is followed by a Basic Film Box, N_CREATE message. The MERGE_PRINT_SCU then sends a Basic Grayscale Image Box, N_SET message. Finally, a N_ACTION message is sent to instruct the Basic Print SCP to print either at the Basic Film Session or at the Basic Film Box level.

2.1.2.1.2 Proposed presentation contexts for Print Image operations

The presentation contexts that are proposed by MERGE_PRINT_SCU AE for the Print Image operation are specified in the following table. All these SOP classes conform to the standard Print Services as specified in the DICOM Standard.

Table E.3: Print Image Presentation Contexts of MERGE_PRINT_SCU

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
Basic Grayscale Print Management (META)	1.2.840.10008.5.1.1.9	Explicit VR Little Endian	1.2.840.10008.1.2.1	SCU	None
		Implicit VR Little Endian	1.2.840.10008.1.2		
		Explicit VR Big Endian	1.2.840.10008.1.2.2		
Print Job	1.2.840.10008.5.1.1.14	Explicit VR Little Endian	1.2.840.10008.1.2.1	SCU	None
		Implicit VR Little Endian	1.2.840.10008.1.2		
		Explicit VR Big Endian	1.2.840.10008.1.2.2		

2.1.2.1.3 SOP Specific Conformance

Attribute values for SOP classes proposed by MERGE_PRINT_SCU are specified in the following table. At this time, it should be noted that this SCU is just a sample application. It reflects only a small portion of the functionality of the toolkit.

Table E.4: Attribute values for supported SOP Classes

SOP Class Name	Command	Attribute Name	Valid Range	Default Value
Basic Film Session	N_CREATE	Number of Copies	1-99	1
		Print Priority	HIGH, MEDIUM, LOW	LOW
		Medium Type	PAPER, CLEAR FILM, BLUE FILM	None
		Film Destination	N/A	None
		Film Session Label	N/A	None
		Memory Allocation	N/A	None
Basic Film Session	N_ACTION	Referenced Print Job Sequence		None
Basic Film Box	N_CREATE	Image Display Format	STANDARD\1,1 STANDARD\1,2 STANDARD\2,2 STANDARD\2,3 STANDARD\3,3 STANDARD\3,4 STANDARD\3,5 STANDARD\4,4 STANDARD\4,5 STANDARD\4,6	Mandatory, no default
		Film Orientation	PORTRAIT, LANDSCAPE	None
		Film Size ID	8INX10IN, 10INX14IN	None
		Magnification Type	N/A	None
		Max Density	N/A	None
		Configuration Information	This information is printer specific	None
		Smoothing Type	N/A	None
		Border Density	N/A	None
		Empty Image Density	N/A	None
		Min Density	N/A	None

SOP Class Name	Command	Attribute Name	Valid Range	Default Value
		Trim	N/A	None
Basic Film Box	N_ACTION	Referenced Print Job Sequence		None
Basic Grayscale Image Box	N_SET	Image Position	1-24	Mandatory, no default
		Samples Per Pixel	1	None
		Photometric Interpretation	MONOCHROME 1, MONOCHROME 2	None
		Rows	any integer	None
		Columns	any integer	None
		Pixel Aspect Ratio	1/1	None
		Bits Allocated	8	None
		Bit Stored	8	None
		High Bit	7	None
		Pixel Representation	0000	None
		Polarity	N/A	None
		Magnification Type	N/A	None
		Smoothing Type	N/A	None
		Requested Image Size	N/A	None
Printer	N_GET/ N_EVENT_REPORT	Printer Status	*	None
		Printer Status Info	*	None
		Printer Name	*	None
		Manufacturer	*	None
		Manufacturer Model Name	*	None
		Software Version	*	None

SOP Class Name	Command	Attribute Name	Valid Range	Default Value
Print Job	N_EVENT_REPORT	Execution Status	*	None
		Execution Status Info	*	None
		Print Priority	*	None
		Creation Date	*	None
		Creation Time	*	None
		Printer Name	*	None
		Originator	*	None

* The MERGE_PRINT_SCU will display any information returned for Printer or Print Job messages

3. Profiles

3.1 Supported Communication Stacks

MERGE_PRINT_SCU, in conjunction with Merge DICOM Toolkit provides DICOM V3.0 TCP/IP Network Communication Support as defined in PS 3.8.

3.2 TCP/IP Stack

MERGE_PRINT_SCU uses the Merge DICOM Toolkit to communicate over the TCP/IP protocol stack on any physical interconnection media supporting the TCP/IP stack. The Toolkit inherits the TCP/IP stack from the host operating system upon which it executes. The Toolkit has been implemented on almost every major operating system platform.

3.2.1 Physical Media Support

The MERGE_PRINT_SCU AE is indifferent to the physical medium over which TCP/IP executes; it inherits this from the operating system on which it exists.

4. Extensions/specializations/privatizations

4.1 Standard extended/specialized/private SOPs

None supported.

4.2 Private Transfer Syntaxes

None supported.

5. Configuration

5.1 MERGE_PRINT_SCU Configuration files

The MERGE_PRINT_SCU application references four configuration files. The first, “merge.ini,” is found through the MERGE_INI environment variable. They are as follows:

merge.ini	Specifies the names of the other three configuration files and also contains message logging parameters.
mergecom.pro	Specifies run-time parameters for the MERGE_PRINT_SCU application.
mergecom.app	Defines applications on other network nodes, to which connections are possible.
mergecom.srv	Service and sequence definitions.

5.1.1 AE title/presentation address mapping

Presentation address mapping is configured in the “mergecom.app” file. This is where the Host Name, Port Number, and Application Title map an Application Entity (AE) Title to a Presentation Address in TCP/IP for the provider to which you wish to connect.

Note: The host name maps to an IP address as specified by your host table.

5.1.2 Configurable parameters

The “mergecom.pro” configuration file can be used to set or modify other lower-level communication parameters. This includes time-outs and other parameters. Some information about supported SOP classes is also printed here. **Most parameters in this file should NEVER be changed. Doing so could break DICOM conformance.** Before modifying any parameters, such as time-out, be sure to have a backup of the originally supplied mergecom.pro file. Also, before modifying other parameters, you should consider contacting Merge Healthcare for advice.

6. Support of extended character sets

Not supported.

Appendix F: Print SCP Conformance Statement

0. Introduction

This is a conformance statement for the Merge sample program (MERGE_PRINT_SCP) which supports DICOM Print Services as a Service Class Provider (SCP). This application implements the DICOM print management service class to compile information necessary to display print related data on screen. This sample application does not fully utilize the functionality of the Merge DICOM Toolkit. Its purpose was to provide a basic example of a DICOM conformant SCU.

DICOM has been implemented by Merge Healthcare and is called Merge DICOM Toolkit. Therefore, Merge DICOM Toolkit and DICOM are used synonymously within this document.

1. Implementation model

MERGE_PRINT_SCP with Merge DICOM Toolkit input and output is, very basically, an implementation of a DICOM Basic Print provider (SCP) which can receive DICOM images from a DICOM Basic Print user (SCU).

