PS3.10

DICOM PS3.10 2024c - Media Storage and File Format for Media Interchange

PS3.10: DICOM PS3.10 2024c - Media Storage and File Format for Media Interchange

A DICOM® publication

Table of Contents

List of Figures

List of Tables

Notice and Disclaimer

The information in this publication was considered technically sound by the consensus of persons engaged in the development and approval of the document at the time it was developed. Consensus does not necessarily mean that there is unanimous agreement among every person participating in the development of this document.

NEMA standards and guideline publications, of which the document contained herein is one, are developed through a voluntary consensus standards development process. This process brings together volunteers and/or seeks out the views of persons who have an interest in the topic covered by this publication. While NEMA administers the process and establishes rules to promote fairness in the development of consensus, it does not write the document and it does not independently test, evaluate, or verify the accuracy or completeness of any information or the soundness of any judgments contained in its standards and guideline publications.

NEMA disclaims liability for any personal injury, property, or other damages of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, application, or reliance on this document. NEMA disclaims and makes no guaranty or warranty, expressed or implied, as to the accuracy or completeness of any information published herein, and disclaims and makes no warranty that the information in this document will fulfill any of your particular purposes or needs. NEMA does not undertake to guarantee the performance of any individual manufacturer or seller's products or services by virtue of this standard or guide.

In publishing and making this document available, NEMA is not undertaking to render professional or other services for or on behalf of any person or entity, nor is NEMA undertaking to perform any duty owed by any person or entity to someone else. Anyone using this document should rely on his or her own independent judgment or, as appropriate, seek the advice of a competent professional in determining the exercise of reasonable care in any given circumstances. Information and other standards on the topic covered by this publication may be available from other sources, which the user may wish to consult for additional views or information not covered by this publication.

NEMA has no power, nor does it undertake to police or enforce compliance with the contents of this document. NEMA does not certify, test, or inspect products, designs, or installations for safety or health purposes. Any certification or other statement of compliance with any health or safety-related information in this document shall not be attributable to NEMA and is solely the responsibility of the certifier or maker of the statement.

Foreword

This DICOM Standard was developed according to the procedures of the DICOM Standards Committee.

The DICOM Standard is structured as a multi-part document using the guidelines established in [ISO/IEC Directives, Part 2].

DICOM® is the registered trademark of the National Electrical Manufacturers Association for its standards publications relating to digital communications of medical information, all rights reserved.

HL7® and CDA® are the registered trademarks of Health Level Seven International, all rights reserved.

SNOMED®, SNOMED Clinical Terms®, SNOMED CT® are the registered trademarks of the International Health Terminology Standards Development Organisation (IHTSDO), all rights reserved.

LOINC® is the registered trademark of Regenstrief Institute, Inc, all rights reserved.

1 Scope and Field of Application

This Part of the DICOM Standard specifies a general model for the storage of Medical Imaging information on removable media. The purpose of this Part is to provide a framework allowing the interchange of various types of medical images and related information on a broad range of physical storage media.

This Part specifies:

a. a layered model for the storage of medical images and related information on storage media. This model introduces the concept of Media Storage Application Profiles, which specify application specific subsets of the DICOM Standard to which a Media Storage implementation may claim conformance. Such a conformance applies only to the writing, reading and updating of the content of storage media. Specific Application Profiles are not included in this Part but in PS3.11;

b. a DICOM File Format supporting the encapsulation of any Information Object Definition;

c. a Secure DICOM File Format supporting the encapsulation of a DICOM File Format in a cryptographic envelope;

d. a DICOM File Service providing independence from the underlying media format and physical media. The policies specific to the DICOMDIR file used to store the Media Storage Directory Service/Object Pair Class are also addressed.

This Part is related to other parts of the DICOM Standard in that:

- PS3.2, Conformance, specifies the requirements that shall be met to achieve DICOM Conformance in Media Storage;
- PS3.3, Information Object Definitions, specifies a number of Information Object Definitions (e.g., various types of images) that may be used in conjunction with this Part;
- PS3.4, builds upon this Part to define the Media Storage Service Class;
- PS3.5, Data Structure and Encoding, addresses the encoding rules necessary to construct a Data Set that is encapsulated in a file as specified in this Part;
- PS3.6, Data Dictionary, contains a registry by Tag of all Data Elements related to the Attributes of Information Objects defined in PS3.3. This index includes the Value Representation and Value Multiplicity for each Data Element;
- PS3.11, Media Storage Application Profiles standardizes a number of choices related to a specific clinical need (selection of a Physical Medium and Media Format as well as specific Service/Object Pair Classes). It aims at facilitating the interoperability between implementations that claim conformance to the same Application Profile. PS3.11 is intended to be extended as the clinical needs for Media Storage Interchange evolve;
- PS3.12, Media Formats and Physical Media for Data Interchange, defines a number of selected Physical Medium and corresponding Media Formats. These Media Formats and Physical Medium selections are referenced by one or more of the Application Profiles of PS3.11. PS3.12 is intended to be extended as the technologies related to Physical Medium evolve.
- PS3.15, Security Profiles defines a number of profiles for use with Secure DICOM Media Storage Application Profiles. The Media Storage Security Profiles specify the cryptographic techniques to be used for each Secure DICOM File in a Secure Media Storage Application Profile.

PS3.10 lays a foundation for open Media Interchange by standardizing an overall architecture and addressing some of the major barriers to interoperability: the definition of a DICOM File Format, a DICOM File Service and the policies associated with a Media Storage Directory structure.

> Note
>
> PS3.3 specifies a general medical imaging Basic Directory Information Object Definition and PS3.4 specifies the corresponding Media Storage Directory SOP Class that is a member of the Media Storage Service Class.

Adherence to the provisions of PS3.10 by implementations reading, writing or updating Storage Media represents a key foundation for open Storage Media Interchange. However, it is only with the selection of standard Physical Media and corresponding Media Formats in PS3.12 and the use of specific Application Profiles in PS3.11 that effective Media Storage Interchange interoperability is achieved. Therefore, claiming conformance to PS3.10 only, is not a valid DICOM Conformance Statement. DICOM Media Storage Conformance shall be made in relation to a PS3.11 Application Profile according to the framework defined by PS3.2.

2 References

2.1 Normative References

The following standards contain provisions that, through reference in this text, constitute provisions of this Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibilities of applying the most recent editions of the standards indicated below.

[ISO/IEC Directives, Part 2] ISO/IEC. 2016/05. 7.0. *Rules for the structure and drafting of International Standards*. http://www.iec.ch/members_experts/refdocs/iec/isoiecdir-2%7Bed7.0%7Den.pdf .

[ISO 7498-1] ISO. 1994. *Information Processing Systems - Open Systems Interconnection - Basic Reference Model*.

[ISO 7498-2] ISO. 1989. *Information processing systems - Open Systems Interconnection - Basic reference Model - Part 2: Security Architecture*.

[ISO/TR 8509] ISO. *Information Processing Systems - Open Systems Interconnection - Service Conventions*. *ISO/TR 8509 has been withdrawn. See ISO/IEC 2382-26:1993 Information technology - Vocabulary - Part 26: Open systems interconnection* .

[ISO 8822] ISO. 1988. *Information processing systems - Open Systems Interconnection - Connection oriented presentation service definition*.

[ISO/IEC 8859-1] ISO/IEC. 1987. *Information processing - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No. 1*.

[RFC2557] IETF. March 1999. *MIME Encapsulation of Aggregate Documents, such as HTML (MHTML)*. http://tools.ietf.org/html/rfc2557 .

[RFC3240] IETF. February 2002. *Digital Imaging and Communications in Medicine (DICOM) - Application/dicom MIME Sub-type Registration*. http://tools.ietf.org/html/rfc3240 .

[RFC5652] IETF. September 2009. *Cryptographic Message Syntax*. http://tools.ietf.org/html/rfc5652 .

3 Definitions

For the purposes of this Standard the following definitions apply.

3.1 Reference Model Definitions

This Part of the Standard use of the following terms defined in [ISO 7498-1] and [ISO 7498-2]:

Application Entity	See [ISO 7498-1].
Application Process	See [ISO 7498-1].
Service	See [ISO 7498-1].
Transfer Syntax	See [ISO 7498-1].
Data Confidentiality	See [ISO 7498-2]. Note The definition is "the property that information is not made available or disclosed to unauthorized individuals, entities or processes."
Data Origin Authentication	See [ISO 7498-2]. Note The definition is "the corroboration that the source of data received is as claimed."
Data Integrity	See [ISO 7498-2]. Note The definition is "the property that data has not been altered or destroyed in an unauthorized manner."

3.2 Service Conventions Definitions

This Part of the Standard makes use of the following terms defined in [ISO/TR 8509]:

Service Provider	See [ISO/TR 8509].
Service User	See [ISO/TR 8509].

