NEMA PS*3.5 93 6470247 0505440 797

NEMA STANDARDS PUBLICATION NO. PS 3.5

nema

Digital Imaging and Communications in Medicine (DICOM)

Part 5: Data Structures and Encoding

NEMA

NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION ■ 2101 L STREET, N.W., WASHINGTON, D.C. 20037

Provided by IHS
No reproduction or networking permitted without license from IHS
Not for Resale

NEMA Standards Publication No. PS 3.5-1993

Digital Imaging and Communications in Medicine (DICOM)
Part 5: Data Structures and Encoding

Approved by:
The American College of Radiology

Published by

National Electrical Manufacturers Association
2101 L Street, N.W.
Washington, DC 20037

© 1994 by National Electrical Manufacturers Association

Provided by IHS
No reproduction or networking permitted without license from IHS
Not for Resale

© NEMA PS 3.5-1993

CONTENTS

Page

Foreword 3

Clause

1 Scope and field of application 4
2 Normative references 4
3 Definitions 6
3.1 Reference model definitions 6
3.2 ACSE service definitions 6
3.3 Presentation service definitions 6
3.4 Object identification definitions 6
3.5 DICOM introduction and overview definitions 6
3.6 DICOM conformance definitions 7
3.7 DICOM information object definitions 7
3.8 DICOM service class specifications definitions 7
3.9 DICOM network communication support for message exchange definitions 7
3.10 DICOM data structures and encoding definitions 7
4 Symbols and abbreviations 11
5 Conventions 11
6 Value encoding 12
6.1 Support of character repertoires 12
6.2 Value representation (VR) 15
6.3 Enumerated values and defined terms 15
6.4 Value multiplicity (vm) and delimitation 20
7 The data set 21
7.1 Data elements 21
7.2 Group length 25
7.3 Big endian vs. Little endian byte ordering 25
7.4 Data element type 26
7.5 Nesting of data sets 27
7.6 Repeating groups 30
7.7 Retired data elements 30
7.8 Private data elements 31
8 Encoding of pixel data 32
8.1 Pixel data encoding of related data elements 32
8.2 Native or encapsulated format encoding 33
9 Unique identifiers (UIDs) 34
9.1 UID encoding rules 35
9.2 Unique identifier registration 35
10 Transfer syntax 35
10.1 DICOM default transfer syntax 36
10.2 Transfer syntax for a DICOM default of lossless JPEG compression 36
10.3 Transfer syntaxes for a DICOM defaults of lossy JPEG compression 37

Provided by IHS
No reproduction or networking permitted without license from IHS
Not for Resale

Annexes

Annex A (Normative) Transfer syntax specifications 38
A.1 DICOM implicit VR little endian transfer syntax 38
A.2 DICOM little endian transfer syntax (explicit VR) 38
A.3 DICOM big endian transfer syntax (explicit VR) 39
A.4 Transfer syntaxes for encapsulation of encoded pixel data 40
Annex B (Informative) Creating a privately defined unique identifier 45
Annex C (Informative) DICOM unique identifier registration process 47
Annex D (Informative) Examples of various pixel data and overlay encoding schemes 49
Annex E (Normative) DICOM default character repertoire 50
Annex F (Informative) Encapsulated images as part of a DICOM message 51
F.1 Encapsulated JPEG encoded images 51
Annex G (Informative) Index to data element tags 57
Annex H (Informative) Index to transfer syntax UIDs 58

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

ACR-NEMA

Digital Imaging and Communications in Medicine
Part 5: Data Structures and Encoding

FOREWORD

The American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA) formed a joint committee to develop a standard for Digital Imaging and Communications in Medicine (DICOM). This DICOM Standard was developed according to the NEMA procedures.

This standard is developed in liaison with other standardization organizations including CEN TC251 in Europe and JIRA in Japan, with review also by other organizations including IEEE, HL7 and ANSI in the USA.

The DICOM Standard is structured as a multi-part document using the guidelines established in the following document:

-ISO/IEC Directives, 1989 Part 3: Drafting and Presentation of International Standards.

This document is one part of the DICOM Standard which consists of the following parts:

PS 3.1: Introduction and Overview
PS 3.2: Conformance
PS 3.3: Information Object Definitions
PS 3.4: Service Class Specifications
PS 3.5: Data Structures and Encoding
PS 3.6: Data Dictionary
PS 3.7: Message Exchange
PS 3.8: Network Communication Support for Message Exchange
PS 3.9: Point-to-Point Communication Support for Message Exchange

These parts are related but independent documents. Their development level and approval status may differ. Additional parts may be added to this multi-part standard. Part PS 3.1 should be used as the base reference for the current parts of this Standard.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Digital Imaging and Communications in Medicine
Part 5: Data Structures and Encoding

1 Scope and field of application

This part of the DICOM Standard is Part 5 (PS 3.5) of a multi-part standard produced to facilitate the interchange of information between digital imaging computer systems in medical environments. This interchange will enhance diagnostic imaging and potentially other clinical applications. The multi-part DICOM Standard covers the protocols and data that shall be supplied to achieve this interchange of information.

In this part of the standard the structure and encoding of the Data Set is specified. In the context of Application Entities communicating over a network (see PS 3.7), a Data Set is that portion of a DICOM Message that conveys information about real world objects being managed over the network. A Data Set may have other contexts in other applications of this standard; e.g., in media exchange the Data Set might translate to file content structure.

This part of the DICOM Standard specifies:

a) the encoding of Values
b) the structure and usage of a Data Set
c) Data Element usage and relationships to other elements
d) the construction and usage of Nested Data Sets
e) the construction and usage of Data Sets containing Pixel Data
f) how to uniquely identify information
g) the specification of the standard DICOM Transfer Syntaxes

This part of the DICOM Standard does not specify:

a) the structure and syntax of a message (this is specified in PS 3.7)
b) the structure and usage of a command set (this is specified in PS 3.7)
c) how an application service functions or is classified (this is specified in PS 3.3 and PS 3.4)
d) how data sets relate to network communication, media storage, or other services

2 Normative references

The following standards contain provisions that, through references 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.

ANSI MSDS	Message Standard Developers Subcommittee PROPOSAL on Data Types
ANSI X3.4 - 1986	Coded Character Set - 7-Bit American National Standard Code for Information Interchange
ANSI X3.9 - 1978	Programming Language FORTRAN
ASTM E-1238-91	Standard Specification for Transferring Clinical Observations Between Independent Computer Systems; Draft Revision 4.2.1
IEEE 754:1985	32-bit and 64-bit Floating Point Number Representations

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

ISO 646:1990	Information Processing—ISO 7-bit coded character set for information interchange
ISO 2375:1986	Data Processing—Procedure for the registration of escape sequences
ISO 6429:1990	Information Processing—Control functions for 7-bit and 8-bit coded character sets
ISO 6523:1984	Data interchange—Structures for identification of organizations
ISO 7498:1984	Information processing systems—Open System Interconnection—Basic Reference Model
ISO 7498-4:1989	Information processing systems—Open Systems Interconnection—Part 4: Management Framework
ISO 8649:1988	Information processing systems—Open Systems Interconnection—Service definition for the Association Control Service Element (ACSE)
ISO 8822:1988	Information processing systems -- Open Systems Interconnection - Connection oriented presentation service definition
ISO/IEC 8824:1990	Information processing systems—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1)
ISO 8859-1:1987	Information processing—8-bit single-byte coded graphic character sets—Part 1: Latin alphabet No. 1
ISO 8859-2:1987	Information processing—8-bit single-byte coded graphic character sets—Part 2: Latin alphabet No. 2
ISO 8859-3:1988	Information processing—8-bit single-byte coded graphic character sets—Part 3: Latin alphabet No. 3
ISO 8859-4:1988	Information processing—8-bit single-byte coded graphic character sets—Part 4: Latin alphabet No. 4
ISO 8859-5:1988	Information processing—8-bit single-byte coded graphic character sets—Part 5: Latin/Cyrillic alphabet
ISO 8859-6:1987	Information processing—8-bit single-byte coded graphic character sets—Part 6: Latin/Arabic alphabet
ISO 8859-7:1987	Information processing—8-bit single-byte coded graphic character sets—Part 7: Latin/Greek alphabet
ISO 8859-8:1988	Information processing—8-bit single-byte coded graphic character sets—Part 8: Latin/Hebrew alphabet
ISO 8859-9:1989	Information processing—8-bit single-byte coded graphic character sets—Part 9: Latin alphabet No. 5
ISO/IS 10918-1	JPEG Standard for digital compression and encoding of continuous-tone still images. Part 1—Requirements and implementation guidelines
ISO/IS 10918-2	DRAFT: JPEG Standard for digital compression and encoding of continuous-tone still images. Part 2—Testing
ENV 41 503:1990	Information systems interconnection—European graphic character repertoires and their coding
ENV 41 508:1990	Information systems interconnection—East European graphic character repertoires and their coding

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9834-3:1990	Part 3: Procedures for the Assignment of Object Identifier Component Values for Joint ISO-CCITT Use ISO/IEC Directives, 1989 Part 3 - Drafting and presentation of International Standards

3 Definitions

For the purposes of this standard, the following definitions apply.