1.1 Application data flow diagram

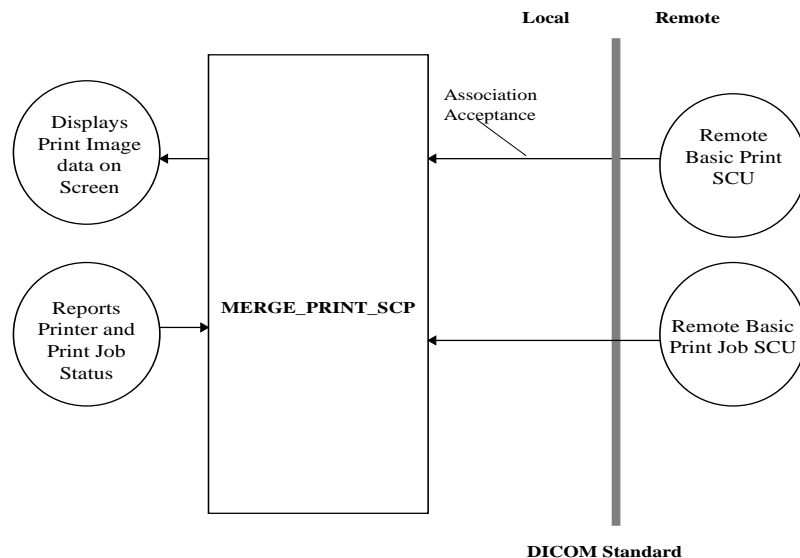


Figure F.1: MERGE_PRINT_SCP application data flow diagram

1.2 Functional definition of Application Entity (AE)

All communications and image printing with the remote application is accomplished utilizing the DICOM protocol. MERGE_PRINT_SCP will respond, if asked, with the Verification SOP Class UID as an SCP for one of its implemented SOP classes. MERGE_PRINT_SCP waits for an association to accept at the TCP/IP port number that is configured at the time this application is initiated. When an association request is received with valid connections criteria, MERGE_PRINT_SCP responds with a list of SOP class UIDs that it will accept. It then waits for a Print request. If a Print is received, then all incoming image data that is conformant to the association is printed on screen depending on the command-line arguments used when the application was initiated. MERGE_PRINT_SCP will also return Printer and Print Job information upon request from the SCU.

1.3 Sequencing of real-world activities

Not applicable.

2. AE specifications

2.1 AE Specification for MERGE_PRINT_SCP

2.1.1 Association Establishment Policies for MERGE_PRINT_SCP AE

Not applicable.

2.1.1.1 General

The MERGE_PRINT_SCP application will wait for an association as an SCP of Print Services. When a Print request is received, the corresponding image data is printed to the screen. The maximum PDU size is configurable from a minimum of 4,096 bytes.

2.1.1.2 Number of associations

Under multi-tasking operating systems, the MERGE_PRINT_SCP AE allows multiple simultaneous Print associations. The default number of associations is 5.

2.1.1.3 Asynchronous nature

The MERGE_PRINT_SCP AE does not support asynchronous communication (multiple outstanding transactions over a single association).

2.1.1.4 Implementation Identifying information

The Implementation Class Unique Identifier (UID) for the MERGE_PRINT_SCP Application Entity is:

2.16.840.1.113669.2.1.1 (place your Implementation Class UID here).

The Implementation Version Name for the MERGE_PRINT_SCP Application Entity is:

MergeCOM3_222 (place your Implementation Version Name here).

2.1.2 Association acceptance Policy for MERGE_PRINT_SCP

2.1.2.1 Real-world activity: Echo Check

2.1.2.1.1 Associated real-world activity for Echo Check operation

MERGE_PRINT_SCP with Merge DICOM Toolkit, provides Standard Conformance to the following DICOM V3.0 Service Object Pair (SOP) Class as a Verification Service Class Provider (SCP). As an SCP it sends out an Echo response after it receives an Echo request from a remote AE.

Table F.1: Valid SCP Verification SOP Class for MERGE_PRINT_SCP AE

SOP Class Name	SOP Class UID
Verification SOP Class	1.2.840.10008.1.1

2.1.2.1.2 Proposed presentation contexts for Echo Check operation

MERGE_PRINT_SCP supports the Verification SOP Class fully as specified in the DICOM Standard.

The presentation context proposed by a MERGE_PRINT_SCP client for the Echo Check operation are specified in the following table.

Table F.2: Echo Check Presentation Contexts of MERGE_PRINT_SCP

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
Verification Service Class	1.2.840.10008.1.1	Implicit VR Little Endian	1.2.840.10008.1.2	SCU	None

2.1.2.2 Real-world activity: Print image

MERGE_PRINT_SCU, in conjunction with Merge DICOM Toolkit, provides Standard Conformance to the DICOM Basic Grayscale Print Management Meta SOP class and Print Job SOP class as a DICOM Basic Print Provider (SCP).

Table F.3: Valid SCP Print SOP Classes for MERGE_PRINT_SCP

SOP Class Name	SOP Class UID
Basic Grayscale Print Management (META)	1.2.840.10008.5.1.1.9
Basic Film Session	1.2.840.10008.5.1.1.1
Basic Film Box	1.2.840.10008.5.1.1.2
Basic Grayscale Image Box	1.2.840.10008.5.1.1.4
Printer	1.2.840.10008.5.1.1.16
Print Job	1.2.840.10008.5.1.1.14

The MERGE_PRINT_SCP application accepts an association for the appropriate Basic Print Service Classes that corresponds to the set of images requested to be printed. The association is closed by the DICOM Basic Print user which initiated the association.

The MERGE_PRINT_SCP is able to abort an association when a time-out occurs. The following table outlines the time-outs and their values. The client is also able to abort the association when an error occurs.

Table F.4: Time-Out Values for MERGE_PRINT_SCP

Variable	Seconds	Description
ARTIM_TIMEOUT	30	The number of seconds to use as a time-out waiting for association request or waiting for the peer to shut down an association.
ASSOC_REPLY_TIMEOUT	15	The number of seconds to wait for reply to associate request.
RELEASE_TIMEOUT	15	The number of seconds to wait for reply to associate release.
WRITE_TIMEOUT	15	The number of seconds to wait for a network write to be accepted.

2.1.2.2.1 Associated real-world activity for Print Image operations

Once the association has been established, the MERGE_PRINT_SCP waits for transmission of conformant Print Service messages.

2.1.2.2.2 Proposed presentation contexts for Print Image operations

The presentation contexts that are proposed by MERGE_PRINT_SCP AE for the Print Image operation are specified in the following table.

All these SOP classes conform to the standard Print Services as specified in the DICOM Standard.

Table F.5: Print Image Presentation Contexts of MERGE_PRINT_SCP

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
Basic Grayscale Print Management (META)	1.2.840.10008.5.1.1.9	Explicit VR Little Endian	1.2.840.10008.1.2.1	SCU	None
		Implicit VR Little Endian	1.2.840.10008.1.2		
		Explicit VR Big Endian	1.2.840.10008.1.2.2		
Print Job	1.2.840.10008.5.1.1.14	Explicit VR Little Endian	1.2.840.10008.1.2.1	SCU	None
		Implicit VR Little Endian	1.2.840.10008.1.2		
		Explicit VR Big Endian	1.2.840.10008.1.2.2		

2.1.2.2.3 SOP specific conformance for print SOP Classes

Attribute values for SOP classes proposed by MERGE_PRINT_SCP are specified in the following table. At this time, it should be noted that this SCP is just a sample application that simulates a printer. It reflects only a small portion of the functionality of the toolkit.