3.3 Presentation Service Definitions

This Part of the Standard makes use of the following terms defined in [ISO 8822]:

Abstract Syntax	See [ISO 8822].
Abstract Syntax Name	See [ISO 8822].

3.4 DICOM Introduction and Overview Definitions

This Part of the Standard makes use of the following terms defined in PS3.1:

Attribute	See Attribute in PS3.1.
Service-Object Pair Class (SOP Class)	See Service-Object Pair Class in PS3.1.

3.5 DICOM Information Object Definitions

This Part of the Standard makes use of the following terms defined in PS3.3:

Information Object Definition (IOD)
See Information Object Definition in PS3.3.

3.6 DICOM Data Structure and Encoding Definitions

This Part of the Standard makes use of the following terms defined in PS3.5:

Data Element
See Data Element in PS3.5.

Data Set
See Data Set in PS3.5.

Data Element Type
See Data Element Type in PS3.5.

Value
See Value in PS3.5.

Value Multiplicity
See Value Multiplicity in PS3.5.

Value Representation
See Value Representation in PS3.5.

3.7 DICOM Message Exchange Definitions

This Part of the Standard makes use of the following terms defined in PS3.7:

Implementation Class UID
See Implementation Class UID in PS3.7.

3.8 DICOM Media Storage and File Format Definitions

The following definitions are commonly used in this Part of the Standard:

Media Storage Application Profile
A specification that defines a selection of choices at the various layers of the DICOM Media Storage Model that are applicable to a specific need or context in which the media interchange is intended to be performed.

DICOM File Service
A minimum abstract view of files and operations to be provided by the Media Format Layer. Constraining access to the content of files by the Application Entities through such a DICOM File Service boundary ensures Media Format and Physical Media independence.

DICOM File
A File with a content formatted according to the requirements of this Part of the DICOM Standard. In particular such files contain the File Meta Information and a properly formatted Data Set.

DICOMDIR File
A DICOM File within a File-set that contains a Media Storage Directory SOP Instance. This File is given a single component File ID, DICOMDIR.

File
A File is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the File. Files are identified by a unique File ID and may by written, read and/or deleted.

File ID
Identifier for a File, which is unique within the context of the File-set to which it belongs. A set of ordered File ID Components (up to a maximum of eight) forms a File ID.

File ID Component
A string of one to eight characters of a defined character set.

File Meta Information
Identifying information on the encapsulated Data Set. It is a header at the beginning of every DICOM File.

File-set
A collection of DICOM Files (and possibly non-DICOM Files) that share a common naming space within which File IDs are unique.

File-set Creator

An Application Entity that creates the DICOMDIR File (see Section 8.6) and zero or more DICOM Files.

File-set Reader

An Application Entity that accesses one or more files in a File-set.

File-set Updater

An Application Entity that accesses Files, creates additional Files, or deletes existing Files in a File-set. A File-set Updater makes the appropriate alterations to the DICOMDIR file reflecting the additions or deletions.

DICOM File Format

The means to encapsulate in a File the Data Set representing a SOP Instance related to a DICOM Information Object.

Media Format

Data structures and associated policies that organize the bit streams defined by the Physical Media format into data file structures and associated file directories.

DICOM Media Storage Model

The data structures and operations used at different protocol layers to achieve interoperability through media interchange.

Media Storage Services

A set of operations with media that facilitate storage to and retrieval from the media of DICOM SOP Instances. Part of the DICOM File Service specification.

Physical Media

A piece of material with recording capabilities for streams of bits. Characteristics of a Physical Media include form factor, mechanical characteristics, recording properties and rules for recording and organizing bit streams in accessible structures

Secure DICOM File

A DICOM File that is encapsulated with the Cryptographic Message Syntax specified in IETF STD 70 [RFC5652].

Secure File-set

A File-set in which all DICOM Files are Secure DICOM Files.

Secure Media Storage Application Profile

A Media Storage Application Profile that requires a Secure File-set.

3.9 DICOM Service Class Definitions

This Part of the Standard makes use of the following terms defined in PS3.4 of the DICOM Standard:

Service Object Pair Instance (SOP Instance)

See Service-Object Pair Instance in PS3.4.

4 Symbols and Abbreviations

The following symbols and abbreviations are used in this Part of the Standard.

ACC	American College of Cardiology
ACR	American College of Radiology
ASCII	American Standard Code for Information Interchange
AE	Application Entity
ANSI	American National Standards Institute
CEN/TC/251	Comite Europeen de Normalisation - Technical Committee 251 - Medical Informatics
DICOM	Digital Imaging and Communications in Medicine
FSC	File-set Creator
FSR	File-set Reader
FSU	File-set Updater
HL7	Health Level 7
HTML	Hypertext Transfer Markup Language
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Standards Organization
ID	Identifier
IOD	Information Object Definition
JIRA	Japan Medical Imaging and Radiological Systems Industries Association
MIME	Multipurpose Internet Mail Extensions
NEMA	National Electrical Manufacturers Association
OSI	Open Systems Interconnection
SOP	Service-Object Pair
TCP/IP	Transmission Control Protocol/Internet Protocol
UID	Unique Identifier
VR	Value Representation
XML	Extensible Markup Language

5 Conventions

Words are capitalized in this document to help the reader understand that these words have been previously defined in Section 3 of this document and are to be interpreted with that meaning.

A Tag is represented as (gggg,eeee), where gggg equates to the Group Number and eeee equates to the Element Number within that Group. Tags are represented in hexadecimal notation as specified in PS3.5.

Attributes of File Meta Information are assigned a Type that indicates if a specific Attribute is required depending on the Media Storage Services. The following Type designations are derived from the PS3.5 designations but take into account the Media Storage environment:

- Type 1: Such Attributes shall be present with an explicit Value in files created by File-set Creators and File-set Updaters. They shall be supported by File-set Readers and File-set Updaters;
- Type 1C: Such Attributes shall be present with an explicit Value in Files created by File-set Creators and File-set Updaters if the specified condition is met. They shall be supported by File-set Readers and File-set Updaters;
- Type 2: Such Attributes shall be present with an explicit Value or with a zero-length Value if unknown, in Files created by File-set Creators and File-set Updaters. They shall be supported by File-set Readers and File-set Updaters;
- Type 2C: Such Attributes shall be present with an explicit Value or with a zero-length if unknown, in Files created by File-set Creators and File-set Updaters if the specified condition is met. They shall be supported by File-set Readers and File-set Updaters;
- Type 3: Such Attributes may be present with an explicit Value or a zero-length Value in Files created by File-set Creators and File-set Updaters. They may be supported or ignored by File-set Readers and File-set Updaters.

6 DICOM Models for Media Storage

This section defines the DICOM Media Storage Model used by DICOM Application Entities for the purpose of communication through the interchange of removable storage media. Specifically, this Section provides a model to clarify a number of concepts for digital imaging and communications and introduces key terms used throughout the DICOM Standard. This model has been used to partition the DICOM Standard into separate parts related to storage media interchange.

6.1 General DICOM Communication Model

Figure 5-1 in PS3.1 presents the general communication model of the DICOM Standard, which spans both network (on-line) and media interchange (off-line) communication. Application Entities may utilize any of the following transport mechanisms:

- the DICOM Message Service and Upper Layer Service, which provides independence from specific physical networking communication support and protocols such as TCP/IP,
- the DICOM Web Service API and HTTP Service, which allows use of common hypertext and associated protocols for transport of DICOM services,
- the Basic DICOM File Service, which provides access to Storage Media independently from specific physical media storage formats and file structures, or
- DICOM Real-Time Communication, which provides real-time transport of DICOM metadata based on SMPTE and RTP.

PS3.10 describes the Basic DICOM File Service, as depicted in Figure 6.1-1.