3.1 *Reference model definitions*

This part of the standard makes use of the following terms defined in ISO 7498:

a) Application Entities

b) OSI Presentation Protocol

3.2 *ACSE service definitions*

This part of the standard makes use of the following terms defined in ISO 8649:

a) Association

3.3 *Presentation service definitions*

This part of the standard makes use of the following terms defined in ISO 8822:

a) Presentation Context

b) Presentation Data Value (PDV)

c) Transfer Syntax

d) Transfer Syntax Name

3.4 *Object identification definitions*

This part of the standard makes use of the following terms defined in ISO 8824:

a) OSI Object Identification

3.5 *DICOM introduction and overview definitions*

This part of the standard makes use of the following terms defined in PS 3.1:

a) Attribute

b) Command Element

c) Data Dictionary

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

3.6 *DICOM conformance definitions*

This part of the standard makes use of the following terms defined in PS 3.2:

a) Conformance Statement

3.7 *DICOM information object definitions*

This part of the standard makes use of the following terms defined in PS 3.3:

a) Attribute Tag

b) Information Entity

c) Information Object Definition (IOD)

d) Multi-Frame Image

3.8 *DICOM service class specifications definitions*

This part of the standard makes use of the following terms defined in PS 3.4:

a) Service-Object Pair (SOP) Class

3.9 *DICOM network communication support for message exchange definitions*

This part of the standard makes use of the following terms defined in PS 3.8:

a) DICOM Upper Layer Service

3.10 *DICOM data structures and encoding definitions*

The following definitions are commonly used in this Standard:

3.10.1 BASIC OFFSET TABLE

A table of pointers to individual frames of an encapsulated multi-frame image.

3.10.2 BIG ENDIAN

A form of byte ordering where multiple byte binary values are encoded with the most significant byte encoded first, and the remaining bytes encoded in decreasing order of significance.

3.10.3 CHARACTER REPERTOIRE

A finite set of different characters that is considered to be complete for a given purpose and is specified independently of their encoding (also referred to as a character set).

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

3.10.4 CODED CHARACTER SET

The character set along with the one to one relationship between each character of the set and its coded representation.

3.10.5 CONTROL CHARACTER

Character used within a text string to indicate formatting.

3.10.6 DATA ELEMENT

A unit of information as defined by a single entry in the data dictionary. An encoded Information Object Definition (IOD) Attribute that is composed of, at a minimum, three fields: a Data Element Tag, a Value Length, and a Value Field. For some specific Transfer Syntaxes, a Data Element also contains a VR Field where the Value Representation of that Data Element is specified explicitly.

3.10.7 DATA ELEMENT TAG

A unique identifier for a Data Element composed of an ordered pair of numbers (a Group Number followed by an Element Number).

3.10.8 DATA ELEMENT TYPE

Used to specify whether an Attribute of an Information Object Definition or an Attribute of a SOP Class Definition is mandatory, mandatory only under certain conditions, or optional. This translates to whether a Data Element of a Data Set is mandatory, mandatory only under certain conditions, or optional.

3.10.9 DATA SET

Exchanged information consisting of a structured set of Attribute values directly or indirectly related to Information Objects. The value of each Attribute in a Data Set is expressed as a Data Element. A collection of Data Elements ordered by increasing Data Element Tag number that is an encoding of the values of Attributes of a real world object.

3.10.10 DEFINED TERM

The Value of a Data Element is a Defined Term when the Value of the element may be one of an explicitly specified set of standard values, and these values may be extended by implementors.

3.10.11 ELEMENT NUMBER

The second number in the ordered pair of numbers that makes up a Data Element Tag.

3.10.12 ENUMERATED VALUE

The Value of a Data Element is an Enumerated Value when the value of the element must be one of an explicitly specified set of standard values, and these values shall not be extended by implementors.

3.10.13 GROUP NUMBER

The first number in the ordered pair of numbers that makes up a Data Element Tag.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

3.10.14 ITEM

A component of the Value of a Data Element that is of Value Representation Sequence of Items. An Item contains a Data Set.

3.10.15 ITEM DELIMITATION DATA ELEMENT

Used to mark the end of an Item of Undefined Length in a Sequence of Items. This is the last Data Element in an Item of Undefined Length.

3.10.16 LITTLE ENDIAN

A form of byte ordering where multiple byte binary values are encoded with the least significant byte encoded first; and the remaining bytes encoded in increasing order of significance.

3.10.17 NESTED DATA SET

A Data Set contained within a Data Element of another Data Set. Data Sets can be nested recursively. Only Data Elements with Value Representation Sequence of Items may, themselves, contain Data Sets.

3.10.18 PIXEL DATA

Graphical data (e.g., images or overlays) of variable pixel-depth encoded in the Pixel Data Element, with Value Representation OW or OB. Additional descriptor Data Elements are often used to describe the contents of the Pixel Data element.

3.10.19 PRIVATE DATA ELEMENT

Additional Data Element, defined by an implementor, to communicate information that is not contained in Standard Data Elements. Private Data elements have odd Group Numbers.

3.10.20 REPEATING GROUP

Standard Data Elements within a particular range of Group Numbers where elements that have identical Element Numbers have the same meaning within each Group (and the same VR, VM, and Data Element Type). Repeating Groups shall only exist for Curves and Overlay Planes (Group Numbers (50xx,eeee) and (60xx,eeee), respectively) and are a remnant of versions of this standard prior to V3.0.

3.10.21 RETIRED DATA ELEMENT

A Data Element that is unsupported beginning with Version 3.0 of this standard. Implementations may continue to support Retired Data Elements for the purpose of backward compatibility with versions prior to V3.0, but this is not a requirement of this version of the standard.

3.10.22 SEQUENCE DELIMITATION ITEM

Item used to mark the end of a Sequence of Items of Undefined Length. This Item is the last Item in a Sequence of Items of Undefined Length.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

3.10.23 SEQUENCE OF ITEMS (VALUE REPRESENTATION SQ)

A Value Representation for Data Elements that contain a sequence of Data Sets. Sequence of Items allows for Nested Data Sets.

3.10.24 STANDARD DATA ELEMENT

A Data Element defined in the DICOM Standard, and therefore listed in the DICOM Data Element Dictionary in PS 3.6.

3.10.25 TRANSFER SYNTAX (Standard and Private)

A set of encoding rules that allow Application Entities to unambiguously negotiate the encoding techniques (e.g., Data Element structure, byte ordering, compression) they are able to support, thereby allowing these Application Entities to communicate.

3.10.26 UNDEFINED LENGTH

The ability to specify an unknown length for a Data Element Value (of Value Representation SQ, OW, or OB) or Item. Data Elements and Items of Undefined Length are delimited with Sequence Delimitation Items and Item Delimiter Data Elements, respectively.

3.10.27 UNIQUE IDENTIFIER (UID)

A string of characters that uniquely identifies a wide variety of items; guaranteeing uniqueness across multiple countries, sites, vendors and equipment.

3.10.28 VALUE

A component of a Value Field. A Value Field may consist of one or more of these components.

3.10.29 VALUE FIELD

The field within a Data Element that contains the Value(s) of that Data Element.

3.10.30 VALUE LENGTH

The field within a Data Element that contains the length of the Value Field of the Data Element.

3.10.31 VALUE MULTIPLICITY (VM)

Specifies the number of Values contained in the Value Field of a Data Element.

3.10.32 VALUE REPRESENTATION (VR)

Specifies the data type and format of the Value(s) contained in the Value Field of a Data Element.

3.10.33 VALUE REPRESENTATION FIELD

The field where the Value Representation of a Data Element is stored in the encoding of a Data Element structure with explicit VR.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

4 Symbols and abbreviations

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

4.1 **ACR** American College of Radiology

4.2 **AE** Application Entity

4.3 **ANSI** American National Standards Institute

4.4 **CEN TC251** Comite Europeen de Normalisation - Technical Committee 251 - Healthcare Informatics

4.5 **DICOM** Digital Imaging and Communications in Medicine

4.6 **HISPP** Healthcare Information Standards Planning Panel

4.7 **HL7** Healthcare Industry Level 7 Interface Standards

4.8 **IEEE** Institute of Electrical and Electronics Engineers

4.9 **IOD** Information Object Definition

4.10 **ISO** International Standards Organization

4.11 **JIRA** Japanese Industry Radiology Apparatus

4.12 **MSDS** Healthcare Message Standard Developers Sub-Committee

4.13 **NEMA** National Electrical Manufacturers Association

4.14 **OSI** Open Systems Interconnection

4.15 **TCP/IP** Transmission Control Protocol/Internet Protocol

4.16 **UID** Unique Identifier

4.17 **SOP** Service-Object Pair

4.18 **VM** Value Multiplicity

4.19 **VR** Value Representation

5 Conventions

Word(s) are capitalized in this document (not headings) to help the reader understand that these word(s) have been previously defined in Section 3 of this document and are to be interpreted with that meaning.

The Data Element Tag is represented as (gggg,eeee), where gggg equates to the Group Number and eeee equates to the Element Number within that Group. The Data Element Tag is represented in hexadecimal notation as specified for each Data Element in PS 3.6.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

The notation XXXXH, where XXXX is one or more hexadecimal digits, and "H" is used to signify a hexadecimal number.

6 Value encoding

A Data Set is constructed by encoding the values of Attributes specified in the Information Object Definition (IOD) of a Real-World Object. The specific content and semantics of these Attributes are specified in Information Object Definitions (see PS 3.3). The range of possible data types of these values and their encoding are specified in this section. The structure of a Data Set, which is composed of Data Elements containing these values, is specified in Section 7.

Throughout this part, as well as other parts of the DICOM Standard, Tags are used to identify both specific Attributes and their corresponding Data Elements.