Table F.6: Print Image Presentation Contexts of MERGE_PRINT_SCP

SOP Class Name	Command	Attribute Name	Valid Range	Default Value
Basic Film Session	N_CREATE/ N_SET	Number of Copies	1-99	1
		Print Priority	HIGH, MEDIUM, LOW	LOW
		Medium Type	PAPER, CLEAR FILM, BLUE FILM	DEFAULT
		Film Destination	N/A	DEFAULT
		Film Session Label	N/A	DEFAULT
		Memory Allocation	N/A	N/A

SOP Class Name	Command	Attribute Name	Valid Range	Default Value
Basic Film Session	N_ACTION	Referenced Print Job Sequence		None
Basic Film Box	N_CREATE	Image Display Format	STANDARD\1,1 STANDARD\1,2 STANDARD\2,2 STANDARD\2,3 STANDARD\3,3 STANDARD\3,4 STANDARD\3,5 STANDARD\4,4 STANDARD\4,5 STANDARD\4,6	Mandatory, no default
		Film Orientation	PORTRAIT, LANDSCAPE	*
		Film Size ID	8INX10IN, 10INX14IN	*
		Magnification Type	N/A	DEFAULT
		Max Density	N/A	DEFAULT
		Configuration Information	This information is printer specific	DEFAULT
		Smoothing Type	N/A	DEFAULT
		Border Density	N/A	DEFAULT
		Empty Image Density	N/A	DEFAULT
		Min Density	N/A	DEFAULT
		Trim	N/A	DEFAULT
Basic Film Box	N_SET	Film Size ID	8INX10IN, 10INX14IN	*
		Magnification Type	N/A	DEFAULT
		Max Density	N/A	DEFAULT
		Configuration Information	This information is printer specific	DEFAULT
		Smoothing Type	N/A	DEFAULT

SOP Class Name	Command	Attribute Name	Valid Range	Default Value
		Border Density	N/A	DEFAULT
		Empty Image Density	N/A	DEFAULT
		Min Density	N/A	DEFAULT
		Trim	N/A	DEFAULT
Basic Film Box	N_ACTION	Referenced Print Job Sequence		None
Basic Grayscale Image Box	N_SET	Image Position	1 -24	Mandatory, no default
		Samples Per Pixel	1	1
		Photometric Interpretation	MONOCHROME1, MONOCHROME2	DEFAULT
		Rows	any integer	any integer
		Columns	any integer	any integer
		Pixel Aspect Ratio	1	DEFAULT
		Bits Allocated	8	8
		Bit Stored	8	8
		High Bit	7	7
		Pixel Representation		0000
		Polarity	N/A	DEFAULT
		Magnification Type	N/A	DEFAULT
		Smoothing Type	N/A	DEFAULT
		Requested Image Size	N/A	DEFAULT
Printer	N_GET/ N_EVENT_REPORT	Printer Status	N/A	NORMAL
		Printer Status Info	N/A	None

SOP Class Name	Command	Attribute Name	Valid Range	Default Value
		Printer Name	MERGE PRINTER	None
		Manufacturer	MERGE HEALTHCARE	None
		Manufacturer Model Name	MERGE DEMO PRINTER	None
		Software Version	VERSION BETA 1.0	None
Print Job	N_GET/ N_EVENT_REPORT	Execution Status	1, 2, 3, 4	DONE
		Exec. Status Info	N/A	None
		Print Priority	N/A	None
		Creation Date	N/A	None
		Creation Time	N/A	None
		Printer Name	N/A	MERGE PRINTER
		Originator	N/A	None

* Some default values are defined by the printer configuration information (e.g. Film Orientation); therefore, the MERGE_PRINT_SCP will not defined some default values.

2.1.2.2.3 Presentation context acceptance criterion for Print Image operations

Not applicable since only a single presentation context for each Storage Service Class is supported.

2.1.2.2.4 Transfer syntax selection policies for Print Image operations

When executing on a Little Endian machine, transfer syntaxes are accepted in the following order:

Transfer Syntax UID	Transfer Syntax Name
1.2.840.10008.1.2	Explicit Little Endian Syntax
1.2.840.10008.1.2.1	Implicit Little Endian Syntax
1.2.840.10008.1.2.2	Explicit Big Endian Syntax

When executing on a Big Endian machine, transfer syntaxes are accepted in the following order:

Transfer Syntax UID	Transfer Syntax Name
1.2.840.10008.1.2.2	Explicit Big Endian Syntax
1.2.840.10008.1.2.1	Explicit Little Endian Syntax
1.2.840.10008.1.2	Implicit Little Endian Syntax

Note: This acceptance criteria can be overridden by the use of a “Transfer Syntax List” in the mergecom.app configuration file.

3. Profiles

3.1 Supported Communication Stacks

MERGE_PRINT_SCP, in conjunction with Merge DICOM Toolkit provides DICOM V3.0 TCP/IP Network Communication Support as defined in PS 3.8.

3.2 TCP/IP Stack

MERGE_PRINT_SCP uses the Merge DICOM Toolkit to communicate over the TCP/IP protocol stack on any physical interconnection media supporting the TCP/IP stack. The Toolkit inherits the TCP/IP stack from the host operating system upon which it executes. The Toolkit has been implemented on almost every major operating system platform.

3.2.1 Physical Media Support

The MERGE_PRINT_SCP AE is indifferent to the physical medium over which TCP/IP executes; it inherits this from the operating system on which it exists.

4. Extensions/specializations/privatizations

4.1 Standard extended/specialized/private SOPs

None supported.

4.2 Private Transfer Syntaxes

None supported.

5. Configuration

5.1 MERGE_PRINT_SCP Configuration Files

The MERGE_PRINT_SCP application references four configuration files. The first, “merge.ini,” is found through the MERGE_INI environment variable. They are as follows:

merge.ini	Specifies the names of the other three configuration files and also contains message logging parameters.
-----------	--

mergecom.pro	Specifies run-time parameters for the MERGE_PRINT_SCP application.
mergecom.app	Defines applications on other network nodes, to which connections are possible.
mergecom.srv	Service and sequence definitions.

5.1.1 AE title/presentation address mapping

Presentation address mapping is configured in the mergecom.app file. The Presentation Address of an SCP application as a provider is specified by configuring the Listen Port in the “mergecom.pro” file, and specifying the AE title for the SCP within the application itself.

5.1.2 Configurable parameters

The “mergecom.pro” configuration file can be used to set or modify other lower-level communication parameters. This includes time-outs and other parameters. Some information about supported SOP classes is also printed here. **Most parameters in this file should NEVER be changed. Doing so could break DICOM conformance.** Before modifying any parameters, such as time-out, be sure to have a backup of the originally supplied mergecom.pro file. Also, before modifying other parameters, you should consider contacting Merge Healthcare for advice.

6. Support of extended character sets

Not supported.

Appendix G: WORK_SCU Conformance Statement

0. Introduction

This is a conformance statement for the Merge sample program (MERGE_WORK_SCU) which supports DICOM Modality Worklist Services as a Service Class User(SCU).

DICOM has been implemented by Merge Healthcare and is called Merge DICOM Toolkit. Therefore, Merge DICOM Toolkit and DICOM can and are used synonymously within this document.

1. Implementation Model

MERGE_WORK_SCU with Merge DICOM Toolkit input and output is, very basically, an implementation of a DICOM Modality Worklist Service Class User (SCU) which can send DICOM queries to a DICOM Modality Worklist Service Class provider (SCP).