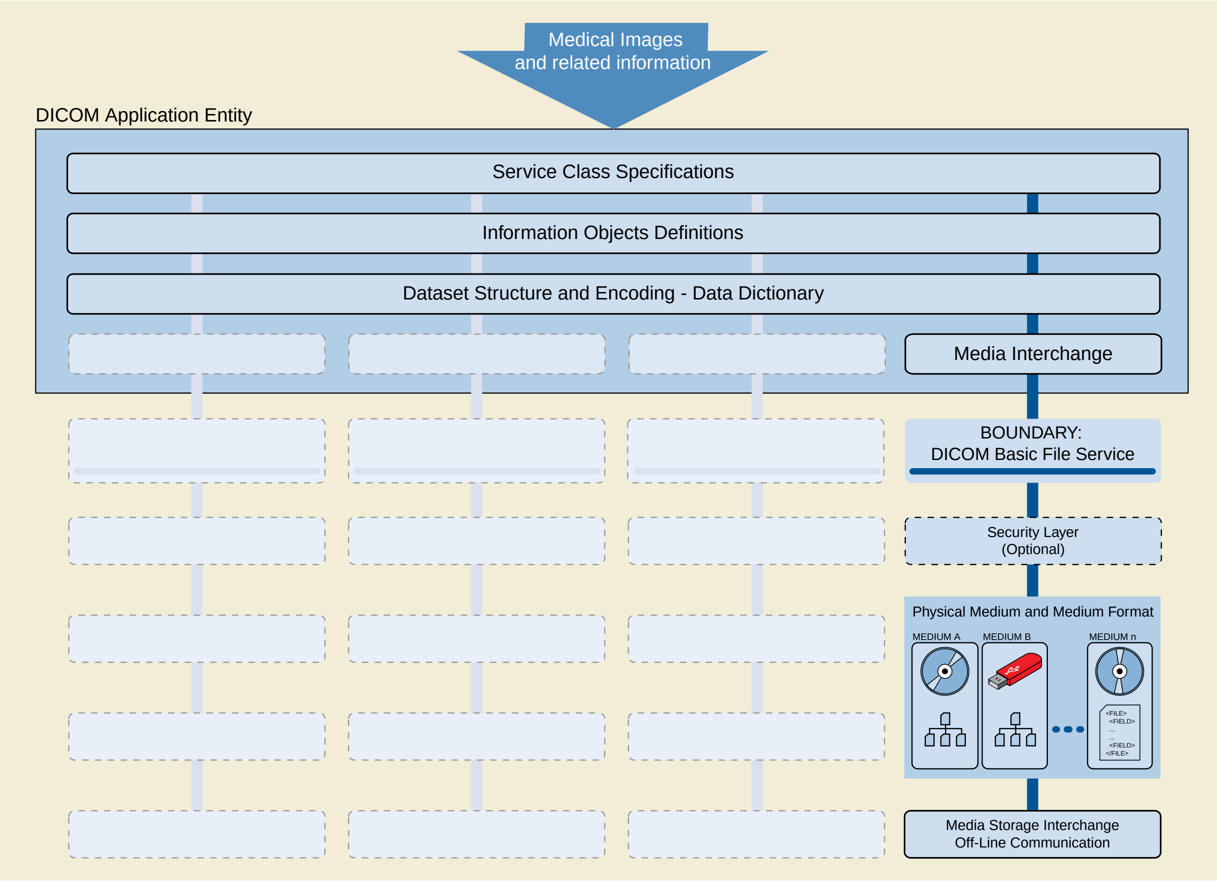


Figure 6.1-1. DICOM Communication Model for Media Interchange

6.2 The DICOM Media Storage Model

The DICOM Media Storage Model is presented by Figure 6.2-1 and expands on the General DICOM Communication Model introduced earlier in Section 6.1.

The DICOM Media Storage Model focuses on the aspects directly related to data interchange through removable storage media. It pertains to the data structures and associated rules used at different layers to achieve interoperability through media interchange. The Services identified in this Model are simple boundaries between functional layers.

Note

It is not within the scope of this Standard to specify Application Programming Interfaces at these boundaries.

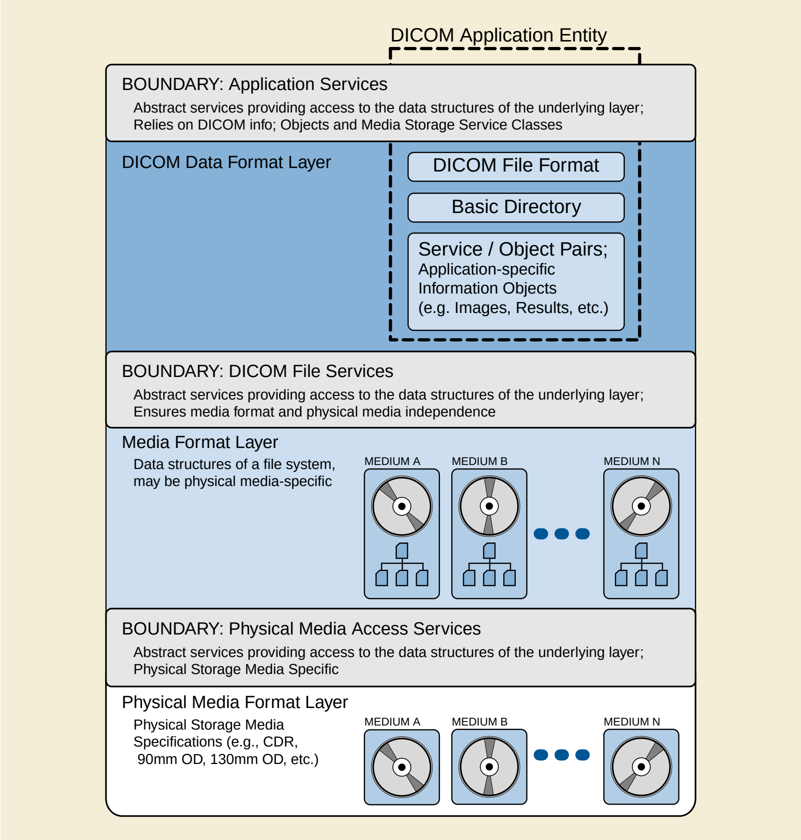

DICOM Application Entity
BOUNDARY: Application Services
Abstract services providing access to the data structures of the underlying layer;
Relies on DICOM info; Objects and Media Storage Service Classes
DICOM Data Format Layer
DICOM File Format
Basic Directory
Service / Object Pairs;
Application-specific
Information Objects
(e.g. Images, Results, etc.)
BOUNDARY: DICOM File Services
Abstract services providing access to the data structures of the underlying layer;
Ensures media format and physical media independence
Media Format Layer
Data structures of a file system,
may be physical media-specific
MEDIUM A
MEDIUM B
MEDIUM N
BOUNDARY: Physical Media Access Services
Abstract services providing access to the data structures of the underlying layer;
Physical Storage Media Specific
Physical Media Format Layer
Physical Storage Media
Specifications (e.g., CDR,
90mm OD, 130mm OD, etc.)
MEDIUM A
MEDIUM B
MEDIUM N

Figure 6.2-1. DICOM Media Storage Model

The DICOM Media Storage Model includes three layers, which are described in the following sections.

6.2.1 Physical Media Layer

Physical media characteristics are defined at the Physical Media Layer. Such characteristics include the physical media form factor, dimension, mechanical characteristics and recording properties. This Layer also defines the organization and grouping of the recorded bits.

Note

1. An example of a Physical Media Layer in the personal computer environment is the 3 1/2 inch floppy disk, double sided, high density.

2. The specification of one or more specific Physical Media for a given application is beyond the scope of this Part of the DICOM Standard. PS3.12 and its annexes specify several Physical Media choices. PS3.11 defines a number of Application Profiles that select specific Physical Media depending on the requirements of specific medical imaging applications.

6.2.2 Media Format Layer

At the Media Format Layer, Physical Media bit streams are organized into specific structures. Data file structures and associated directory structures are defined to allow efficient access and management of the physical media space.

Note

This layer is often specific to a given operating system environment. An example of such a Media Format Layer definition associated with the 3 1/2 inch floppy disk are the data structures used by the operating systems of various personal computer file systems. PS3.12 and its annexes specify several Media Format choices.

Media Formats supported by the DICOM Standard are selected to support the minimum requirements specified by the DICOM File Service as specified in Section 8 of this Part. Constraining access to the File content through such a DICOM File Service ensures that the DICOM Data Format Layer is independent from Media Format and Physical Media selection.

6.2.3 DICOM Data Format Layer

The DICOM Data Format Layer includes the following elements of specification:

a. DICOM Media Storage SOP Classes and associated Information Object Definitions;

b. The DICOM File Format;

c. The Secure DICOM File Format;

d. The DICOM Media Storage Directory SOP Class;

e. DICOM Media Storage Application Profiles;

f. DICOM Security Profiles for Media Storage.

6.2.3.1 DICOM SOP Classes

DICOM SOP Classes and associated Information Object Definitions (IODs) are used to convey specific medical imaging information at the Data Format Layer. Examples of such IODs are modality images, patient information, results, etc.

The use of DICOM IODs in conjunction with Media Storage Services forms a number of Media Storage Service Object Pair Classes or SOP Classes. Media Storage Services (e.g., read, write, delete, etc.) shall be performed through the DICOM File Service. The content of the resulting DICOM Files shall be formatted according to the DICOM File Format as specified below.

PS3.4 defines a number of SOP Classes that may be used for Media Storage in Annex I. These SOP Classes are based on DICOM Standard IODs that may be found in PS3.3.

The structure and encoding of a Data Set representing the data associated with a SOP Class shall follow PS3.5. The specification of Transfer Syntaxes that may be used to encode such a Data Set, is also defined in PS3.5.