6.1 *Support of character repertoires*

Values that are text or character strings can be composed of graphic and control characters. The graphic character set, independent of its encoding, is referred to as a Character Repertoire. Depending on the native language context in which Application Entities wish to exchange data using the DICOM Standard, different Character Repertoires will be used. The Character Repertoires supported by DICOM are defined in the nine parts of ISO 8859.

6.1.1 REPRESENTATION OF ENCODED CHARACTER VALUES

As defined in the ISO Standards referenced in this section, byte values used for encoded representations of characters are represented in this section as two decimal numbers in the form column/row.

This means that the value can be calculated as (column * 16) + row, e.g., 01/11 corresponds to the value 27 (1BH).

NOTE— Two digit hex notation will be used throughout the remainder of this standard to represent character encoding. The column/row notation is used only within Section 6.1 to simplify any cross referencing with applicable ISO standards.

The byte encoding space is divided into four ranges of values:

C0 bytes from 00/00 to 01/15
G0 bytes from 02/00 to 07/15
C1 bytes from 08/00 to 09/15
G1 bytes from 10/00 to 15/15

C0 and C1 are control character sets, while G0 and G1 are graphic character sets. Only some control characters from the C0 set are used in DICOM (see Section 6.1.3), and characters from the C1 set shall not be used.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

6.1.2 GRAPHIC CHARACTERS

A Character Repertoire, or character set, is a collection of graphic characters specified independently of their encoding. In DICOM all references to Character Repertoires are made via the ISO registration number specified in ISO 2375 and are of the form `ISO-IR xxx.`

Many standards, including ISO 8859 (Parts 1-9), specify Coded Character Sets. Coded Character Sets are graphic character sets along with the one to one relationship between each character of the set and its coded representation.

6.1.2.1 Default character repertoire

The default repertoire for character strings in DICOM shall be the Basic G0 Set of the International Reference Version of ISO 646:1990 (ISO-IR 6). See Annex E for a table of the DICOM default repertoire and its encoding.

NOTE— This Basic G0 Set is identical with the common character set of the nine parts of ISO 8859.

6.1.2.2 Extension or replacement of the default character repertoire

If two DICOM Application Entities (AEs) desire to extend or replace the default repertoire, they exchange information about the new repertoire. For the exchange of this information the default repertoire shall be used.

For Data Elements with Value Representations of SH (Short String), LO (Long String), ST (Short Text), LT (Long Text), or PN (Person Name) the default character repertoire may be extended or replaced (these Value Representations are described in more detail in Section 6.2). If such an extension or replacement is used, the relevant "Specific Character Set" shall be defined as an attribute of the SOP Common Module (0008,0005) (see PS 3.3) and shall be stated in the Conformance Statement. PS 3.2 gives conformance guidelines.

NOTES— 1. Preferred repertoires as defined in ENV 41 503 and ENV 41 508 for the use in Western and Eastern Europe, respectively, are: ISO-IR 100, ISO-IR 101, ISO-IR 144, ISO-IR 126. See Section 6.1.2.3.
2. The definition of Asiatic character sets or any other character sets for the usage in DICOM awaits the involvement of the respective users and their standardization organization.
3. Information Object Definitions using different character sets cannot rely—per se— on lexical ordering or string comparison of data elements represented as character strings. These operations can only be carried out within a given character repertoire and not over repertoire boundaries.

6.1.2.3 Encoding of character repertoires

For the extension of the default repertoire one of the nine 8-bit single byte code tables of ISO 8859 shall be used in DICOM. The additional repertoires represent the right hand parts (G1 sets) of the ISO 8859 code table:

ISO-IR 100	Latin alphabet No. 1, supplementary set
ISO-IR 101	Latin alphabet No. 2, supplementary set
ISO-IR 109	Latin alphabet No. 3, supplementary set
ISO-IR 110	Latin alphabet No. 4, supplementary set
ISO-IR 144	Cyrillic supplementary set
ISO-IR 127	Arabic supplementary set

--`,,`,,`-`-`,,`,,`,`,,`---

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

ISO-IR 126 Greek supplementary set
ISO-IR 138 Hebrew supplementary set
ISO-IR 148 Latin alphabet No. 5, supplementary set

NOTES— 1. The code tables are split into a "left hand part" (bit combinations 02/00 to 07/14) and a "right hand part" (bit combinations 10/00 to 15/15). The left hand part represents the invariant default repertoire (ISO-IR 6), whereas the various extension repertoires are mapped onto the right hand part of the table.
2. If the default character set is used only the "left hand part" of the tables (bit combinations 02/00 to 07/14) apply.
3.The encoding of the "left hand part" of the tables is equivalent with that of ASCII (ANSI X3.4 - 1986).

Two characters of the invariant "left-hand part" of the code tables, SPACE and "\", have special significance in the DICOM Standard. The character SPACE, represented by bit combination 02/00, shall be used for the padding of Data Element Values that are character strings. The graphic character "\" (BACKSLASH), represented by the bit combination 05/12, shall only be used in character strings with Value Representations of ST and LT (see Section 6.2). Otherwise "\" is used as a separator for multiple valued Data Elements (see Section 6.4).

The character DELETE (bit combination 07/15) shall not be used in DICOM character strings.

Only one extension repertoire (G1 set) may be used per Data Set; DICOM does not support code extension techniques, i.e., switching to G2 or G3 sets.

NOTE— Considerations on the Handling of Unsupported Character Sets:
In DICOM, character sets are not negotiated between Application Entities but are indicated by a conditional attribute of the SOP Common Module. Therefore, implementations may be confronted with G1 character sets that are unknown to them. The machine should print or display such characters by replacing all G1 characters with the four characters "\nnn", where "nnn" is the three digit octal representation of each byte. An example of this for an ASCII based machine would be as follows:

Character String:	Günther
Encoded representation:	04/07 15/12 06/14 07/04 06/08 06/05 07/02
ASCII based machine:	G\374nther

6.1.3 CONTROL CHARACTERS

Textual data that is interchanged may require some formatting information. Control Characters are used to indicate formatting, but their use in DICOM is kept to a minimum since some machines may handle them inappropriately. ISO 646:1990 and ISO 6429:1990 define Control Characters. As shown in Table 6.1-1 below, only a subset of three control characters from the C0 set shall be used in DICOM for the encoding of control characters in text strings.

Table 6.1-1—DICOM control characters and their encoding

Acronym	Name	Coded Value
LF	Line Feed	00/10
FF	Form Feed	00/12
CR	Carriage Return	00/13

In text strings a new line shall be represented as CR LF.

Note— Some machines (such as UNIX based machines) may interpret LF (00/10) as a new line. In such cases, it is expected that the DICOM format is converted to the correct internal representation for that machine.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

6.2 *Value representation (VR)*

The Value Representation of a Data Element shall describe the data type and format of that Data Element's Value(s). PS 3.6 lists the VR of each Data Element by Data Element Tag.

Values with VRs constructed of character strings, except in the case of the VR UI, shall be padded with SPACE characters (20H, in the Default Character Repertoire) when necessary to achieve even length. Values with a VR of UI shall be padded with a single trailing NULL (00H) character when necessary to achieve even length. Values with a VR of OB shall be padded with a single trailing NULL byte value (00H) when necessary to achieve even length.

An individual Value, including padding, shall not exceed the Length of Value. For multi-valued fields see Section 6.4.

NOTE— The length of Value Representations for which the Character Repertoire can be extended or replaced are expressly specified in characters rather than bytes in Table 6.2-1. This is because the mapping from a character to the number of bytes used for that character's encoding may be dependent on the character set used.

6.3 *Enumerated values and defined terms*

The value of certain Data Elements may be chosen among a set of explicit Values satisfying its VR. These explicit Values are either Enumerated Values or Defined Terms and are specified in PS 3.3 and PS 3.4.

Enumerated Values are used when the specified explicit Values are the only Values allowed for a Data Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition.

NOTE— Patient Sex (0010, 0040) is an example of a Data Element having Enumerated Values. It is defined to have a Value that is either "M", "F", or "O" (see PS 3.3). No other Value shall be given to this Data Element.

Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value.