1.1 Application data flow diagram

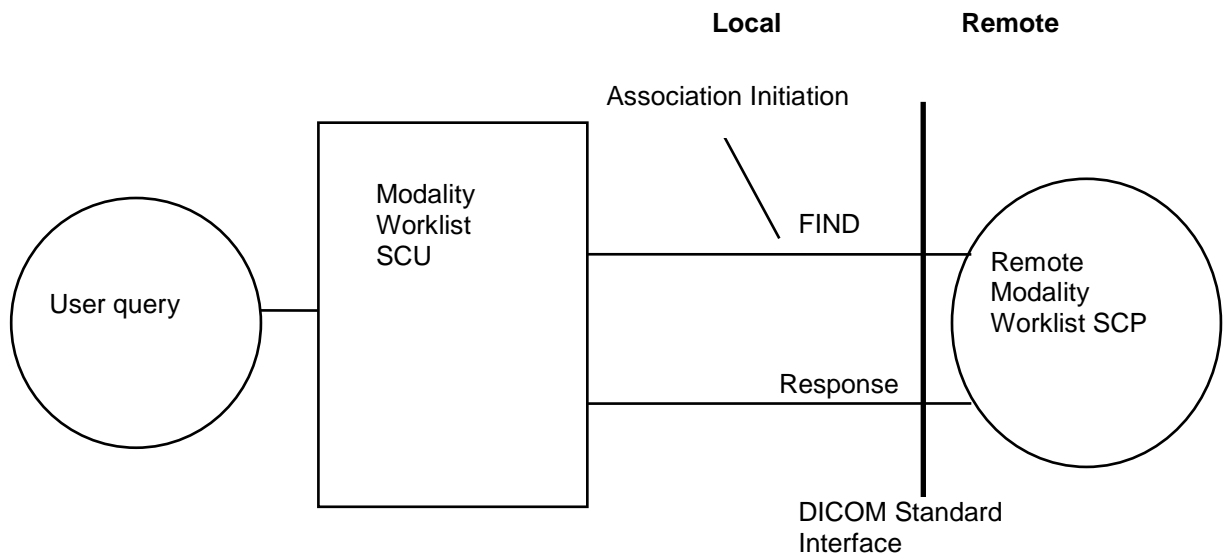


Figure G.1: Application data flow diagram for Modality Worklist SCU

1.2 Functional definition of Application Entity (AE)

All communication and image transfer with the remote application is accomplished utilizing the DICOM protocol over a network using the TCP/IP protocol stack.

The `MERGE_WORK_SCU` establishes an association with a user selected TCP/IP port number that is configured at the time this application is initiated. When an association is requested with a SCP, `MERGE_WORK_SCU` responds with a list of SOP Class UIDs that it will accept. If a find request is sent then it will wait for find responses.

2 AE Specifications

2.1 AE specification for `MERGE_WORK_SCU`

`MERGE_WORK_SCU` in conjunction with Merge DICOM Toolkit, provides Standard Conformance to the following DICOM V3.0 Service Object Pair (SOP) Class as a Modality Worklist Service Class User (SCU).

Table G.1: Valid SCU/SCP STORE SOP Classes for `MERGE_WORK_SCU` AE

SOP Class UID	SOP Class Name
1.2.840.10008.1.1	Verification SOP Class

`MERGE_WORK_SCU`, in conjunction with Merge DICOM Toolkit, provides Standard Conformance to the DICOM V3.0 Service Object Pair (SOP) Classes as a Modality Worklist Service Class User (SCU), when providing the function of querying an archival database for obtaining patient demographic data.

Table G.2: Valid SCP Query/Retrieve SOP Classes for `MERGE_WORK_SCU` AE

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.31	Modality Worklist Find

2.1.1 Association establishment policies for `MERGE_WORK_SCU` AE

2.1.1.1 General

The `MERGE_WORK_SCU` application will initiate an association as a Modality Worklist Service Class User requesting modality and patient data.

2.1.1.2 Number of associations

The `MERGE_WORK_SCU` AE allows a single association for association initiation, and is NOT configurable prior to run-time.

2.1.1.3 Asynchronous nature

The `MERGE_WORK_SCU` AE does not support asynchronous communication (multiple outstanding transactions over a single association).

2.1.1.4 Implementation identifying information

The Implementation Class Unique Identifier (UID) for the MERGE_WORK_SCU Application Entity is:

2.16.840.1.113669.2.1.1 (place your Implementation Class UID here).

The Implementation Version Name for the MERGE_WORK_SCU Application Entity is:

MergeCOM3_222 (place your Implementation Version Name here).

2.1.2 Association initiation by real-world activity for MERGE_WORK_SCU AE

The MERGE_WORK_SCU application initiates an association for the appropriate Modality Worklist Service Class that corresponds to the set of data requested to be transferred. The association is closed when all queries have been sent to the remote DICOM network node. MERGE_WORK_SCU is also able to abort the association through an operator requested abort or when an error occurs.

2.1.2.1 Real-world activity for Find and Move Execution operations of MERGE_WORK_SCU

A MERGE_WORK_SCU opens an association and performs C-FINDs. The association is closed after an error or when the initiator requests that it be closed.

2.1.2.1.1 Associated real-world activity for Find and Move Execution operations

Once the association has been established, the MERGE_WORK_SCU waits for transmission of conformant Modality Worklist Service messages. If a valid Find is received, then the local archival database is searched and the requested information is returned to the requester.

2.1.3.3.2 Presentation context table for Find and Move Execution operations

The presentation contexts that are proposed by MERGE_WORK_SCU AE for the Acceptable Find Execution operation are specified in the following table.

Table G.3: Find Execution Presentation Contexts of MERGE_WORK_SCU

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
Modality Worklist Find	1.2.840.10008.5.1.4.31	Explicit VR Little Endian	1.2.840.10008.1.2.1	SCP	None
		Implicit VR Little Endian	1.2.840.10008.1.2		
		Explicit VR Big Endian	1.2.840.10008.1.2.2		

3 Profiles

3.1 Supported Communication Stacks

`MERGE_WORK_SCP`, in conjunction with Merge DICOM Toolkit provides DICOM V3.0 TCP/IP Network Communication Support as defined in PS 3.8.

3.2 TCP/IP Stack

`MERGE_WORK_SCP` uses the Merge DICOM Toolkit to communicate over the TCP/IP protocol stack on any physical interconnection media supporting the TCP/IP stack. The Toolkit inherits the TCP/IP stack from the host operating system upon which it executes. The Toolkit has been implemented on almost every major operating system platform.

3.2.1 Physical Media Support

The `MERGE_WORK_SCP` AE is indifferent to the physical medium over which TCP/IP executes; it inherits this from the operating system on which it exists.

4. Extensions/specializations/privatizations

4.1 Standard extended/specialized/private SOPs

None supported.

4.2 Private Transfer Syntaxes

None supported.

5. Configuration

5.1 MERGE_WORK_SCU AE Configuration Files

The `MERGE_WORK_SCU` applications references four configuration files. The first, `merge.ini`, is found through the `MERGE_INI` environment variable. They are as follows:

<code>merge.ini</code>	Specifies the names of the other three configuration files and also contains message logging parameters.
<code>mergecom.pro</code>	Specifies run-time parameters for <code>MERGE_WORK_SCU</code> applications.
<code>mergecom.app</code>	Defines applications on other network nodes, to which connections are possible.
<code>mergecom.srv</code>	Service and sequence definitions.

5.1.1 AE title/presentation address mapping

Presentation address mapping is configured in the `mergecom.app` file. This is where the Host Name, Port Number, and Application Title map an Application Entity (AE) Title to a Presentation Address in TCP/IP for the provider to which you wish to connect. Similarly, the Presentation Address of your SCP as a

provider is specified by configuring the Listen Port in the mergecom.pro file, and specifying the AE title for your SCP within the application itself.

Note: The host name maps to an IP address as specified by your host table. Also, port 104 should always be used for standard connectivity; since this is the well-defined port for a DICOM server.

5.1.2 Configurable parameters

The mergecom.pro configuration file can be used to set or modify other lower-level communication parameters. This includes time-outs and other parameters. Some information about supported SOP classes is also stored here. **Most parameters in this file should NEVER be changed. Doing so could break DICOM conformance.** Before modifying any parameters, such as time-out, be sure to have a backup of the originally supplied mergecom.pro file. Also, before modifying other parameters, you should consider contacting Merge Healthcare for advice.

6. Support of extended character sets

Not supported.

Appendix H: WORK_SCP Conformance Statement

0. Introduction

This is a conformance statement for the Merge sample program (MERGE_WORK_SCP) which supports DICOM Modality Worklist Services as a Service Class Provider.