6.2.3.2 Concept of the DICOM File Format

The encapsulation of a DICOM Data Set in a File shall follow the specifications of Section 7 of this Part. These encapsulation rules define a DICOM File Format able to contain in a File any DICOM Data Set. Files are identified by File IDs. No semantics shall be inferred from these File IDs, nor from their structure.

Note

A medical imaging application acting as a creator of a DICOM File may use semantic information to generate a File ID, but readers of DICOM files should not rely on apparent semantic content of a File ID.

Data Set encapsulation shall be based on the DICOM File Service as specified in Section 8 of this Part.

Note

It is acceptable that a specific Media Format offers more file services than those specified in the DICOM File Service. Such services may be local or internal to an implementation. Their usage is beyond the scope of the DICOM Standard. However, in cases where such services are reflected in the file structures of the Media format Layer or in the Data Set encoding of an Information Object, the extension of such services in a manner that jeopardizes interoperability should not be done (e.g., File IDs longer than those specified in the DICOM File Service).

The encapsulation of a DICOM File in a Secure DICOM File shall follow the specifications of Section 7.4 of this Part. These encapsulation rules define a mechanism for creating a Secure DICOM File by encapsulating an unprotected DICOM File as payload within a secure envelope.

6.2.3.3 DICOM Medical Information Directory

In addition to the DICOM Image and Image related SOP Classes (e.g., results, patients) other SOP Classes tailored for media storage may be used to provide references (or directories) based on medical information, thus facilitating access to the clinical imaging information. Such a SOP Class is the Media Storage Directory SOP Class as defined in PS3.4. Instances of this SOP Class are conveyed in the File with a File ID of DICOMDIR.

6.2.4 DICOM Media Storage Application Profiles

A Media Storage Application Profile defines a selection of choices at the various layers of the DICOM Media Storage Model that are applicable to a specific need or context in which the media interchange is intended to be performed. Such choices are formally specified as a Media Storage Application Profile in order to ensure interoperability between implementations conforming to the same Media Storage Application Profile. It facilitates conformance statements that allow users to assess interoperability of different implementations.

Media Storage Application Profiles shall include:

a. The description of the need addressed by the Application Profile (e.g., cardiac, echography, angiography) and its context of application;

b. The selection, at the Data Format Layer, of a number of specific IODs and associated SOP Classes. For standard DICOM SOP Classes, this shall be done by reference to PS3.4. These SOP Classes, like any other DICOM SOP Classes are assigned a unique registered UID. For each SOP Class it shall be stated if its support is required or optional within the context of this profile;

c. The selection of a specific Media Format definition. This is done by reference to PS3.12 that specify the selected Physical Medium, a specific associated Media Format and the mapping of this Media Format (or file system) services onto the DICOM File Service;

d. The selection of appropriate Transfer Syntaxes;

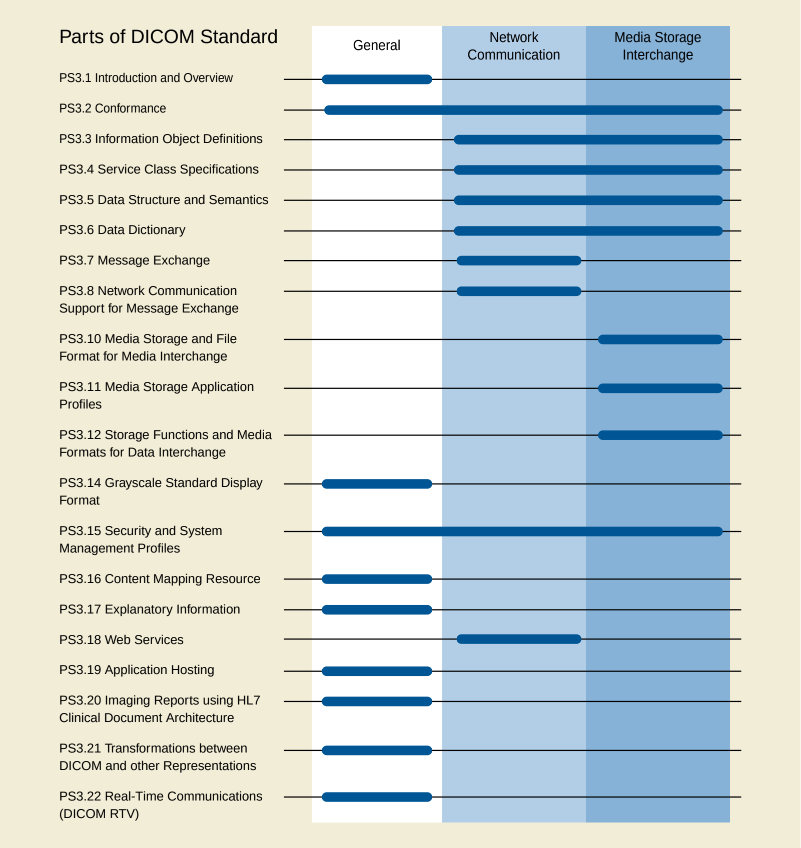
e. The selection of a specific Security Profile. This is done by reference to PS3.15 that specifies the cryptographic algorithms to be used to encapsulate the DICOM Files of the DICOM File Set into Secure DICOM Files. If a Media Storage Application Profile selects no Security Profile, then the Application Profile is unsecure and the Secure DICOM File Format shall not be used with that Application Profile;

f. Other choices facilitating interoperability such as specific limits (e.g., maximum file sizes, if necessary, support of options, if any).

The complete definition and structure of a Media Storage Application Profiles is specified by PS3.11. A number of Standard Application Profiles corresponding to different needs are included in PS3.11.

6.2.5 Media Storage and The DICOM Standard Structure

Figure 6.2-2 provides an overview of the relationship between the functional areas identified by the DICOM Media Storage Model introduced in Section 6.2 and the various Parts of the DICOM Standard related to Media Storage. A number of Parts of the DICOM Standard are common between Network Communication and Media Interchange.



Parts of DICOM Standard	General	Network Communication	Media Storage Interchange
PS3.1 Introduction and Overview	●		
PS3.2 Conformance	●	●	●
PS3.3 Information Object Definitions		●	●
PS3.4 Service Class Specifications		●	●
PS3.5 Data Structure and Semantics		●	●
PS3.6 Data Dictionary		●	●
PS3.7 Message Exchange		●	
PS3.8 Network Communication Support for Message Exchange		●	
PS3.10 Media Storage and File Format for Media Interchange			●
PS3.11 Media Storage Application Profiles			●
PS3.12 Storage Functions and Media Formats for Data Interchange			●
PS3.14 Grayscale Standard Display Format	●		
PS3.15 Security and System Management Profiles	●	●	●
PS3.16 Content Mapping Resource	●		
PS3.17 Explanatory Information	●		
PS3.18 Web Services		●	
PS3.19 Application Hosting	●		
PS3.20 Imaging Reports using HL7 Clinical Document Architecture	●		
PS3.21 Transformations between DICOM and other Representations	●		
PS3.22 Real-Time Communications (DICOM RTV)	●		

Figure 6.2-2. Media Storage and DICOM Parts

7 DICOM File Format

The DICOM File Format provides a means to encapsulate in a file the Data Set representing a SOP Instance related to a DICOM IOD. As shown in Figure 7-1, the byte stream of the Data Set is placed into the file after the DICOM File Meta Information. Each file contains a single SOP Instance.

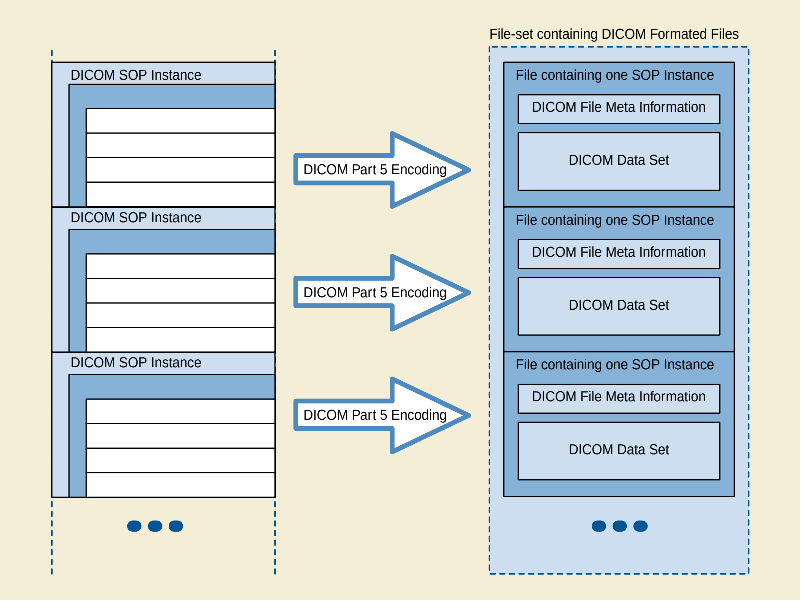

File-set containing DICOM Formated Files
DICOM SOP Instance
DICOM SOP Instance
DICOM SOP Instance
DICOM Part 5 Encoding
DICOM Part 5 Encoding
DICOM Part 5 Encoding
File containing one SOP Instance
DICOM File Meta Information
DICOM Data Set
File containing one SOP Instance
DICOM File Meta Information
DICOM Data Set
File containing one SOP Instance
DICOM File Meta Information
DICOM Data Set

Figure 7-1. File-set and File Format

7.1 DICOM File Meta Information

The File Meta Information includes identifying information on the encapsulated Data Set. This header consists of a 128 byte File Preamble, followed by a 4 byte DICOM prefix, followed by the File Meta Elements shown in Table 7.1-1. This header shall be present in every DICOM file.