NOTE— Interpretation Status ID (4008,0212) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; "CREATED," "RECORDED," "TRANSCRIBED," or "APPROVED" (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Status IDs may be defined by the implementor.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Table 6.2-1—DICOM value representations

VR Name	Definition	Character Repertoire	Length of Value
AE Application Entity	A string of characters with leading and trailing spaces (20H) being non-significant. The value made of 16 spaces, meaning "no application name specified", shall not be used.	Default Character Repertoire excluding control characters LF, FF, and CR.	16 bytes maximum
AS Age String	A string of characters with one of the following formats—nnnD, nnnW, nnnM, nnnY; where nnn shall contain the number of days for D, weeks for W, months for M, or years for Y. Example: "018M" would represent an age of 18 months.	"0"-"9", "D", "W", "M", "Y" of Default Character Repertoire	4 bytes fixed
AT Attribute Tag	Ordered pair of 16-bit unsigned integers that is the value of a Data Element Tag. Example: A Value of 001800FFH represents Data Element Tag (0018,00FF)	Not applicable	4 bytes fixed
CS Code String	A string of characters with leading or trailing spaces (20H) being non-significant.	Uppercase characters, "0"-"9", the SPACE character, and underscore "_", of the Default Character Repertoire	16 bytes maximum
DA Date	A string of characters of the format yyyymmdd; where yyyy shall contain year, mm shall contain the month, and dd shall contain the day. This conforms to the ANSI HISPP MSDS Date common data type. Example: "19930822" would represent August 22, 1993. NOTES: 1. For reasons of backward compatibility with versions of this standard prior to V3.0, it is recommended that implementations also support a string of characters of the format yyyy.mm.dd for this VR. 2. See also DT VR in this table.	"0"-"9" of Default Character Repertoire NOTE: For reasons specified in the previous column, implementations may wish to support the "." character as well.	8 bytes fixed NOTE: For reasons specified in the previous columns, implementations may also wish to support a 10 byte fixed length as well.
DS Decimal String	A string of characters representing either a fixed point number or a floating point number. A fixed point number shall contain only the characters 0-9 with an optional leading "+" or "-" and an optional "." to mark the decimal point. A floating point number shall be conveyed as defined in ANSI X3.9, with an "E" or "e" to indicate the start of the exponent. Decimal Strings may be padded with leading or trailing spaces. Embedded spaces are not allowed.	"0"-"9", "+", "-", "E", "e", "." of Default Character Repertoire	16 bytes maximum

Table continued on next page

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Table 6.2-1 continued

DT Date Time	The Date Time common data type. Indicates a concatenated date-time ASCII string in the format: CCYYMMDDHHMMSS.FFFFFF&ZZZZ The components of this string, from left to right, are CC = Century, YY = Year, MM = Month, DD = Day, HH = Hour, MM = Minute, SS = Second, FFFFFF = Fractional Second, & = "+" or "-", and ZZZZ = Hours and Minutes of offset. &ZZZZ is an optional suffix for plus/minus offset from Coordinated Universal Time. A component that is omitted from the string is termed a null component. Trailing null components of Date Time are ignored. Non-trailing null components are prohibited, given that the optional suffix is not considered as a component. NOTE: For reasons of backward compatibility with versions of this standard prior to V3.0, many existing DICOM Data Elements use the separate DA and TM VRs. Standard and Private Data Elements defined in the future should use DT , when appropriate, to be more compliant with ANSI HISPP MSDS.	"0"-"9", "+", "-", "." of Default Character Repertoire	26 bytes maximum
FL Floating Point Single	Single precision binary floating point number represented in IEEE 754:1985 32-bit Floating Point Number Format.	not applicable	4 bytes fixed
FD Floating Point Double	Double precision binary floating point number represented in IEEE 754:1985 64-bit Floating Point Number Format.	not applicable	8 bytes fixed
IS Integer String	A string of characters representing an Integer in base-10 (decimal), shall contain only the characters 0 - 9, with an optional leading "+" or "-". It may be padded with leading and/or trailing spaces. Embedded spaces are not allowed. The integer, n, represented shall be in the range $-(2^{31} - 1) \leq n \leq (2^{31} - 1)$.	"0"-"9", "+", "-" of Default Character Repertoire	12 bytes maximum
LO Long String	A character string that may be padded with leading and/or trailing spaces. The character "\" shall not be present, as it is used as the delimiter between values in multiple valued data elements. The string shall not have control characters.	Default Character Repertoire and/or as defined by (0008,0005)	64 chars maximum (see NOTE in 6.2)
LT Long Text	A character string that may contain one or more paragraphs. It may contain the graphic character set and the control characters, CR, LF, and FF. It may be padded with trailing spaces, which may be ignored, but leading spaces are considered to be significant. Data Elements with this VR shall not be multi-valued and therefore the BACKSLASH "\" character may be used.	Default Character Repertoire and/or as defined by (0008,0005)	10240 chars maximum (see NOTE in 6.2)

Table continued on next page

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Table 6.2-1 continued

OB Other Byte String	A string of bytes where the encoding of the contents is specified by the negotiated Transfer Syntax. OB is a VR which is insensitive to Little/Big Endian byte ordering (see Section 7.3). The string of bytes shall be padded with a single trailing NULL byte value (00H) when necessary to achieve even length.	not applicable	see Transfer Syntax definition
OW Other Word String	A string of 16-bit words where the encoding of the contents is specified by the negotiated Transfer Syntax. OW is a VR which requires byte swapping within each word when changing between Little Endian and Big Endian byte ordering (see Section 7.3).	not applicable	see Transfer Syntax definition
PN Person Name	A character string encoded using a 5 component convention. The string may be padded with trailing spaces. The five components in their order of occurrence are: family name complex , given name complex , middle name, name prefix, name suffix. Any of the five components may be an empty string. The component delimiter shall be the caret "^" character (5EH). Delimiters are required for interior null components. Trailing null components and their delimiters may be omitted. Multiple entries are permitted in each component and are encoded as natural text strings, in the format preferred by the named person. This conforms to the ANSI HISPP MSDS Person Name common data type. Examples: Rev. John Robert Quincy Adams, B.A. M.Div. "Adams^John Robert Quincy^^Rev.^B.A. M.Div." [One family name; three given names; no middle name; one prefix; two suffixes.] Susan Morrison-Jones, Ph.D., Chief Executive Officer "Morrison-Jones^Susan^^^Ph.D., Chief Executive Officer" [Two family names; one given name; no middle name; no prefix; two suffixes.] John Doe "Doe^John" [One family name; one given name; no middle name, prefix, or suffix. Delimiters have been omitted for the three trailing null components.] NOTES: 1. This five component convention is also used by HL7 as defined in ASTM E-1238-91 and further specialized by the ANSI MSDS. 2. In typical American and European usage the first occurrence of "given name" would represent the "first name". The second and subsequent occurrences of the "given name" would typically be treated as a middle name(s). The "middle name" component is retained for the purpose of backward compatibility with existing standards.	Default Character Repertoire and/or as defined by (0008,0005) excluding control characters LF, FF, and CR.	64 chars maximum (see NOTE in 6.2)

Table continued on next page

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Table 6.2-1 continued

PN continued	...continued 3. The "Degree" component present in ASTM E-1238-91 is absorbed into the "Suffix" component. 4. The implementor should remain mindful of earlier usage forms which represented "given names" as "first" and "middle" and that translations to and from this previous typical usage may be required. 5. For reasons of backward compatibility with versions of this standard prior to V3.0, person names might be considered a single family name complex (single component without "^" delimiters).		
SH Short String	A character string that may be padded with leading and/or trailing spaces. The character "\" shall not be present, as it is used as the delimiter between values for multiple data elements. The string shall not have control characters.	Default Character Repertoire and/or as defined by (0008,0005)	16 chars maximum (see NOTE in 6.2)
SL Signed Long	Signed binary integer 32 bits long. Represents an integer, n, in the range $-(2^{31} - 1) \leq n \leq (2^{31} - 1)$	not applicable	4 bytes fixed
SQ Sequence of Items	Value is a Sequence of one or more Items, as defined in Section 7.5.	not applicable (see Section 7.5)	not applicable (see Section 7.5)
SS Signed Short	Signed binary integer 16 bits long in 2's complement form. Represents an integer n in the range $-(2^{15} - 1) \leq n \leq (2^{15} - 1)$.	not applicable	2 bytes fixed
ST Short Text	A character string that may contain one or more paragraphs. It may contain the graphic character set and the control characters, CR, LF, and FF. It may be padded with trailing spaces, which may be ignored, but leading spaces are considered to be significant. Data Elements with this VR shall not be multi-valued and therefore the BACKSLASH "\" character may be used.	Default Character Repertoire and/or as defined by (0008,0005)	1024 chars maximum (see NOTE in 6.2)
TM Time	A string of characters of the format hhmmss.frac; where hh contains hours (range "00" - "23"), mm contains minutes (range "00" - "59"), ss contains seconds (range "00" - "59"), and frac contains a fractional part of a second as small as 1 millionth of a second (range "000000" - "999999"). A 24 hour clock is assumed. Midnight can be represented by only "0000" since "2400" would violate the hour range. The string may be padded with trailing spaces. Leading and embedded spaces are not allowed. One or more of the components mm, ss, or frac may be unspecified as long as every component to the right of an unspecified component is also unspecified. If frac is unspecified the preceding "." may not be included. Frac shall be held to six decimal places or less to ensure its format conforms to the ANSI HISPP MSDS Time common data type.	"0"-"9", "." of Default Character Repertoire	16 bytes maximum

Table continued on next page

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Table 6.2-1 continued

TM continued	...continued Examples: 1. "070907.0705 " represents a time of 7 hours, 9 minutes and 7.0705 seconds. 2. "1010" represents a time of 10 hours, and 10 minutes. 3. "021 " is an invalid value. NOTES: 1. For reasons of backward compatibility with versions of this standard prior to V3.0, it is recommended that implementations also support a string of characters of the format hh:mm:ss.frac for this VR. 2. See also DT VR in this table.		
UI Unique Identifier (UID)	A character string containing a UID that is used to uniquely identify a wide variety of items. The UID is a series of numeric components separated by the period "." character. If a Value Field containing one or more UIDs is an odd number of bytes in length, the Value Field shall be padded with a single trailing NULL (00H) character to ensure that the Value Field is an even number of bytes in length. See Section 9 and Annex B for a complete specification and examples.	"0"-"9", "." of Default Character Repertoire	64 bytes maximum
UL Unsigned Long	Unsigned binary integer 32 bits long. Represents an integer n in the range $0 \leq n < 2^{32}$.	not applicable	4 bytes fixed
US Unsigned Short	Unsigned binary integer 16 bits long. Represents integer n in the range $0 \leq n < 2^{16}$.	not applicable	2 bytes fixed

6.4 *Value multiplicity (VM) and delimitation*

The Value Multiplicity of a Data Element specifies the number of Values that can be encoded i n the Value Field of that Data Element. The VM of each Data Element is specified explicitly in PS 3.6. If the number of Values that may be encoded in an element is variable, it shall be represented by two numbers separated by a dash; e.g., "1-10" means that there may be 1 to 10 Values in the element.