DICOM has been implemented by Merge Healthcare and is called Merge DICOM Toolkit. Therefore, Merge DICOM Toolkit and DICOM can and are used synonymously within this document.

1. Implementation Model

MERGE_WORK_SCP with Merge DICOM Toolkit input and output is, very basically, an implementation of a DICOM Modality Worklist Service Class Provider (SCP) which can receive DICOM queries from a DICOM Storage Service Class user (SCU).

1.1 Application data flow diagram

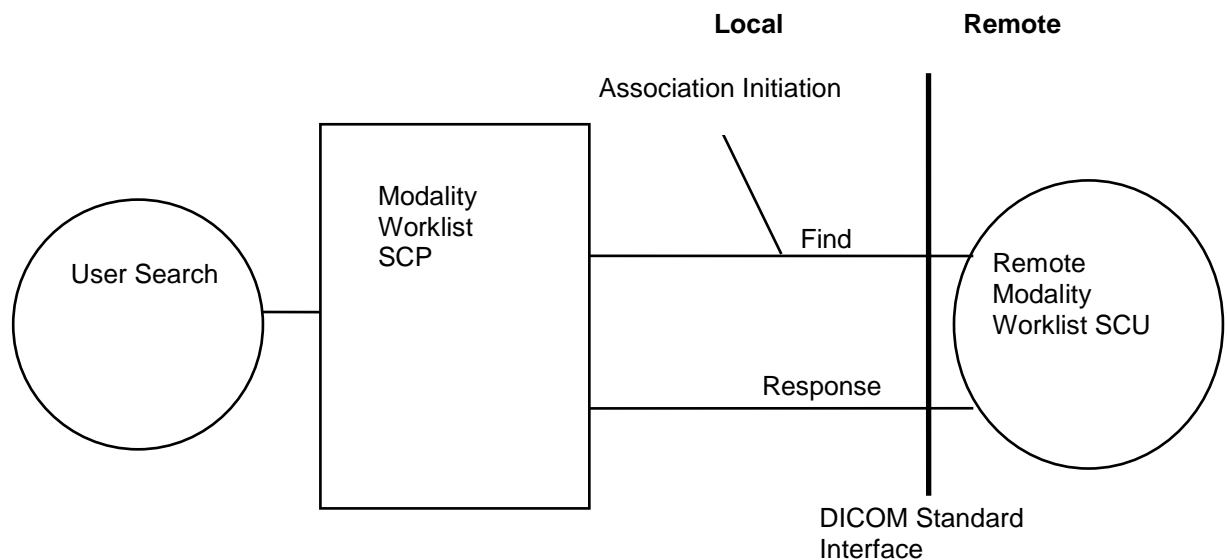


Figure H.1: Application Flow Diagram for Modality Worklist SCP

1.2 Functional definition of Application Entity (AE)

All communication is accomplished utilizing the DICOM protocol over a network using the TCP/IP protocol stack. MERGE_WORK_SCP will respond, if asked, with the Verification SOP Class UID as an SCP for one of its implemented SOP Classes.

The MERGE_WORK_SCP waits for an association to accept at the TCP/IP port number that is configured at the time this application is initiated. When an

association request is received with valid connection criteria, `MERGE_WORK_SCP` responds with a list of SOP Class UIDs that it will accept. It then waits for an Echo, or Find request to determine what specific function it has to perform. If an Echo is received, then an appropriate Echo response is sent back to the initiator. If a Find request is received, then the archive is searched for the requested information and a Find response is returned with all the found information.

2 AE Specifications

2.1 AE specification for `MERGE_WORK_SCP`

`MERGE_WORK_SCP` in conjunction with Merge DICOM Toolkit, provides Standard Conformance to the following DICOM V3.0 Service Object Pair (SOP) Class as a Verification Service Class User and Provider (SCU & SCP). As an SCP it sends out an Echo response after it receives an Echo request from a remote AE.

Table H.1: Valid SCU/SCP Verification SOP Class for `MERGE_WORK_SCP` AE

SOP Class UID	SOP Class Name
1.2.840.10008.1.1	Verification SOP Class

Table H.2: Valid SCU/SCP FIND SOP Classes for `MERGE_WORK_SCP` AE

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.31	Modality Worklist Find

`MERGE_WORK_SCP`, in conjunction with Merge DICOM Toolkit, provides Standard Conformance to the following DICOM V3.0 Service Object Pair (SOP) Class as a Modality Worklist Service Class Provider (SCP), when providing the function of a database for finding patient demographic data.

2.1.1 Association establishment policies for `MERGE_WORK_SCP` AE

2.1.1.1 General

The `MERGE_WORK_SCP` application will wait for an association as an SCP for the Modality Worklist Service Class. When a Find request is received, a search is done of the archival database for the data with the requested attributes, and a list of found attributes is returned to the remote requester. The maximum PDU size is configurable.

2.1.1.2 Number of associations

The `MERGE_WORK_SCP` AE allows a single association for association acceptance, which is not configurable prior to run-time.

2.1.1.3 Asynchronous nature

The `MERGE_WORK_SCP` AE does not support asynchronous communication (multiple outstanding transactions over a single association).

2.1.1.4 Implementation identifying information

The Implementation Class Unique Identifier (UID) for the MERGE_WORK_SCP Application Entity is:

2.16.840.1.113669.2.1.1 (place your Implementation Class UID here).

The Implementation Version Name for the MERGE_WORK_SCP Application Entity is:

MergeCOM3_222 (place your Implementation Version Name here).

2.1.2 Association initiation by real-world activity for MERGE_WORK_SCP AE

The MERGE_WORK_SCP client application initiates an association for the appropriate Modality Worklist Service Class that corresponds to the data requested and returned. The association is closed when all data has been sent to the remote DICOM network node. The client is also able to abort the association through an operator requested abort or when an error occurs.

2.1.2.1 Real-world activity for Echo Check operation of MERGE_WORK_SCP

A MERGE_WORK_SCP client application initiates associations for the echo service. The association is closed either when a correct response is received or when a time-out occurs.

2.1.2.1.1 Associated real-world activity for Echo Check operation

An echo is performed by a MERGE_WORK_SCP client application by using the `MC_Wait_For_Association` or `MC_Read_Message`.

2.1.2.1.2 Proposed presentation contexts for Echo Check operation

MERGE_WORK_SCP supports the Verification SOP Class fully as specified in the DICOM Standard.

The presentation context proposed by a MERGE_WORK_SCP client for the Echo Check operation are specified in the following table.

Table H.3: Echo Check Presentation Contexts of MERGE_WORK_SCP

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
Verification Service Class	1.2.840.10008.1.1	DICOM Implicit VR Little Endian	1.2.840.10008.1.2	SCU	None

2.1.2.1.3 SOP specific conformance for Verification SOP Class

No known SOP specific conformance issues.

2.1.3.3 Real-world activity for Find Execution operations of MERGE_WORK_SCP

A MERGE_WORK_SCP application waits for an association and offers to do C-FINDs. The association is closed after an error or when the initiator requests that it be closed.

2.1.3.3.1 Associated real-world activity for Find Execution operations

Once the association has been established, the MERGE_WORK_SCP waits for transmission of conformant Modality Worklist Service messages. If a valid Find is received, then the local archival database is searched and the requested information is returned to the requester.

2.1.3.3.2 Presentation context table for Find Execution operations

The presentation contexts that are proposed by MERGE_WORK_SCP AE for the Acceptable Find Execution operation are specified in the following table.

Table H.4: Find Execution Presentation Contexts of MERGE_WORK_SCP

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
Modality Worklist Find	1.2.840.10008.5.1.4.31	Explicit VR Little Endian	1.2.840.10008.1.2.1	SCP	None
		Implicit VR Little Endian	1.2.840.10008.1.2		
		Explicit VR Big Endian	1.2.840.10008.1.2.2		

2.1.3.3.3 Presentation context acceptance criterion for Find Execution operations

Not applicable since only a single presentation context for each Storage Service Class is supported.