The File Preamble is available for use as defined by Application Profiles or specific implementations. This Part of the DICOM Standard does not require any structure for this fixed size Preamble. It is not required to be structured as a DICOM Data Element with a Tag and a Length. It is intended to facilitate access to the images and other data in the DICOM file by providing compatibility with a number of commonly used computer image file formats. Whether or not the File Preamble contains information, the DICOM File content shall conform to the requirements of this Part and the Data Set shall conform to the SOP Class specified in the File Meta Information.

Note

1. If the File Preamble is not used by an Application Profile or a specific implementation, all 128 bytes shall be set to 00H. This is intended to facilitate the recognition that the Preamble is used when all 128 bytes are not set as specified above.

2. The File Preamble may for example contain information enabling a multi-media application to randomly access images stored in a DICOM Data Set. The same file can be accessed in two ways: by a multi-media application using the preamble and by a DICOM Application that ignores the preamble.

The four byte DICOM Prefix shall contain the character string "DICM" encoded as uppercase characters of the ISO 8859 G0 Character Repertoire. This four byte prefix is not structured as a DICOM Data Element with a Tag and a Length.

The Preamble and Prefix are followed by a set of DICOM Meta Elements with Tags and Lengths as defined in Table 7.1-1.

Table 7.1-1. DICOM File Meta Information

Attribute Name	Tag	Type	Attribute Description
File Preamble	*No Tag or Length Fields*	1	A fixed 128 byte field available for Application Profile or implementation specified use. If not used by an Application Profile or a specific implementation all bytes shall be set to 00H. File-set Readers or Updaters shall not rely on the content of this Preamble to determine that this File is or is not a DICOM File.
DICOM Prefix	*No Tag or Length Fields*	1	Four bytes containing the character string "DICM". This Prefix is intended to be used to recognize that this File is or is not a DICOM File.
File Meta Information Group Length	(0002,0000)	1	Number of bytes following this File Meta Element (end of the Value field) up to and including the last File Meta Element of the Group 2 File Meta Information
File Meta Information Version	(0002,0001)	1	This is a two byte field where each bit identifies a version of this File Meta Information header. In version 1 the first byte value is 00H and the second value byte value is 01H. Implementations reading Files with Meta Information where this attribute has bit 0 (lsb) of the second byte set to 1 may interpret the File Meta Information as specified in this version of PS3.10. All other bits shall not be checked. Note A bit field where each bit identifies a version, allows explicit indication of the support of multiple previous versions. Future versions of the File Meta Information that can be read by version 1 readers will have bit 0 of the second byte set to 1
Media Storage SOP Class UID	(0002,0002)	1	Uniquely identifies the SOP Class associated with the Data Set. SOP Class UIDs allowed for media storage are specified in Section I.4 “Media Storage SOP Classes” in PS3.4.
Media Storage SOP Instance UID	(0002,0003)	1	Uniquely identifies the SOP Instance associated with the Data Set placed in the file and following the File Meta Information.
Transfer Syntax UID	(0002,0010)	1	Uniquely identifies the Transfer Syntax used to encode the following Data Set. This Transfer Syntax does not apply to the File Meta Information. Note It is recommended to use one of the DICOM Transfer Syntaxes supporting explicit Value Representation encoding to facilitate interpretation of File Meta Element Values. JPIP Referenced Pixel Data Transfer Syntaxes are not used (see PS3.5).
Implementation Class UID	(0002,0012)	1	Uniquely identifies the implementation that wrote this file and its content. It provides an unambiguous identification of the type of implementation that last wrote the file in the event of interchange problems. It follows the same policies as defined by PS3.7 (association negotiation).
Implementation Version Name	(0002,0013)	3	Identifies a version for an Implementation Class UID (0002,0012) using up to 16 characters of the ISO 646:1990 (basic G0 set) repertoire. It follows the same policies as defined by PS3.7 (association negotiation).

Attribute Name	Tag	Type	Attribute Description
Source Application Entity Title	(0002,0016)	3	The DICOM Application Entity (AE) Title of the AE that wrote this file's content (or last updated it). If used, it allows the tracing of the source of errors in the event of media interchange problems. The policies associated with AE Titles are the same as those defined in PS3.8. Note If the Data Set was created de novo by the application writing the file, its AE Title, if it has one, may be used. If the Data Set was received over the network, there is potential ambiguity as to whether the value is the same as Sending Application Entity Title (0002,0017) or Receiving Application Entity Title (0002,0018) or some other value.
Sending Application Entity Title	(0002,0017)	3	The DICOM Application Entity (AE) Title of the AE that sent this file's content over a network. Note This is the AE that was the sender (source) of the content (the Data Set), in the case of a Data Set sent over the network (i.e., the Calling AET of the SCU for a C-STORE operation). If the Data Set was instead created de novo by the application writing the file, it should not be present.
Receiving Application Entity Title	(0002,0018)	3	The DICOM Application Entity (AE) Title of the AE that received this file's content over a network. Note This is the AE that was the recipient (destination) of the content (the Data Set), in the case of a Data Set received over the network (i.e., the Called AET of the SCP for a C-STORE operation). If the Data Set was instead created de novo by the application writing the file, it should not be present.
Source Presentation Address	(0002,0026)	3	The DICOM Presentation Address corresponding to the Source Application Entity Title (0002,0016). See Section 7.1.1.1.
Sending Presentation Address	(0002,0027)	3	The DICOM Presentation Address corresponding to the Sending Application Entity Title (0002,0017). See Section 7.1.1.1.
Receiving Presentation Address	(0002,0028)	3	The DICOM Presentation Address corresponding to the Receiving Application Entity Title (0002,0018). See Section 7.1.1.1.
Private Information Creator UID	(0002,0100)	3	The UID of the creator of the private information (0002,0102).
Private Information	(0002,0102)	1C	Contains Private Information placed in the File Meta Information. The creator shall be identified in (0002,0100). Required if Private Information Creator UID (0002,0100) is present.

Except for the 128 byte preamble and the 4 byte prefix, the File Meta Information shall be encoded using the Explicit VR Little Endian Transfer Syntax (UID=1.2.840.10008.1.2.1) as defined in DICOM PS3.5. Values of each File Meta Element shall be padded when necessary to achieve an even length, as specified in PS3.5 by their corresponding Value Representation. The Unknown (UN) Value Representation shall not be used in the File Meta Information. For compatibility with future versions of this Standard, any Tag (0002,xxxx) not defined in Table 7.1-1 shall be ignored.

Values of all Tags (0002,xxxx) are reserved for use by this Standard and later versions of DICOM. Data Elements with a group of 0002 shall not be used in Data Sets other than within the File Meta Information.

> Note
>
> PS3.5 specifies that Elements with Tags (0001,xxxx), (0003,xxxx), (0005,xxxx), and (0007,xxxx) shall not be used.

7.1.1 DICOM File Meta Information Attributes

7.1.1.1 Presentation Address Attributes

The encoding of the presentation address depends on the network transport protocol.

For objects exchanged using the PS3.8 DICOM Upper Layer Protocol for TCP/IP, the presentation address shall be encoded as a URI consisting of the scheme "dicom" followed by a colon, then either the fully qualified host name or IP address, followed by a colon and then the port number. E.g., "dicom:127.0.0.1:104", "dicom:myhost.mydomain.com:104".

For objects exchanged using the PS3.18 Web Services, the presentation address shall be encoded as the absolute URL of the endpoint of the base of the resource or service, sufficient to identify the system. E.g., "http://myhost.mydomain.com:80/wado-rs/". The presentation address is not expected to be the complete address of the resource. The scheme shall be "http", regardless of whether secure transport was actually used or not.

> Note
>
> For security reasons, care should be taken to assure that no access credentials such as usernames, passwords or authentication token parameters are encoded in the presentation address.

7.2 Data Set Encapsulation

Each File shall contain a single Data Set representing a single SOP Instance related to a single SOP Class (and corresponding IOD).

> Note
>
> A file may contain more than a single 2D image frame as specific IODs may be defined to include multiple frames.