NOTE— Elements having a multiplicity of "S", which represented "single", in versions of this standard preceding V3.0, will have a multiplicity of "1" in this version of this standard.

When a Data Element has multiple Values, those Values shall be delimited as follows:

— For character strings, the BACKSLASH ("\") shall be used as a delimiter between Values.

NOTE— BACKSLASH ("\") is used as a delimiter between character string Values that are of fixed length as well as variable length.

— Multiple binary Values of fixed length shall be a series of concatenated Values without any delimiter.

Each string Value in a multiple valued character string may be of even or odd length, but the length of the entire Value Field (including "\" delimiters) shall be of even length. If padding is required to make the Value Field of even length, a single padding character shall be applied to the end of the Value Field (to the last Value).

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

NOTE— A padding character may need to be appended to a fixed length character string value in the above case.

Only the last UID Value in a multiple valued Data Element with a VR of UI shall be padded with a single trailing NULL (00H) character when necessary to ensure that the entire Value Field (including "\" delimiters) is of even length.

Data Elements with a VR of SQ, OW, or OB shall always have a Value Multiplicity of one.

7 The data set

A Data Set represents an instance of a real world Information Object. A Data Set is constructed of Data Elements. Data Elements contain the encoded Values of Attributes of that object. The specific content and semantics of these Attributes are specified in Information Object Definitions (see PS 3.3).

The construction, characteristics, and encoding of a Data Set and its Data Elements are discussed in this section. Pixel Data, Overlays, and Curves are Data Elements whose interpretation depends on other related elements.

7.1 *Data elements*

A Data Element is uniquely identified by a Data Element Tag. The Data Elements in a Data Set shall be ordered by increasing Data Element Tag Number and shall occur at most once in a Data Set.

NOTE— A Data Element Tag may occur again within Nested Data Sets (see Section 7.5).

Two types of Data Elements are defined:

— Standard Data Elements have an even Group Number that is not (0000,eeee), (0002,eeee), (0004,eeee), or (0006,eeee).

NOTE— Usage of these groups is reserved for DIMSE Commands (see PS 3.7) and DICOM File Formats.

— Private Data Elements have an odd Group Number that is not (0001,eeee), (0003,eeee), (0005,eeee), (0007,eeee), or (FFFF,eeee). Private Data Elements are discussed further in Section 7.8.

NOTE— Although similar or related Data Elements often have the same Group Number; a Data Group does not convey any semantic meaning beginning with DICOM Version 3.0.

A Data Element shall have one of three structures. Two of these structures contain the VR of the Data Element (Explicit VR) but differ in the way their lengths are expressed, while the other structure does not contain the VR (Implicit VR). All three structures contain the Data Element Tag, Value Length and Value for the Data Element. See Figure 7.1-1.

Implicit and Explicit VR Data Elements shall not coexist within a Data Set and Data Sets nested in it (see Section 7.5). Whether a Data Set uses Explicit or Implicit VR, among other characteristics, is determined by the negotiated Transfer Syntax (see Section 10 and Annex A).

NOTE— VRs are not contained in Data Elements when using DICOM Default Transfer Syntax (DICOM Implicit VR Little Endian Transfer Syntax).

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

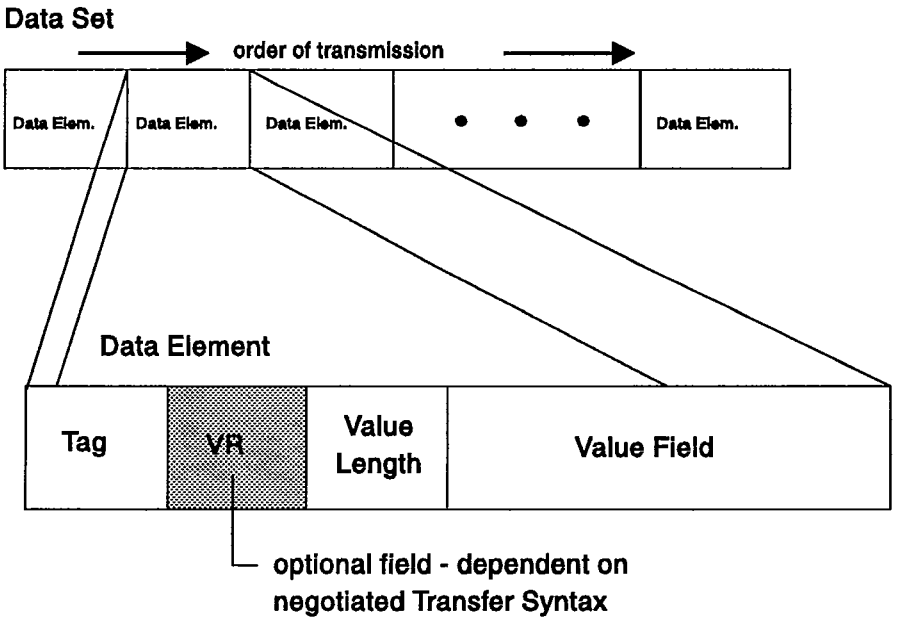


Figure 7.1-1—DICOM Data set and data element structures

7.1.1 DATA ELEMENT FIELDS

A Data Element is made up of fields. Three fields are common to all three Data Element structures; these are the Data Element Tag, Value Length, and Value Field. A fourth field, Value Representation, is only present in the two Explicit VR Data Element structures. The Data Element structures are defined in Sections 7.1.2. and 7.1.3. The definitions of the fields are:

Data Element Tag: An ordered pair of 16-bit unsigned integers representing the Group Number followed by Element Number.

Value Representation: A two-byte character string containing the VR of the Data Element. The VR for a given Data Element Tag shall be as defined by the Data Dictionary as specified in PS 3.6. The two character VR shall be encoded using characters from the DICOM default character set.

Value Length: Either:

— a 16 or 32-bit (dependent on VR and whether VR is explicit or implicit) unsigned integer containing the Explicit Length of the Value Field as the number of bytes (even) that make up the Value. It does not include the length of the Data Element Tag, Value Representation, and Value Length Fields.

— a 32-bit Length Field set to Undefined Length (FFFFFFFFH). Undefined Lengths may be used for Data Elements having the Value Representation (VR) Sequence of Items (SQ). For Data Elements with Value Representation OW or OB Undefined Length may be used depending on the negotiated Transfer Syntax (see Section 10 and Annex A).

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

NOTE— The decoder of a Data Set shall support both Explicit and Undefined Lengths for VRs of SQ and, when applicable, for VRs of OW and OB.

Value Field: An even number of bytes containing the Value(s) of the Data Element.

The data type of Value(s) stored in this field is specified by the Data Element's VR. The VR for a given Data Element Tag can be determined using the Data Dictionary in PS 3.6, or using the VR Field if it is contained explicitly within the Data Element. The VR of Standard Data Elements shall agree with those specified in the Data Dictionary.

The Value Multiplicity specifies how many Values with this VR can be placed in the Value Field. If the VM is greater than one, multiple values shall be delimited within the Value Field as defined previously in Section 6.4. The VMs of Standard Data Elements are specified in the Data Dictionary in PS 3.6.

Value Fields with Undefined Length are delimited through the use of Sequence Delimitation Items and Item Delimitation Data Elements which are described further in Section 7.5.

7.1.2 DATA ELEMENT STRUCTURE WITH EXPLICIT VR

When using the Explicit VR structures, the Data Element shall be constructed of four consecutive fields: Data Element Tag, VR, Value Length, and Value. Depending on the VR of the Data Element, the Data Element will be structured in one of two ways:

— for VRs of OB, OW, and SQ the 16 bits following the two character VR Field are reserved for use by later versions of the DICOM Standard. These reserved bytes shall be set to 0000H and shall not be used or decoded (Table 7.1-1). The Value Length Field is a 32-bit unsigned integer. If the Value Field has an Explicit Length, then the Value Length Field shall contain a value equal to the length (in bytes) of the Value Field. Otherwise, the Value Field has an Undefined Length and a Sequence Delimitation Item marks the end of the Value Field.