2.1.3.3.4 Transfer syntax selection policies for Find and Move Execution operations

When executing on a Little Endian machine, transfer syntaxes are accepted in the following order:

Transfer Syntax UID	Transfer Syntax Name
1.2.840.10008.1.2	Explicit Little Endian Syntax
1.2.840.10008.1.2.1	Implicit Little Endian Syntax
1.2.840.10008.1.2.2	Explicit Big Endian Syntax

When executing on a Big Endian machine, transfer syntaxes are accepted in the following order:

Transfer Syntax UID	Transfer Syntax Name
1.2.840.10008.1.2.2	Explicit Big Endian Syntax
1.2.840.10008.1.2.1	Explicit Little Endian Syntax
1.2.840.10008.1.2	Implicit Little Endian Syntax

Note: This acceptance criteria can be overridden by the use of a “Transfer Syntax List” in the mergecom.app configuration file.

3. Profiles

3.1 Supported Communication Stacks

MERGE_WORK_SCP, in conjunction with Merge DICOM Toolkit provides DICOM V3.0 TCP/IP Network Communication Support as defined in PS 3.8.

3.2 TCP/IP Stack

MERGE_WORK_SCP uses the Merge DICOM Toolkit to communicate over the TCP/IP protocol stack on any physical interconnection media supporting the TCP/IP stack. The Toolkit inherits the TCP/IP stack from the host operating system upon which it executes. The Toolkit has been implemented on almost every major operating system platform.

3.2.1 Physical Media Support

The MERGE_WORK_SCP AE is indifferent to the physical medium over which TCP/IP executes; it inherits this from the operating system on which it exists.

4. Extensions/specializations/privatizations

4.1 Standard extended/specialized/private SOPs

None supported.

4.2 Private Transfer Syntaxes

None supported.

5. Configuration

The MERGE_WORK_SCP application references four configuration files. The first, merge.ini, is found through the MERGE_INI environment variable. They are as follows:

merge.ini	Specifies the names of the other three configuration files and also contains message logging parameters.
mergecom.pro	Specifies run-time parameters for the MERGE_WORK_SCP application.

mergecom.app	Defines applications on other network nodes, to which connections are possible.
mergecom.srv	Service and sequence definitions.

5.1.1 AE title/presentation address mapping

Presentation address mapping is configured in the mergecom.app file. This is where the Host Name, Port Number, and Application Title map an Application Entity (AE) Title to a Presentation Address in TCP/IP for the provider to which you wish to connect. Similarly, the Presentation Address of your SCP as a provider is specified by configuring the Listen Port in the mergecom.pro file, and specifying the AE title for your SCP within the application itself.

Note: The host name maps to an IP address as specified by your host table. Also, port 104 should always be used for standard connectivity; since this is the well-defined port for a DICOM server.

5.1.2 Configurable parameters

The mergecom.pro configuration file can be used to set or modify other lower-level communication parameters. This includes time-outs and other parameters. Some information about supported SOP classes is also stored here. **Most parameters in this file should NEVER be changed. Doing so could break DICOM conformance.** Before modifying any parameters, such as time-out, be sure to have a backup of the originally supplied mergecom.pro file. Also, before modifying other parameters, you should consider contacting Merge Healthcare for advice.

6. Support of extended character sets

Not supported.

Appendix I: Media FSU Conformance Statement

0. Introduction

This is a conformance statement for the Merge media services sample program (MERGE_MEDIA_FSU and MERGE_MEDIA) which supports DICOM Storage Services as a Service Class Provider (SCP) and DICOM Media Storage Services as a File Set Updator (FSU).

Note: MERGE_MEDIA_FSU is a **NON-CONFORMANT** application because it does not implement all of the Media operations required for a File Set Updator.

DICOM has been implemented by Merge Healthcare and is called Merge DICOM Toolkit. Therefore, Merge DICOM Toolkit and DICOM can and are used synonymously within this document.

1. Implementation Model

MERGE_MEDIA_FSU with Merge DICOM Toolkit input and output is, very basically, an implementation of a DICOM File Set Updator (FSU) which can store DICOM images to media with various DICOM SOP instances.

1.1 Application Data Flow Diagram

1.1.1 Accept images for storing to Media

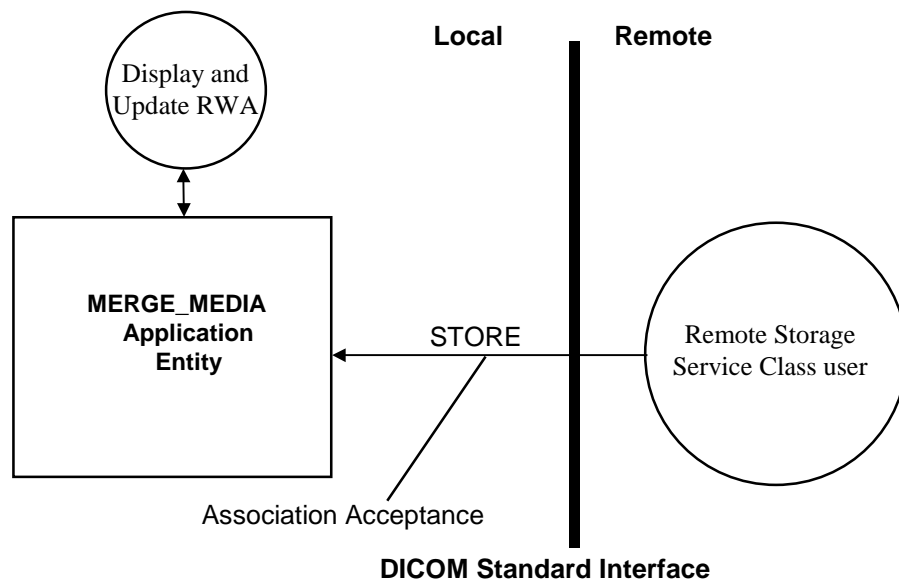


Figure I.1: MERGE_MEDIA application data flow diagram

1.1.2 Write Images to DICOM media

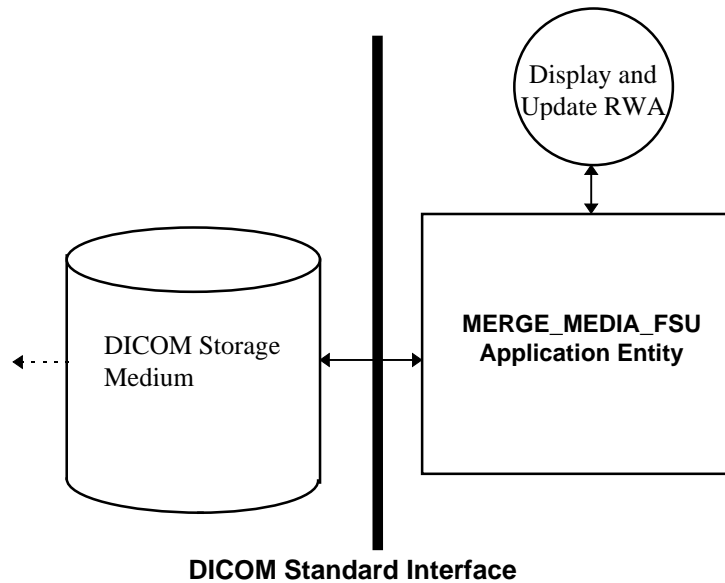


Figure 1.1: MERGE_MEDIA_FSU application data flow diagram

1.2 Functional Definition of Application Entity

Network related functions

All network communications and image transfer with remote applications is accomplished using the DICOM protocol over a network using the TCP/IP protocol stack.