The Transfer Syntax used to encode the Data Set shall be the one identified by the Transfer Syntax UID of the DICOM File Meta Information.

> Note
>
> 1. The Transfer Syntax used to encode the Data Set cannot be changed within the Data Set; i.e., the Transfer Syntax UID Data Element may not occur anywhere within the Data Set, e.g., nested within a Sequence Item.
> 2. A DICOM Data Set does not include its total length. The end of the file indication provided by the DICOM File Service (see Section 8.4) is the only indication of the end of the Data Set.

The last Data Element of a Data Set may be Data Element (FFFC,FFFC) if padding of a Data Set is desired when a file is written. The Value of this Data Set Trailing Padding Data Element (FFFC,FFFC) has no significance and shall be ignored by all DICOM implementations reading this Data Set. File-set Readers or Updaters shall be able to process this Data Set Trailing Padding (FFFC,FFFC) either in the Data Set following the Meta Information or in Data Sets nested in a Sequence (see PS3.5).

7.3 Support of File Management Information

The DICOM File Format does not include file management information in order to avoid duplication with functions related to the Media Format Layer. If necessary for a given DICOM Application Profile, the following information should be offered by the Media Format Layer:

a. File content owner identification;

b. File access statistics (e.g., date and time of creation);

c. Application file access control;

d. Physical media access control (e.g., write protect).

7.4 Secure DICOM File Format

A Secure DICOM File shall contain a single DICOM File encapsulated with the Cryptographic Message Syntax as defined in IETF STD 70 [RFC5652]. Depending on the cryptographic algorithms used for encapsulation, a Secure DICOM File can provide one or more the following security properties:

- Data Confidentiality (by means of encryption)
- Data Origin Authentication (by means of certificates and digital signatures)
- Data Integrity (by means of digital signatures)

In addition, a Secure DICOM File offers the possibility to communicate encryption keys and certificates to the intended recipients by means of key transport, key agreement or symmetric key-encryption key schemes.

7.5 Security Considerations for DICOM File Format

The DICOM File Format has a potential security vulnerability when the 128-byte File Preamble contains malicious executable content. Such malicious executable content may also refer to other malicious content in the file hidden within Data Elements of the File Meta Information or the Data Set.

Depending upon the use and purpose of a particular application it may be appropriate to:

- Sanitize the preamble, such as by:
  - Verifying that the preamble is:
    - all zeroes, or
    - begins with a valid magic number for recognized dual format content (e.g., TIFF or BigTIFF), or
    - contains other known safe content.
  - Clearing the preamble regardless of its content

    Note

    This will prevent use by applications that depend on the non-DICOM format, if the dual format capability has been used.
  - Testing explicitly for executable preamble contents.

    Note

    The proper response to the presence of executable content depends upon the purpose of the application, but generally, legitimate executable content will not be found in a DICOM File. A hypothetical example of an exception would be if the file contained its own executable viewer; this is sufficiently unlikely as to be not worth considering.
- Test explicitly for executable content anywhere within the DICOM File.
- Validate that the DICOM values, structures and content comply with the standard encoding rules and the IOD of the specified SOP Class, including Private Data Elements.

  Note

  Validation that Data Element Values comply with their Value Representation may partially mitigate the risk of hidden malicious content, but it may be necessary to remove or analyze the contents of opaque binary data in OB or other binary numeric value Data Elements, whether they be Standard or Private Data Elements. The VR of Private Data Elements may not be known. Without an executable preamble, such hidden content may not be directly executable, but may still serve as a repository of malicious code to be activated by some other accompanying exploit.

- Validate that the contents are of the appropriate SOP Classes.
- Validate that DICOM File Format files created for HTTP requests and responses do not contain such malicious content.

> Note
>
> For example, it may be appropriate for an archive that stores and retrieves PS3.10 Files to verify and validate both input and output, rather than store and retrieve files without checking the content.

The proper response to a validation failure depends upon the purpose of the application. Validation might be performed on input, output, or both.

> Note
>
> For example, an archive may choose to sanitize SOP Instances upon receipt, sanitize SOP Instances upon retrieval, validate the structure and fail storage requests for SOP Instances that fail validation, or other behavior based on the product purpose and the threat environment. This behavior is not specified by DICOM because the product purpose and the threat environment are highly dependent upon the application.

An implementation shall describe in its Conformance Statement its behavior with respect to sanitization of the preamble and any other validation performed.

8 DICOM File Service

The DICOM File Service specifies an abstract view of files from the point of view of a service user in the Data Format Layer. Constraining access to the content of files by Application Entities through such a DICOM File Service ensures independence of the Data Format Layer functions from specific Media Format and Physical Media selections.

Note

This DICOM File Service definition is abstract in the sense that it is only the specification of a boundary. Its focus is limited to the aspects directly related to the access to the data structures of the Media Format Layer (not the specifications of the data structures themselves). Even though the DICOM File Service may be described by means of a number of abstract primitives such as read, write, delete, etc., it is not intended to be the definition of an Application Programming Interface (API).

The DICOM File Service specified for Media Storage offers a basic service, simple enough to be supported by a wide range of commonly available Media Format (or file systems), but rich enough to provide the key functions to effectively manage files and access their content. The following sections specify the minimum mandatory requirements that shall be met by any physical media and associated media format to comply with the DICOM Media Storage model.

Note

It is acceptable that a specific Media Format offers more file services than those specified in the DICOM File Service. Such services may be internal to an implementation (i.e., not visible through the data structures on the Storage Media). Their usage is beyond the scope of the DICOM Standard. However, in cases where such services are reflected in the file structures of the Media format Layer or in the Data Set encoding an Information Object, the extension of such services in a manner that jeopardizes interoperability should not be done (e.g., File IDs longer than specified in the DICOM File Service).

8.1 File-set

The DICOM File Service offers the ability to create and access one or more files in a File-set. A File-set is a collection of files that share a common naming space within which File IDs (see Section 8.2) are unique. No semantics is attached to the order of Files within a File-set.

Note

1. The DICOM File Service does not require that Files within a File-set be simultaneously accessible (e.g., sequentially accessed media such as tapes are supported).
2. The DICOM File Service does not explicitly include the notion of distributing a File-set or a File across multiple "volumes/physical medium". However the transparent support by the Media Format Layer of such a feature is not precluded.

A File ID naming space (corresponding to a File-set) shall be associated with an appropriate feature of a Media Format defined structure. This mapping shall be specified in PS3.12 for each Media Format specification (this is integral to the specification of the relationship between any specific Media Format services and the DICOM File Services defined in this Part).

Note

An example of such a relationship is to map the File ID naming space to a volume in a personal computer Media Format or a partition in a workstation File System on a removable medium. Another example is to map the File ID naming space to a directory and its tree of sub-directories. In this case it could offer the possibility of supporting multiple File-sets (one per directory) on the same physical medium. Each File-set would have its own DICOMDIR File. To ensure interoperability, PS3.12 shall specify these specific mapping rules between the directories and the File ID naming space of a File-set (including the rules to unambiguously locate the DICOMDIR File).

A single File with the File ID DICOMDIR shall be included in each File-set.

Each File-set shall be uniquely identified by a File-set UID that shall be registered according to the UID registration rules specified in PS3.5. When Files are added or removed from a File-set, the File-set UID shall not change.

A File-set may also be identified by a File-set ID, which provides a simple (but possibly not globally unique) human readable reference. A File-set ID is string of zero (0) to sixteen (16) characters from the subset of the G0 repertoire of ISO 8859 (see Section 8.5). A File-set ID may be associated or mapped to an appropriate identifier at the Media Format Layer.

Note

1. Continuing with the personal computer Media Format example used first in the previous note, a File-set ID may be defined to be identical to a volume label.

2. Non-DICOM Files (Files with a content not formatted according to the requirements of this Part of the DICOM Standard) may be present in a File-set. Such files should not contain the DICOM File Meta Information specified in Table 7.1-1 and may not be referenced by the DICOM Media Storage Directory (see Section 8.6).

A File-set Descriptor File (a "readme" file) may also be attached to a File-set. See PS3.3 for a detailed specification of the Basic Directory IOD.

8.2 File IDs

Files are identified by a File ID that is unique within the context of a File-set. A File ID is an ordered sequence of File ID Components. A File ID may contain one to eight components. Each Component is a string of one to eight characters from a subset of the G0 repertoire of ISO 8859 (see Section 8.5)

Such a structure for File IDs (a sequence of components) allows the DICOM File Service to organize file selection in a hierarchical mode. No conventions are defined by the DICOM Standard for the use of the structure of File IDs components and their content (except for the reserved File ID DICOMDIR, see Section 8.6). Furthermore, no semantics shall be conveyed by the structure and content of such File IDs. This implies that when a File ID is assigned to any File in a File-set, the creating DICOM Application Entity may choose to structure the File ID as it wishes. Any other AE reading existing files or creating new files shall not be required to know any semantics the original creator may have associated with such a structure.