— for all other VRs the Value Length Field is the 16-bit unsigned integer following the two character VR Field (Table 7.1-2). The value of the Value Length Field shall equal the length of the Value Field.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Table 7.1-1—Data element with explicit VR of OB, OW, or SQ

Tag		VR		Value Length	Value
Group Number (16-bit unsigned integer)	Element Number (16-bit unsigned integer)	VR (2 byte character string) of "OB", "OW", or "SQ"	Reserved(2 bytes)set to a value of 0000H	32-bit unsigned integer	Even number of bytes containing the Data Element Value(s) encoded according to the VR and negotiated Transfer Syntax. Delimited with Sequence Delimitation Item if of Undefined Length.
2 bytes	2bytes	2bytes	2bytes	4 bytes	'Value Length' bytes if of Explicit Length

Table 7.1-2—Data element with explicit VR other than OB, OW or SQ

Tag		VR	Value Length	Value
Group Number (16-bit unsigned integer)	Element Number (16-bit unsigned integer)	VR (2 byte character string)Length	(16-bit unsigned integer)	Even number of bytes containing the Data Element Value(s) encoded according to the VR and negotiated Transfer Syntax.
2 bytes	2 bytes	2 bytes	2 bytes	'Value Length' bytes

7.1.3 DATA ELEMENT STRUCTURE WITH IMPLICIT VR

When using the Implicit VR structure the Data Element shall be constructed of three consecutive fields: Data Element Tag, Value Length, and Value (see Table 7.1-3). If the Value Field has an Explicit Length then the Value Length Field shall contain a value equal to the length (in bytes) of the Value Field. Otherwise, the Value Field has an Undefined Length and a Sequence Delimitation Item marks the end of the Value Field.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Table 7.1-3—Data element with implicit VR

Tag		Value Length	Value
Group Number (16-bit unsigned integer)	Element Number (16-bit unsigned integer)	32-bit unsigned integer	Even number of bytes containing the Data Elements Value encoded according to the VR specified in PS 3.6 and the negotiated Transfer Syntax. Delimited withSequence Delimitation Item if of Undefined Length.
2 bytes	2 bytes	4 bytes	'Value Length' bytes or Undefined Length

7.2 *Group length*

The Group Length (gggg,0000) Standard Data Element shall be an optional (Type 3 Data Element Type) Data Element in DICOM V3.0 (see Section 7.4 for a description of Data Element Types). It provides an optional optimization scheme in partial parsing of Data Sets by allowing the skipping of entire groups of Data Elements. Implementations may or may not choose to encode explicit Group Length in a Data Set. All implementations shall be able to accept or ignore such elements.

7.3 *Big endian versus little endian byte ordering*

Another component of the encoding of a Data Set that shall be agreed upon by communicating Application Entities is the Byte Ordering.

Little Endian byte ordering is defined as follows:

— In a binary number consisting of multiple bytes (e.g. a 32-bit unsigned integer value, the Group Number, the Element Number, etc.), the least significant byte shall be encoded first; with the remaining bytes encoded in increasing order of significance.

— In a character string consisting of multiple 8-bit single byte codes, the characters will be encoded in the order of occurrence in the string (left to right).

Big Endian byte ordering is defined as follows:

— In a binary number consisting of multiple bytes, the most significant byte shall be encoded first; with the remaining bytes encoded in decreasing order of significance.

— In a character string consisting of multiple 8-bit single byte codes, the characters will be encoded in the order of occurrence in the string (left to right).

Byte ordering is a component of an agreed upon Transfer Syntax (see Section 10). The default DICOM Transfer Syntax, which shall be supported by all AEs, uses Little Endian encoding and is specified in Annex A.1. Alternate Transfer Syntaxes, some of which use Big Endian encoding, are also specified in Annex A.

NOTE— The Command Set structure as specified in PS 3.7 is encoded using the Little Endian Implicit VR Transfer Syntax.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

In the default case of Little Endian encoding, Big Endian Machines interpreting Data Sets shall do 'byte swapping' before interpreting or operating on certain Data Elements. The Data Elements affected are all those having VRs that are multiple byte Values and that are not a character string of 8-bit single byte codes. VRs constructed of a string of characters of 8-bit single byte codes are really constructed of a string of individual bytes, and are therefore not affected by byte ordering. The VRs that are not a string of characters and consist of multiple bytes are:

— 2-byte US, SS, and OW

— 4-byte UL, SL, and FL

— 8 byte FD

NOTE— For the above VRs, the multiple bytes are presented in increasing order of significance when in Little Endian format. For example, an 8-byte Data Element with VR of FD, might be written in hexadecimal as 68AF4B2CH, but encoded in Little Endian would be 2C4BAF68H.

7.4 *Data element type*

An attribute, encoded as a Data Element, may or may not be required in a Data Set, depending on that Attribute's Data Element Type.

The Data Element Type of an Attribute of an Information Object Definition or an Attribute of a SOP Class Definition is used to specify whether that Attribute is mandatory or optional. The Data Element Type also indicates if an Attribute is conditional (only mandatory under certain conditions). The Data Element Types of Attributes of Composite IODs are specified in PS 3.3. The Data Element Types of Attributes of Normalized IODs are specified as Attributes of SOP Classes in PS 3.4.

7.4.1 TYPE 1 REQUIRED DATA ELEMENTS

IODs and SOP Classes define Type 1 Data Elements that shall be included and are mandatory elements. The Value Field shall contain valid data as defined by the elements VR and VM as specified in PS 3.6. The Length of the Value Field shall not be zero. Absence of a valid Value in a Type 1 Data Element is a protocol violation.

7.4.2 TYPE 1C CONDITIONAL DATA ELEMENTS

IODs and SOP Classes define Data Elements that shall be included under certain specified conditions. Type 1C elements have the same requirements as Type 1 elements under these conditions. It is a protocol violation if the specified conditions are met and the Data Element is not included.

When the specified conditions are not met, Type 1C elements shall not be included in the Data Set.

7.4.3 TYPE 2 REQUIRED DATA ELEMENTS

IODs and SOP Classes define Type 2 Data Elements that shall be included and are mandatory Data Elements. However, it is permissible that if a Value for a Type 2 element is unknown it can be encoded with zero Value Length and no Value. If the Value is known the Value Field shall contain that value as defined by the elements VR and VM as specified in PS 3.6. These Data Elements shall be included in the Data Set and their absence is a protocol violation.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

7.4.4 TYPE 2C CONDITIONAL DATA ELEMENTS

IODs and SOP Classes define Type 2C elements that have the same requirements as Type 2 elements under certain specified conditions. It is a protocol violation if the specified conditions are met and the Data Element is not included.

When the specified conditions are not met, Type 2C elements shall not be included in the Data Set.

NOTE— An example of a Type 2C Data Element is Inversion Time (0018,0082). For several SOP Class Definitions, this Data Element is required only if the Scanning Sequence (0018,0020) has the Value "IR." It is not required otherwise. See PS 3.3.

7.4.5 TYPE 3 OPTIONAL DATA ELEMENTS

IODs and SOP Classes define Type 3 Data Elements that are optional Data Elements. Absence of a Type 3 element from a Data Set does not convey any significance and is not a protocol violation. Type 3 elements may also be encoded with zero length and no Value. The meaning of a zero length Type 3 Data Element shall be precisely the same as that element being absent from the Data Set.

7.5 *Nesting of data sets*

The VR identified "SQ" shall be used for Data Elements with a Value consisting of a Sequence of zero or more Items, where each Item contains a set of Data Elements. SQ provides a flexible encoding scheme that may be used for simple structures of repeating sets of Data Elements, or the encoding of more complex Information Object Definitions often called folders. SQ Data Elements can also be used recursively to contain multi-level nested structures.

The definition of the Data Elements encapsulated in each Item is provided by the specification of the Data Element (or associated Attribute) of Value Representation SQ. Items in a sequence of Items may or may not contain the same set of Data Elements. Data Elements with a VR of SQ may contain multiple Items but shall always have a Value Multiplicity of one (ie. a single Sequence).

There are three special SQ related Data Elements that are not ruled by the VR encoding rules conveyed by the Transfer Syntax. They shall be encoded as Implicit VR. These special Data Elements are Item (FFFE,E000), Item Delimitation Item (FFFE,E00D), and Sequence Delimitation Item (FFFE,E0DD). However, the Data Set within the Value Field of the Data Element Item (FFFE ,E000) shall be encoded according to the rules conveyed by the Transfer Syntax.

7.5.1 ITEM ENCODING RULES

Each Item of a Data Element of Value Representation SQ shall be encoded as a DICOM Standard Data Element with a specific Data Element Tag of Value (FFFE,E000). The Item Tag is followed by a 4 byte Item Length field encoded in one of the following two ways:

a) Explicit Length: The number of bytes (even) contained in the Sequence Item Value (following but not including the Item Length Field) is encoded as a 32-bit unsigned integer value (see Section 7.1). This length shall include the total length of all Data Elements conveyed by this Item. This Item Length shall be equal to 00000000H if the Item contains no Data Set.

b) Undefined Length: The Item Length Field shall contain the value FFFFFFFFH to indicate an undefined Item length. It shall be used in conjunction with an Item Delimitation Data Element. This Item Delimitation Data Element has a Data Element Tag of (FFFE,E00D) and shall

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

follow the Data Elements encapsulated in the Item. No Value shall be present in the Item Delimitation Data Element and its Length shall be 00000000H.

The encoder of a Data Set may choose either one of the two ways of encoding. Both ways of encoding shall be supported by decoders of Data Sets. Data Element Tags (FFFF,eeee) are reserved by this standard and shall not be used.

Each Item Value shall contain a DICOM Data Set composed of Data Elements. Within the context of each Item, these Data Elements shall be ordered by increasing Data Element Tag value and appear only once (as Data Set is defined in Section 7.1). There is no relationship between the ordering of the Data Elements contained within an Item and the ordering of the Data Element Tag of SQ Value Representation that contains that Item. One or more Data Elements in an Item may be of Value Representation SQ, thus allowing for recursion.

7.5.2 DELIMITATION OF THE SEQUENCE OF ITEMS

Delimitation of the last Item of a Sequence of Items, encapsulated in a Data Element of Value Representation SQ, shall be in one of the two following ways:

a) Explicit Length: The number of bytes (even) contained in the Data Element Value (following but not including the Data Element Length Field) is encoded as a 32-bit unsigned integer value (see Section 7.1). This length shall include the total length resulting from the sequence of zero or more items conveyed by this Data Element. This Data Element Length shall be equal to 00000000H if the sequence of Items contains zero Items.

b) Undefined Length: The Data Element Length Field shall contain a Value FFFFFFFFH to indicate an Undefined Sequence length. It shall be used in conjunction with a Sequence Delimitation Item. A Sequence Delimitation Item shall be included after the last Item in the sequence. Its Item Tag shall be (FFFE,E0DD) with an Item Length of 00000000H. No Value shall be present.