The MERGE_MEDIA AE waits for an association to accept at the TCP/IP port number that is configured at the time this application is initiated. When association request is received with valid connection criteria, MERGE_MEDIA responds with a list of SOP Class UIDs that it will accept. It then waits for an Echo or Store request to determine what specific function it has to perform. If an Echo is received, an appropriate Echo response is sent back to the initiator. If a Store is received, all incoming images that are conformant to the association are written to disk for storage.

Media related functions

The MERGE_MEDIA_FSU AE conforms to the General Purpose CD-R Image Interchange (STD_GEN_CD) DICOM Application Profile as a File Set Updater (FSU). MERGE_MEDIA_FSU can perform the following functions:

- It can update a piece of media, writing additional SOP instances to an already existing DICOM File-set.
- It can write a new DICOM File-set onto media.
- It can display a directory listing of the File-set on a piece of media.

MERGE_MEDIA_FSU **DOES NOT** perform the following functions (and is therefore non-conformant to the DICOM definition of an FSU):

- It does not inquire the creation date and time of the files contained within the File-set.
- It does not inquire the remaining space left in the File-set.
- It does not access DICOM files other than the DICOMDIR.

1.3 Sequencing of real-world activities

There are no sequencing requirements.

1.4 File Meta Information Options

The MERGE_MEDIA AE writes the following File Meta Information into every file on the DICOM medium:

File Meta Information Version	0001	Hard coded in program.
Implementation Class UID	2.16.840.1.113669.2.1.1	Configurable option specified in the file "mergecom.pro".
Implementation Version Name	MergeCOM3_222	Configurable option specified in the file "mergecom.pro".

2. AE Specifications

Currently two DICOM application entities are present in the Merge media sample application. The MERGE_MEDIA AE handles DICOM network services and the MERGE_MEDIA_FSU handles DICOM media storage services.

All associations with the MERGE_MEDIA AE shall be established using the DICOM 3.0 Application Context. A single DICOM Application Context Name is defined for this version of the DICOM standard. The name is "1.2.840.10008.3.1.1.1".

2.1 AE Specification for MERGE_MEDIA

MERGE_MEDIA with Merge DICOM Toolkit, provides Standard Conformance to the following DICOM V3.0 Service Object Pair (SOP) Class as a Verification Service Class Provider (SCP). As an SCP it sends out an Echo response after it receives an Echo request from a remote AE.

Table I.1: Valid SCP Verification SOP Class for MERGE_MEDIA AE

SOP Class UID	SOP Class Name
1.2.840.10008.1.1	Verification SOP Class

MERGE_MEDIA_FSU, in conjunction with Merge DICOM Toolkit, provides Standard Conformance to the following DICOM V3.0 Service Object Pair (SOP) Classes as a Storage Service Class Provider (SCP).

Table I.2: Valid SCP Storage SOP Classes for MERGE_MEDIA AE

SOP Class UID	SOP Class Name
1.2.840.10008.5.1.4.1.1.1	Computed Radiography Image Storage
1.2.840.10008.5.1.4.1.1.2	CT Image Storage
1.2.840.10008.5.1.4.1.1.3.1	Ultrasound Multi-frame Image Storage
1.2.840.10008.5.1.4.1.1.4	MR Image Storage
1.2.840.10008.5.1.4.1.1.20	Nuclear Medicine Image Storage
1.2.840.10008.5.1.4.1.1.6.1	Ultrasound Image Storage
1.2.840.10008.5.1.4.1.1.7	Secondary Capture Image Storage
1.2.840.10008.5.1.4.1.1.8	Standalone Overlay Storage
1.2.840.10008.5.1.4.1.1.9	Standalone Curve Storage
1.2.840.10008.5.1.4.1.1.10	Standalone Modality LUT Storage
1.2.840.10008.5.1.4.1.1.11	Standalone VOI LUT Storage
1.2.840.10008.5.1.4.1.1.12.1	X-ray Angiographic Image Storage
1.2.840.10008.5.1.4.1.1.12.2	X-ray RadioFluoroscopic Image Storage
1.2.840.10008.5.1.4.1.1.12.3	X-ray Angiographic Bi-plane Image Storage
1.2.840.10008.5.1.4.1.1.128	Standard Positron Emission Tomography Image Storage
1.2.840.10008.5.1.4.1.1.129	Standard Positron Emission Tomography Curve Storage
1.2.840.10008.5.1.4.1.1.481.4	Standard RT Beams Treatment Record Storage
1.2.840.10008.5.1.4.1.1.481.6	Standard RT Brachy Treatment Record Storage
1.2.840.10008.5.1.4.1.1.481.1	Standard RT Image Storage
1.2.840.10008.5.1.4.1.1.481.2	Standard RT Dose Storage
1.2.840.10008.5.1.4.1.1.481.3	Standard RT Structure Set Storage
1.2.840.10008.5.1.4.1.1.481.5	Standard RT Plan Storage

1.2.840.10008.5.1.4.1.1.481.7	Standard RT Treatment Summary Record Storage
1.2.840.10008.5.1.1.30	Hardcopy Color Image Storage
1.2.840.10008.5.1.1.29	Hardcopy Grayscale Image Storage
1.2.840.10008.5.1.1.27	Stored Print Storage
1.2.840.10008.5.1.4.1.1.1.1	Digital X-Ray Image Storage - For Presentation
1.2.840.10008.5.1.4.1.1.1.1.1	Digital X-Ray Image Storage - For Processing
1.2.840.10008.5.1.4.1.1.1.3	Digital Intra-oral X-Ray Image Storage - For Presentation
1.2.840.10008.5.1.4.1.1.1.3.1	Digital Intra-oral X-Ray Image Storage - For Processing
1.2.840.10008.5.1.4.1.1.1.2	Digital Mammography Image Storage - For Presentation
1.2.840.10008.5.1.4.1.1.1.2.1	Digital Mammography Image Storage - For Processing
1.2.840.10008.5.1.4.1.1.77.1.1	VL Endoscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.2	VL Microscopic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.4	VL Photographic Image Storage
1.2.840.10008.5.1.4.1.1.77.1.3	VL Slide-Coordinates Microscopic Image Storage

Note: MERGE_MEDIA operates as an SCP of the DICOM Storage Services. Therefore, it does not support all of the services mentioned earlier in this document, but only those services relevant to the DICOM Storage Services.

2.1.1 Association establishment policies for MERGE_MEDIA AE

2.1.1.1 General

The MERGE_MEDIA application will wait for an association as an SCP of Storage Services. When a Store request is received, the corresponding images are either saved to DICOM files on disk. The maximum PDU size is configurable from a minimum of 4,096 bytes.

2.1.1.2 Number of associations

Under multi-tasking systems such as UNIX, the MERGE_MEDIA AE does allow multiple simultaneous Store associations. All other systems only allow a single association at any point in time.

2.1.1.3 Asynchronous nature

The MERGE_MEDIA AE does not support asynchronous communication (multiple outstanding transactions over a single association).

2.1.1.4 Implementation identifying information

The Implementation Class Unique Identifier (UID) for the MERGE_MEDIA Application Entity is:

2.16.840.1.113669.2.1.1 (place your Implementation Class UID here).

The Implementation Version Name for the MERGE_MEDIA Application Entity is:

MergeCOM3_222 (place your Implementation Version Name here).

2.1.2 Association acceptance by real-world activity

The MERGE_MEDIA client application accepts an association for the appropriate Storage Service Class that corresponds to the set of images requested to be transferred. The association is closed by the Storage Service Class user which initiated the association.

MERGE_MEDIA is able to abort the association when an error occurs.