The File ID used to access a File through the abstract DICOM File Service is not necessarily the sole file identifier. The interchange Media Format (file system) may allow multiple file names to address the same physical file. Any use of alternate file names is beyond the scope of the DICOM Standard.

Note

1. A DICOM File ID is equivalent to the commonly used concept of "path name" concatenated with a "file name". An example of a valid DICOM File ID with four components shown separated by backslashes is:SUBDIR1\SUBDIR2\SUBDIR3\AB-CDEFGH

2. As specified in the DICOM Storage Media Model, no semantics is attached to File ID content and structure as it relates to the DICOM Information Objects stored in these files. If used, the hierarchical structure simply provides a means to organize the Files of a File-set and facilitate their selection.

3. The DICOM File Service does not specify any "separator" between the Components of the File ID. This is a Value Representation issue that may be addressed in a specific manner by each Media Format Layer. In DICOM IODs, File ID Components are generally handled as multiple Values and separated by "backslashes". There is no requirement that Media Format Layers use this separator.

4. DICOM files stored on interchange media may have an alternate file name or link that uses less restricted file names, such as a filename extension (e.g., ".dcm" in accordance with [RFC3240]).

8.3 File Management Roles and Services

When DICOM Application Entities participate in the exchange of information by the interchange of Storage Media, they perform through the DICOM File Service a number of Media Storage Services:

a. M-WRITE, to create new files in a File-set and assign them a File ID;

b. M-READ to read existing files based on their File ID;

c. M-DELETE to delete existing files based on their File ID;

d. M-INQUIRE FILE-SET to inquire free space available for creating new files within the File-set;

e. M-INQUIRE FILE to inquire date and time of file creation (or last update if applicable) for any file within the File-set.

A DICOM Application Entity may take one or more of the following three roles:

a. File-set Creator (FSC). Such an Application Entity, exercises this role by means of M-WRITE Operations to create the DICOMDIR File (see Section 8.6) and zero or more DICOM Files;

b. File-set Reader (FSR). Such an Application Entity, exercises this role by means of M-READ Operations to access one or more Files in a File-set. A File-set Reader shall not modify any of the files of the File-set (including the DICOMDIR File);

c. File-set Updater (FSU). Such an Application Entity, exercises this role by means of M-READ, M-WRITE, and M-DELETE Operations. It reads, but shall not modify, the content of any of the DICOM files in a File-set except for the DICOMDIR File. It may create additional Files by means of an M-WRITE or delete existing Files in a File-set by means of an M-DELETE.

Note

Although a File-set Updater (FSU) may include the functions corresponding to a File-set Creator (FSC) and a File-set Reader (FSR), it is not required that implementations supporting an FSU role also support an FSC or an FSR role.

The use of the concept of roles in DICOM Conformance Statements will result in a more precise expression of the capabilities of implementations supporting DICOM Media Storage. Conforming implementations shall support one of the following choices:

a. File-set Creator,

b. File-set Reader,

c. File-set Creator and File-set Reader,

d. File-set Updater,

e. File-set Updater and File-set Creator,

f. File-set Updater and File-set Reader,

g. File-set Updater, File-set Creator and File-set Reader.

Based on the roles supported by a DICOM Application Entity, the DICOM File Service shall support the Media Operations defined in Table 8.3-1.

Table 8.3-1. Media Operations and Roles

Media Operations Roles	M-WRITE	M-READ	M-DELETE	M-INQUIRE FILE-SET	M-INQUIRE FILE
FSC	Mandatory	*Not required*	*Not required*	Mandatory	*Not required*
FSR	*Not required*	Mandatory	*Not required*	*Not required*	Mandatory
FSC+FSR	Mandatory	Mandatory	*Not required*	Mandatory	Mandatory
FSU	Mandatory	Mandatory	Mandatory	Mandatory	Mandatory
FSU+FSC	Mandatory	Mandatory	Mandatory	Mandatory	Mandatory
FSU+FSR	Mandatory	Mandatory	Mandatory	Mandatory	Mandatory
FSU+FSC+FSR	Mandatory	Mandatory	Mandatory	Mandatory	Mandatory

Note

1. Media Preparation is outside the scope of this Part of the DICOM Standard. However it is assumed to be performed by the FS Creator.

2. The DICOM File Service does not require that file update capabilities (e.g., append) be supported by every Media Format Definition selected. The non-support of such file update capabilities to the DICOMDIR File may simply result in having to delete and create a new file in order to keep the directory information consistent.

3. If the content of a file needs to be updated or changed by an FSU, it is considered by this Part of the DICOM Standard as an M-DELETE Operation followed by an M-WRITE Operation. The FSU is responsible for ensuring the internal consistency of the File and its conformance to PS3.10 and the specific SOP Class stored, exactly as if the FSU was creating a new File. In particular, if an FSU implementation needs to update the file content but is not able to recognize and fully process the content of the File Preamble (see Section 7.1), it may consider setting the first four bytes of the Preamble to "DICM" followed by 124 bytes to 00H. This would avoid introducing inconsistencies between the content of the File Preamble and the remainder of the file content. An example of this situation may occur when a TIFF IFD 0 Offset in the File Preamble points at a further TIFF IFD embedded in the DICOM Data Set, and the update operation changes the location of this embedded TIFF IFD.

8.4 File Content Access

The DICOM File Service offers the ability to access the content of any File of a File-set. The File content is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the File.

Note

This File content definition as an ordered string of bytes is related to the view provided at the DICOM File Service level. It may not correspond to the physical ordering of bytes of data on a specific medium.

The DICOM File Service shall manage the delimitation of the end of the File by ensuring the user of the File Service that read access beyond the last byte will be detected and reported to the DICOM File Service user. This delimitation function is performed by the Media Format Layer.

The DICOM File Service shall offer the ability:

a. for an FSR or FSU to perform an M-READ to read zero or more bytes of the content of a File;

b. for an FSC or FSU to perform an M-WRITE to write one or more bytes making the content of a File.

Note

The DICOM File Service does not require any specific capability for the selective read access or write access of the content of a file (e.g., seek or append). However it does not restrict specific Media Format definitions to support such features.

8.5 Character Set

File IDs and File-set IDs shall be character strings made of characters from a subset of the G0 repertoire of ISO 8859. The following characters form this subset:

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z (uppercase)

1, 2, 3, 4, 5, 6, 7, 8, 9, 0 and _ (underscore)

Note

1. This is the character set defined for Code Strings (Value Representation CS - see PS3.5) except that SPACE is not included.

2. This character set is selected to limit characters in File IDs and File-set IDs to those that do not conflict with reserved characters and delimiters in the file systems defined in PS3.12. Component delimiters or other required demarcations defined in PS3.12 are not part of File IDs or File-set IDs

8.6 Reserved DICOMDIR File ID

A single File with a File ID, DICOMDIR, shall exist as a member of every File-set. This File ID is made of a single Component (see Section 8.2 for the File ID structure). It contains the DICOM Media Storage Directory (see PS3.3 for detailed specification of the Basic

Directory IOD), which includes general information about the whole File-set. This general information is always present, but optionally the directory content may be left empty in environments where it would not be needed. If the DICOMDIR File does not exist in a File-set, the File-set does not conform to PS3.10. The DICOMDIR shall not reference Files outside of the File-set to which it belongs.

Note

1. An example of the content of the DICOMDIR File may be found in Annex A.
2. If a Media Format specification in PS3.12 maps the origin of a File-set to a specific directory node in a specific file system, the File IDs, including the DICOMDIR File IDs, would be relative to this directory node path name.

The DICOMDIR File shall use the Explicit VR Little Endian Transfer Syntax (UID=1.2.840.10008.1.2.1) to encode the Media Storage Directory SOP Instance. The DICOMDIR File shall comply with the DICOM File Format specified in Section 7 of this Standard. In particular the:

a. SOP Class UID in the File Meta Information (header of the DICOMDIR File) shall have the Value specified in PS3.4 of this Standard for the Media Storage Directory SOP Class;

b. SOP Instance UID in the File Meta Information (header of the DICOMDIR File) shall contain the File-set UID Value. The File-set UID is assigned by the Application Entity that created the File-set (FSC role, see Section 8.3) with zero or more DICOM Files. This File-set UID Value shall not be changed by any other Application Entities reading or updating the content of the File-set.

Note

1. This policy reflects that a File-set is an abstraction of a "container" within which Files may be created or read. The File-set UID is related to the "container" not its content. A File-set in the DICOM File Service is intended to be mapped to a supporting feature of a selected Media Format (e.g., volume or partition).
2. The Standard does not prevent the making of duplicate copies of a File-set (i.e., a File-set with the same File-set UID). However, within a managed domain of File-sets, a domain specific policy may be used to prevent the creation of such duplicate File-sets.