The encoder of a Sequence of Items may choose either one of the two ways of encoding. Both ways of encoding shall be supported by decoders of the Sequence of Items.

NOTE— *The Sequence Delimitation Item Tag (FFFE,E0DD) is different from the Item Delimitation Tag (FFFE,E00D) introduced above in that it indicates the end of a Sequence of Items whose Length was left undefined. If an undefined length Item is the last Item of a Sequence of Items of undefined length, then an Item Delimitation Tag will be followed by a Sequence Delimitation Tag.*

For an example of an SQ Data Element of Explicit Length encapsulating Items of Explicit Length see Table 7.5-1.

For an example of an SQ Data Element of Undefined Length encapsulating Items of Explicit Length see Table 7.5-2.

For an example of an SQ Data Element of Undefined Length encapsulating Items of both Explicit and Undefined Length see Table 7.5-3.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Table 7.5-1—Example of a data element with implicit VR defined as a sequence of items (VR = SQ) with three items of explicit length

Data Element Tag	Data Element Length	Data Element Value								
(gggg, eeee) with VR of SQ	00000F00H	First Item			Second Item			Third Item		
		Item Tag (FFFE, E000)	Item Length 0000 04F8H	Item Value Data Set	Item Tag (FFFE, E000)	Item Length 0000 04F8H	Item Value Data Set	Item Tag (FFFE, E000)	Item Length 0000 04F8H	Item Value Data Set
4 bytes	4 bytes	4 bytes	4 bytes	04F8H bytes	4 bytes	4 bytes	04F8H bytes	4 bytes	4 bytes	04F8H bytes

Table 7.5-2—Example of a data element with explicit VR defined as a sequence of items (VR = SQ) of undefined length, containing two items of explicit length

Data Element Tag	Value Representation		Data Element Length	Data Element Value							
(gggg, eeee) with VR of SQ	SQ	0000H Reserved	FFFF FFFFH undefined length	First Item			Second Item			Sequence Delimitation Item	
				Item Tag (FFFE, E000)	Item Length 98A5 2C68H	Item Value Data Set	Item Tag (FFFE, E000)	Item Length B321 762CH	Item Value Data Set	Seq. Delim. Tag (FFFE, E0DD)	Item Length 0000 0000H
4 bytes	2 bytes	2 bytes	4 bytes	4 bytes	4 bytes	98A5 2C68H bytes	4 bytes	4 bytes	B321 762CH bytes	4 bytes	4 bytes

NOTE— The Data Set within the Item Values in Table 7.5-2 have VRs Explicitly defined.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Table 7.5-3—Example of a data element with implicit VR defined as a sequence of items (VR = SQ) of undefined length, containing two items where one item is of explicit length and the other item is of undefined length

Data Element Tag	Data Element Length	Data Element Value									
		First Item			Second Item					Sequence Delimitation Item	
(gggg,eeee) with VR of SQ	FFFF FFFFH un-defined length	Item Tag (FFFE, E000	Item Length 0000 17B6H	Item Value Data Set	Item Tag (FFFE, E000)	Item Length FFFF FFFFH un-defined length	Item Value Data Set	Item Delim. Tag (FFFE, E00D)	Length 0000 0000H	Seq. Delim. Tag (FFFE, E0DD)	Item Length 0000 0000H
4 bytes	4 bytes	4 bytes	4 bytes	17B6H bytes	4 bytes	4 bytes	un-defined length	4 bytes	4 bytes	4 bytes	4 bytes

7.6 *Repeating groups*

Multiple Overlay Planes and Curves are often associated with a single Image (see PS 3.3). Standard Data Elements with even Group Numbers (5000-501E,eeee) represent Curves, while elements with even Group Numbers (6000-601E,eeee) represent Overlay Planes. Both of these ranges of Group numbers are known as Repeating Groups. This use of group numbers is a remnant of versions of this standard prior to V3.0 that associated a semantic meaning with particular Groups.

In each of these ranges of Group Numbers, Standard Data Elements that have identical Element Numbers have the same meaning within each Group (and the same VR, VM, and Data Element Type). The notation (50xx,eeee) and (60xx,eeee) are used for the Group Number in Data Element Tags when referring to a common Data Element across these groups (see PS 3.6). Groups (50xx,eeee) and (60xx,eeee) are called Repeating Groups because of these characteristics.

NOTE— Repeating Elements have been retired beginning with DICOM Version 3.0.

Repeating Groups shall only be allowed in the even Groups (6000-601E,eeee) and even Groups (5000-501E,eeee) cases. In the future, Data Elements with VRs of SQ shall be used to serve a similar purpose.

7.7 *Retired data elements*

Certain Data Elements are no longer supported beginning with Version 3.0 of this standard. These Data Elements are retired and are denoted as such (RET) in the VR column in Part 6 of the DICOM Standard. Implementations may continue to support these Data Elements for the purpose of backward compatibility with versions of this standard prior to V3.0, but this is not a requirement of this standard. If a retired Data Element is used it must contain valid data as specified in versions of this

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Data Element, and its associated Data Element Tag, is reserved by this standard. Retired Data Element Tags shall not be redefined in later versions of this standard.

7.8 *Private data elements*

Implementations may require communication of information that cannot be contained in Standard Data Elements. Private Data Elements are intended to be used to contain such information.

Private Data Elements have the same structure as Standard Data Elements specified earlier in Section 7.1 (i.e., Data Element Tag field, optional VR field, length field, and value field). The Group Number used in the Element Tag of Private Data Elements shall be an odd number. Private Data Elements shall be contained in the Data Set in increasing numeric order of Data Element Tag. The Value Field of a Private data element shall have one of the VRs specified by this standard in Section 6.2.

For each Information Object Definition or SOP Class Definition, certain Data Elements are required (Data Element Type 1, 1C, 2, or 2C) as specified in PS 3.3 and PS 3.4. Private Data Elements shall not be used in place of required Standard Data Elements.

7.8.1 PRIVATE DATA ELEMENT TAGS

It is possible that multiple implementors may define Private Elements with the same (odd) group number. To avoid conflicts, Private Elements shall be assigned Private Data Element Tags according to the following rules.

a) Private Creator Data Elements numbered (gggg,0010-00FF) (gggg is odd) shall be used to reserve a block of Elements with Group Number gggg for use by an individual implementor. The implementor shall insert an identification code in the first unused (unassigned) Element in this series to reserve a block of Private Elements. The VR of the private identification code shall be LO (Long String) and the VM shall be equal to 1.

b) Private Creator Data Element (gggg,0010), is a Type 1 Data Element that identifies the implementor reserving element (gggg,1000-10FF), Private Creator Data Element (gggg,0011) identifies the implementor reserving elements (gggg,1100-11FF), and so on, until Private Creator Data Element (gggg,00FF) identifies the implementor reserving elements (gggg,FF00-FFFF).

c) Encoders of Private Data Elements shall be able to dynamically assign private data to any available (unreserved) block(s) within the Private group, and specify this assignment through the blocks corresponding Private Creator Data Element(s). Decoders of Private Data shall be able to accept reserved blocks with a given Private Creator identification code at any position within the Private group specified by the blocks corresponding Private Creator Data Element.

NOTES— 1. The versions of this standard prior to V3.0 described shadow groups. These were groups with a group number one greater than the standard groups. Elimination of conflicts in Private Data Element Tags have made this distinction obsolete and this terminology has been retired.
2. The versions of this standard prior to V3.0 specified private group element numbers (gggg,10FF-7FFF) reserved for manufacturers and private group element numbers (gggg, 8100-FFFF) reserved for users. Elimination of conflicts in Private Data Element Tags has made this distinction obsolete and this specification has been retired.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

7.8.2 VR RULES FOR PRIVATE ELEMENTS

The Value Representations used for Private Data Elements shall be the same as those VRs specified for Standard Data Elements in Section 6.2.

8 Encoding of pixel data

The Pixel Data Element (7FE0,0010) shall be used for the exchange of encoded graphical image data. This element along with additional Data Elements, specified as Attributes of the Image Information Entities defined in PS 3.3, shall be used to describe the way in which the Pixel Data is encoded and shall be interpreted. Finally, depending on the negotiated Transfer Syntax (see Section 10 and Annex A), Pixel Data may be compressed.

The Pixel Data Element (7FE0,0010) has a VR of OW or OB, depending on the negotiated Transfer Syntax (see Annex A). The only difference between OW and OB being that OB, a string of bytes, shall be unaffected by Byte Ordering (see Section 7.3).

8.1 *Pixel data encoding of related data elements*

Encoded Pixel Data of various bit depths shall be accommodated. The following three Data Elements shall define the Pixel structure:

— Bits Allocated (0028,0100)

— Bits Stored (0028,0101)

— High Bit (0028,0102)

Bits Allocated (0028,0100) specifies the total number of bits allocated for the storage of a pixel sample. Bits Stored (0028,0101) defines the total number of these allocated bits that will be used to represent a pixel sample Value. Bits Stored (0028,0101) shall never be larger than Bits Allocated (0028,0100). High Bit (0028,0102) specifies where the high order bit of the Bits Stored (0028,0101) is to be placed in respect to the Bits Allocated (0028,0100) specification. Bits not used for pixel samples can be used for overlay planes described further in PS 3.3.