2.1.2.1 Real-world activity: Echo Check

2.1.2.2.1 Associated real-world activity for Echo Check operation

MERGE_MEDIA with Merge DICOM Toolkit, provides Standard Conformance to the following DICOM V3.0 Service Object Pair (SOP) Class as a Verification Service Class Provider (SCP). As an SCP it sends out an Echo response after it receives an Echo request from a remote AE.

Table I.3: Valid SCP Verification SOP Class for MERGE_MEDIA AE

SOP Class Name	SOP Class UID
Verification SOP Class	1.2.840.10008.1.1

2.1.2.1.2 Proposed presentation contexts for Echo Check operation

MERGE_MEDIA supports the Verification SOP Class fully as specified in the DICOM Standard.

The presentation context proposed by a MERGE_MEDIA client for the Echo Check operation are specified in the following table.

Table I.4: Echo Check Presentation Contexts of MERGE_MEDIA

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
Verification Service Class	1.2.840.10008.1.1	Explicit VR Little Endian	1.2.840.10008.1.2.1	SCP	None
		Implicit VR Little Endian	1.2.840.10008.1.2		
		Explicit VR Big Endian	1.2.840.10008.1.2.2		

2.1.2.2 Real-world activity: Receive Images operation

2.1.2.2.1 Associated real-world activity for Receive Image Operation

MERGE_MEDIA waits for an association and offers to do the Image Storage service. The association is closed after an error or when the initiator requests that it be closed.

2.1.2.2.2 Presentation context table for Receive Image operations

Table I.5: Receive Image Presentation Contexts of MERGE_MEDIA_FSU

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
All Services Listed in Table I.2	All Services Listed in Table I.2	Explicit VR Little Endian	1.2.840.10008.1.2.1	SCP	None
		Implicit VR Little Endian	1.2.840.10008.1.2		
		Explicit VR Big Endian	1.2.840.10008.1.2.2		

2.1.2.2.2.1 SOP specific conformance for all storage SOP Classes

The MERGE_MEDIA AE responds to a C-STORE request with one of the response codes in the following table.

Table I.6: SOP specific conformance

Service Status	Status Description	Status Code (0000,0900)	Related Fields
----------------	--------------------	-------------------------	----------------

REFUSED	Out of Resources - There were insufficient resources to process the request. The request was not processed	A765	(0000,0902) contains a short description of the condition.
ERROR	Data Set does not match SOP Class - A required attribute is not present in the message. The request was not processed.	A965	(0000,0901) contains a listing of attribute tags missing. (0000,0902) contains a short description of the condition.
	Cannot understand - The message was not properly DICOM-encoded. The request was not processed.	C065	(0000,0902) contains a short description of the condition.
	Processing failure - A condition arose which prevented the processing of the request	0111	None
SUCCESS		0000	None

2.1.2.2.2 Presentation context acceptance criterion for Receive Image operations

Not applicable since only a single presentation context for each Storage Service Class is supported.

2.1.2.2.3 Transfer syntax selection policies for Receive Image operations

When executing on a Little Endian machine, transfer syntaxes are accepted in the following order:

Transfer Syntax UID	Transfer Syntax Name
1.2.840.10008.1.2	Explicit Little Endian Syntax
1.2.840.10008.1.2.1	Implicit Little Endian Syntax
1.2.840.10008.1.2.2	Explicit Big Endian Syntax

When executing on a Big Endian machine, transfer syntaxes are accepted in the following order:

Transfer Syntax UID	Transfer Syntax Name
1.2.840.10008.1.2.2	Explicit Big Endian Syntax
1.2.840.10008.1.2.1	Explicit Little Endian Syntax
1.2.840.10008.1.2	Implicit Little Endian Syntax

Note: This acceptance criteria can be overridden by the use of a “Transfer Syntax List” in the mergecom.app configuration file.

2.2 AE Specification for MERGE_MEDIA_FSU

MERGE_MEDIA_FSU provides **PARTIAL** conformance to DICOM Interchange Option of the Media Storage Service Class. The Application Profiles and roles are listed below.

Table I.7: AE Related Application Profiles, Real-World Activities, and Roles

Supported Application Profile	Real-World Activity	Roles	SC Option
STD_GEN_CD	Update DICOM Media	FSU	Interchange
	Create DICOM Media	FSC	Interchange

Note: The STD_GEN_CD application profile is for writing to CD-R. The MERGE_MEDIA_FSU AE actually writes to a local directory on disk and does not support directly writing to CD-R.

2.2.1 File Meta Information for the Application Entity MERGE_MEDIA_FSU

The Application Entity Title is MERGE_MEDIA_FSU.

2.2.2 Real World Activities for MERGE_MEDIA_FSU

2.2.2.1 Create and Update DICOM medium for exchange

MERGE_MEDIA_FSU acts as an FSU using the Interchange option when requested to provide a directory listing or updating a File-set.

The MERGE_MEDIA_FSU Display and Update Application will add SOP instances received over a DICOM association to a File-set.

2.1.2.1.1 Media Storage Application Profile for create and update DICOM medium

For a list of the Application Profiles that invoke this AE for the Display and Update RWA, see Table A.1. There are no extensions or specializations.

3. Profiles

3.1 Supported Communication Stacks

MERGE_MEDIA_FSU, in conjunction with Merge DICOM Toolkit provides DICOM V3.0 TCP/IP Network Communication Support as defined in PS 3.8.

3.2 TCP/IP Stack

MERGE_MEDIA_FSU uses the Merge DICOM Toolkit to communicate over the TCP/IP protocol stack on any physical interconnection media supporting the TCP/IP stack. The toolkit inherits the TCP/IP stack from the host operating system upon which it executes. The Toolkit has been implemented on almost every major operating system platform.

3.2.1 Physical Media Support

The MERGE_MEDIA AE is indifferent to the physical medium over which TCP/IP executes; it inherits this from the operating system on which it exists.

3.3 Augmented and Private Application Profiles

No augmented or private application profiles are supported by the MERGE_MEDIA_FSU implementation of the Media Storage Service Class.

4. Extensions, Specializations, and Privatizations of SOP Classes and Transfer Syntaxes

None.

5. Configuration

5.1 MERGE_MEDIA_FSU and MERGE_MEDIA Configuration Files

The MERGE_MEDIA_FSU application references four configuration files. The first, merge.ini, is found through the MERGE_INI environment variable. They are as follows:

merge.ini	Specifies the names of the other three configuration files and also contains message logging parameters.
mergecom.pro	Specifies run-time parameters for the MERGE_MEDIA_FSU application.
mergecom.app	Defines applications on other network nodes, to which are possible.
mergecom.srv	Service and sequence definitions.

The mergecom.pro configuration file can be used to set or modify other network communication parameters. These parameters do not affect the media storage class.

5.1.1 AE title/presentation address mapping

Presentation address mapping is configured in the mergecom.app file. The Presentation Address of an SCP application as a provider is specified by configuring the Listen Port in the mergecom.pro file, and specifying the AE title for the SCP within the application itself.

5.1.2 Other Configurable parameters for MERGE_MEDIA_FSU AE and MERGE_MEDIA AE

The mergecom.pro configuration file can be used to set or modify other lower-level communication parameters. This includes time-outs and other parameters. Some information about supported SOP classes is also stored here. **Most parameters in this file should NEVER be changed. Doing so could break DICOM conformance.** Before modifying any parameters, such as time-out, be sure to have a backup of the originally supplied mergecom.pro file. Also, before modifying other parameters, you should consider contacting Merge Healthcare Software for advice.

5.2 File Meta Information for MERGE_MEDIA_FSU AE

The mergecom.pro configuration file contains the implementation class UID and implementation version names which are set in the file meta information.

6. Support of extended character sets

Not supported.