9 Conformance Requirements

An implementation of PS3.10 shall:

a. have a Conformance Statement based on a PS3.11 Media Storage Application Profile in accordance with the framework defined in PS3.2, which will include addressing the Security Requirements defined in Section 7.5;

b. meet the requirements of the DICOM File Format as specified in Section 7;

c. support the DICOM File Service as specified in Section 8, in one or more of the roles identified in Section 8.3;

d. perform the Media Operations defined in Table 8.3-1 according to the role supported;

e. support the DICOMDIR File with a content as specified in the Media Storage Directory SOP Class in PS3.4.

A Example of DICOMDIR File Content (Informative)

This Annex provides an example of a File content that is based on selected aspects of the example introduced in PS3.3 for the Basic Directory Information Object. This is not a normative Annex. It is only an illustration, which is simply intended to help the reader better understand the organization of a DICOM Directory stored in a DICOMDIR File.

A.1 Simple Directory Content Example

Table A.1-1 shows in a simplified manner, the content of a simple DICOMDIR File. Values of elements are noted between square brackets (e.g., [1.2.840.10008.34.7.6]). Byte Offsets are shown by symbolic Values noted between brackets (e.g., {1493}).

Table A.1-1. Directory Content Example

	Meta-Info	128 bytes	File Preamble **[all bytes set to 00H]**
		4 bytes	DICOM Prefix **[DICM]**
		0002,0000	File Meta Information Group Length
		0002,0001	File Meta-Information Version **[0001]**
		0002,0002	Media Storage SOP Class UID **[1.2.840.10008.1.3.10]**
		0002,0003	Media Storage SOP Instance UID **[1.2.840.23856.36.45.3]**
		0002,0010	Transfer Syntax UID **[1.2.840.10008.1.1]**
		0002,0012	Implementation Class UID **[1.2.840.23856.34.90.3]**
		...	...
	File-set Identification	0004,1130	File-set ID **[EXAMPLE]**...
		...	
	General Directory Information	0004,1200	Offset of First Record of Root Directory Entity **{1688}**
		0004,1202	Offset of Last Record of Root Directory Entity **{6F18}**
		0004,1212	File-set Consistency Flag **[0000H]**
		...	...
		0004,1220	Directory Record Sequence.
			This Data Element Value includes the following Sequence of Items.
{1688}	**Item Tag**	FFFE,E000	Item Data Element (includes the following Data Elements)
	Patient A	0004,1400	Offset of the next Directory Record in Directory Entity {4624}
	Directory Record	0004,1410	Record In-use Flag **[FFFFH]**
		0004,1420	Offset of Referenced Lower Level Directory Entity **{1828}**
		...	...
		0004,1430	Directory Record Type **[PATIENT]**

	Selection Keys	0010,0010 0010,0020 ...	Patient's Name **[Patient A]** Patient ID **[123456789AB]** ....
	Item Delimitation Tag	FFFE,E00D	Item Delimitation Tag is present only if Item is of undefined length
{1828}	**Item Tag**	FFFE,E000	Item Data Element (includes the following Data Elements)
	Study 1 Directory Record	0004,1400 0004,1410 0004,1420 ...	Offset of the next Directory Record in Directory Entity (not shown in example) Record In-use Flag **[FFFFH]** Offset of Referenced Lower Level Directory Entity **{2300}** ...
		0004,1430	Directory Record Type **[STUDY]**
	Selection Keys	0020,000D 0020,0010 ...	Study Instance UID **[1.2.840.4656.23.4568745]** Study ID **[srt78UJ]** ....
	Item Delimitation Tag	FFFE,E00D	Item Delimitation Tag is present only if Item is of undefined length
{2300}	**Item Tag**	FFFE,E000	Item Data Element (includes the following Data Elements)
	Series 1 Directory Record	0004,1400 0004,1410 0004,1420 ...	Offset of the next Directory Record in Directory Entity (not shown in example) Record In-use Flag **[0FFFFH]** Offset of Referenced Lower Level Directory Entity **{2682}** ...
		0004,1430	Directory Record Type **[SERIES]**
	Selection Keys	0008,0060 0020,0011 ...	Modality **[NM]** Series Number **[2]** ...
	Item Delimitation Tag	FFFE,E00D	Item Delimitation Tag is present only if Item is of undefined length
{2682}	**Item Tag**	FFFE,E000	Item Data Element (includes the following Data Elements)
	Image 1 Directory Record	0004,1400 0004,1410 0004,1420 ...	Offset of the next Directory Record in Directory Entity **{3420}** Record In-use Flag **[FFFFH]** Offset of Referenced Lower Level Directory Entity **[00000000H]** ...
		0004,1430	Directory Record Type **[IMAGE]**
		0004,1500 0004,1510 0004,1511 0004,1512	Referenced File ID **[DIR\TDRI\3856G3]** Referenced SOP Class UID in File **[1.2.840.10008.5.1.4.1.1.20]** Referenced SOP Instance UID in File **[1.2.840.34.56.78999654.234]** Referenced Transfer Syntax UID in File **[1.2.840.10008.1.2.1]**

	Selection Keys	0008,0018 0020,0013 ...	Image SOP Instance UID **[1.2.840.34.56.78999654.234]** Image Number **[1]** ...
	Item Delimitation Tag	FFFE,E00D	Item Delimitation Tag is present only if Item is of undefined length
{3420}	**Item Tag**	FFFE,E000	Item Data Element (includes the following Data Elements)
	Image 2 Directory Record	0004,1400 0004,1410 0004,1420 ...	Offset of the next Directory Record in Directory Entity **[00000000H]** Record In-use Flag **[FFFFH]** Offset of Referenced Lower Level Directory Entity **[00000000H]**...
		0004,1430	Directory Record Type **[IMAGE]**
		0004,1500 0004,1510 0004,1511 0004,1512	Referenced File ID **[DIR\TDRI\3856G7]** Referenced SOP Class UID in File **[1.2.840.10008.5.1.4.1.1.20]** Referenced SOP Instance UID in File **[1.2.840.34.56.78999654.235]** Referenced Transfer Syntax UID in File **[1.2.840.10008.1.2.2]**
	Selection Keys	0008,0018 0020,0013 ...	Image SOP Instance UID **[1.2.840.34.56.78999654.235]** Image Number **[2]** ...
	Item Delimitation Tag	FFFE,E00D	Item Delimitation Tag is present only if Item is of undefined length
{4624}	**Item Tag**	FFFE,E000	Item Data Element (includes the following Data Elements)
	Patient B Directory Record	0004,1400 0004,1410 0004,1420 ...	Offset of the next Directory Record in Directory Entity **{6F18}** Record In-use Flag **[FFFFH]** Offset of Referenced Lower Level Directory Entity **{5012}** ...
		0004,1430	Directory Record Type **[PATIENT]**
	Selection Keys	0010,0010 0010,0020 ...	Patient's Name **[Patient B]** Patient ID **[23456789ABC]** ....
	Item Delimitation Tag	FFFE,E00D	Item Delimitation Tag is present only if Item is of undefined length
{5012}	**Item Tag**	FFFE,E000	Item Data Element (includes the following Data Elements)
	Study 1 Directory Record ...		
... (Sequence Items for the rest of the subordinate Directory Records)			
{6F18}	**Item Tag**	FFFE,E000	Item Data Element (includes the following Data Elements)

	Patient C Directory Record	0004,1400 0004,1410 0004,1420 0004,1430 ...	Offset of the next Directory Record in Directory Entity **{00000000H}** Record In-use Flag **[FFFFH]** Offset of Referenced Lower Level Directory Entity **{...}** Directory Record Type **[PATIENT]** ...
	Selection Keys	0010,0010 0010,0020	Patient Name **[Patient C]** Patient ID **[34567890ABC]** ....
	Item Delimitation Tag	FFFE,E00D	Item Delimitation Tag is present only if Item is of undefined length
	Sequence Delimitation Tag	FFFE,E0DD	Used only if the Directory Record Sequence (0004,1220) is of undefined length to delimit the end of the Value of the Directory Record Sequence Data Element.
... (Sequence Items for the rest of the subordinate Directory Records)			

A.2 Example of DICOMDIR File Content With Multiple Referenced Files

This section was previously defined in DICOM. It is now retired. See PS3.3-1998.

B HL7 Structured Document Files

Structured Documents as defined by an HL7 standard may be stored on DICOM Interchange Media, and may be referenced from within DICOM SOP Instances (including the DICOMDIR Media Storage Directory).

An Encapsulated CDA is referenced from the Media Storage Directory like any other DICOM SOP Instance.

An HL7 Structured Document is an aggregate multimedia object, consisting of a base XML-encoded document, plus zero or more multimedia components (e.g., graphics) that are considered an integral part of the object. The multimedia components shall be encoded in-line in the encapsulated XML document unless they are references to other DICOM SOP Instances contained on the media.