NOTE— For example, in Pixel Data with 16 bits (2 bytes) allocated, 12 bits stored, and bit 15 specified as the high bit, one pixel sample is encoded in each 16-bit word, with the 4 least significant bits of each word not containing Pixel Data. See Annex D for other examples of the basic encoding schemes.

Starting with DICOM Version 3.0, restrictions are placed on the number of Bits Allocated (0028,0100) and are specified in the Information Object Definitions in PS 3.3. Also, the Pixel Data Value Field, like all other Value Fields in DICOM, shall be an even number of bytes in length. This means that the Value Field may need to be padded with data that is not part of the image and shall not be considered significant.

The field of bits representing the value of a pixel sample shall be a binary 2's complement integer or an unsigned integer, as specified by the Data Element Pixel Representation (0028,0103). The sign bit shall be the High Bit in a pixel sample that is a 2's complement integer. The minimum actual Pixel Value encountered in the Pixel Data is specified by Smallest Image Pixel Value (0028,0106) while the maximum value is specified by Largest Image Pixel Value (0028,0107).

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

8.2 *Native or encapsulated format encoding*

Pixel data conveyed in the Pixel Data Element (7FE0,0010) may be sent either in a Native (uncompressed) Format or in an Encapsulated Format (e.g. compressed) defined outside the DICOM standard.

If sent in a Native Format, the Value Representation OW is used. The pixel samples are encoded as the direct concatenation of the bits of each pixel sample, where the number of bits of each sample is defined by the Bits Allocated Data Element Value (0028,0100). The byte ordering in the 2-byte words is defined by one of the Little Endian or Big Endian Transfer Syntaxes negotiated at the Association Establishment (see Annex A).

If sent in an Encapsulated Format (i.e. other than the Native Format) the Value Representation OB is used. The pixel samples are encoded according to the encoding process defined by one of the negotiated Transfer Syntaxes (see Annex A). The encapsulated pixel stream of encoded pixel data is segmented in one or more Fragments which convey their explicit length. The sequence of Fragments of the encapsulated pixel stream is terminated by a delimiter, thus allowing the support of encoding processes where the resulting length of the entire pixel stream is not known until it is entirely encoded. This Encapsulated Format supports both Single-Frame and Multi-Frame images (as defined in PS 3.3).

8.2.1 JPEG IMAGE COMPRESSION

DICOM provides a mechanism for supporting the use of JPEG Image Compression through the Encapsulated Format (see PS 3.3). Annex A defines a number of Transfer Syntaxes which reference the JPEG Standard and provide a number of lossless (bit preserving) and lossy compression schemes.

NOTE— The context where the usage of lossy compression of medical images is clinically acceptable is beyond the scope of the DICOM Standard. The policies associated with the selection of appropriate compression parameters (e.g. compression ratio) for JPEG lossy compression is also beyond the scope of this standard.

In order to facilitate interoperability of implementations conforming to the DICOM Standard which elect to use one or more of the Transfer Syntaxes for JPEG Image Compression, the following policy is specified:

— Any implementation which conforms to the DICOM Standard and has elected to support any one of the Transfer Syntaxes for lossless JPEG Image Compression, shall support the following lossless compression: The subset (first-order horizontal prediction [Selection Value 1) of JPEG Process 14 (DPCM, non-hierarchical with Huffman coding) (see Annex F).

— Any implementation which conforms to the DICOM Standard and has elected to support any one of the Transfer Syntaxes for 8-bit lossy JPEG Image Compression, shall support the JPEG Baseline Compression (coding Process 1).

— Any implementation which conforms to the DICOM Standard and has elected to support any one of the Transfer Syntaxes for 12-bit lossy JPEG Image Compression, shall support the JPEG Compression Process 4.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

NOTE— The DICOM conformance statement shall differentiate whether or not the implementation is capable of simply receiving or receiving and processing JPEG encoded images (see PS 3.2).

The use of the DICOM Encapsulated Format to support JPEG Compressed Pixel Data implies that the Data Elements which are related to the Native Format Pixel Data encoding (e.g. Bits Allocated, Bits Stored, High Bit, Pixel Representation, Rows, Columns, etc.) shall contain values which are consistent with the characteristics of the uncompressed pixel data from which the compressed Data Stream was derived. The Pixel Data characteristics included in the JPEG Interchange Format shall be used to decode the compressed data stream.

9 Unique identifiers (UIDs)

Unique Identifiers (UIDs) provide the capability to uniquely identify a wide variety of items. They guarantee uniqueness across multiple countries, sites, vendors and equipment. This scheme is used in DICOM to uniquely identify items such as SOP Classes, image instances, network negotiation parameters, etc.

The UID identification scheme is based on the OSI Object Identification (numeric form) as defined by the ISO 8824 standard. All Unique Identifiers, used within the context of the DICOM Standard, are registered values as defined by ISO 9834-3 to ensure global uniqueness. The uses of such UIDs are defined in the various Parts of the DICOM Standard.

Each UID is composed of two parts, an <org root> and a <suffix>:

UID = <org root>.<suffix>

The <org root> portion of the UID uniquely identifies an organization, (i.e., manufacturer, research organization, NEMA, etc.), and is composed of a number of numeric components as defined by ISO 8824. The <suffix> portion of the UID is also composed of a number of numeric components, and shall be unique within the scope of the <org root>. This implies that the organization identified in the <org root> is responsible for guaranteeing <suffix> uniqueness by providing registration policies. These policies shall guarantee <suffix> uniqueness for all UID's created by that organization. Unlike the <org root>, which may be common for UID's in an organization, the <suffix> shall take different unique values between different UID's that identify different objects.

The <org root> "1.2.840.10008" is reserved for DICOM defined items (such as DICOM Transfer Syntaxes) and shall not be used for privately defined items (such as an Image Instance).

Although a specific implementation may choose some particular structure for its generated UIDs, it should never assume that a UID carries any semantics. Thus, a UID shall not be "parsed" to find a particular value or component. Component definition (for the suffix) is implementation specific and may change as long as uniqueness is maintained. Parsing UID's may jeopardize the ability to inter-operate as implementations evolve.

Example of UID structure is given in Annex C.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

9.1 *UID encoding rules*

The DICOM UID encoding rules are defined as follows:

— Each component of a UID is a number and shall consist of one or more digits. The first digit of each component shall not be zero unless the component is a single digit.

NOTE— Registration authorities may distribute components with non-significant leading zeroes. The leading zeroes should be ignored when being encoded (ie. "00029" would be encoded "29").

— Each component numeric value shall be encoded using the characters 0-9 of the Basic G0 Set of the International Reference Version of ISO 646:1990 (the DICOM default character repertoire).

— Components shall be separated by the character "." (2EH).

— If ending on an odd byte boundary, except when used for network negotiation (See PS 3.8), one trailing NULL (00H), as a padding character, shall follow the last component in order to align the UID on an even byte boundary.

— UID's, shall not exceed 64 total characters, including the digits of each component, separators between components, and the NULL (00H) padding character if needed.

9.2 *Unique identifier registration*

Each UID used in DICOM shall be defined and registered in one of the following two ways:

— DICOM defined and registered UID's

— Privately defined and registered UID's

Both UIDs use the same encoding rules as defined in Section 9.1. See Annex C for a more detailed description of the UID registration process.

9.2.1 DICOM DEFINED AND REGISTERED UNIQUE IDENTIFIERS

A limited number of registered DICOM Defined UIDs are used within the DICOM Standard. The organization responsible for the definition and registration of such DICOM UIDs is NEMA. The registration process will rely on the publication of the DICOM Registered UIDs in PS 3.6.

9.2.2 PRIVATELY DEFINED UNIQUE IDENTIFIERS

Privately Defined UIDs are commonly used within DICOM. However, such UIDs will not be registered by NEMA. Organizations that define private UIDs are responsible for properly registering their UIDs (at least obtain a registered <Org Root>) as defined for OSI Object Identifiers (ISO 9834-3). The private organization defining the UID shall accept the responsibility of ensuring its uniqueness.

10 Transfer syntax

A Transfer Syntax is a set of encoding rules able to unambiguously represent one or more Abstract Syntaxes. In particular, it allows communicating Application Entities to negotiate common encoding techniques they both support (e.g., byte ordering, compression, etc.). A Transfer Syntax is an attribute of a

--`,,`,`,--`,,`,`,,`---

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Presentation Context, one or more of which are negotiated at the establishment of an Association between DICOM Application Entities. This Association negotiation is specified in PS 3.8.

The selection of a Transfer Syntax applies to the encoding rules for the Data Set portion of a DICOM Message only. All DICOM Standard and Private Transfer Syntaxes implicitly specify a fixed encoding for the Command Set portion of a DICOM Message as Specified in PS 3.7.

This part of the DICOM Standard defines standard DICOM Transfer Syntaxes and assigns a unique Transfer Syntax Name to each one. The standard DICOM Transfer Syntaxes are specified in Annex A. The DICOM notation for Transfer Syntax names is the notation used for UIDs (see Section 9).

The organization responsible for the definition and registration of DICOM Transfer Syntaxes is NEMA. NEMA guarantees uniqueness for all DICOM Transfer Syntax Names.

Privately defined Transfer Syntax Names may also be used; however, they will not be registered by NEMA. Organizations that define private Transfer Syntax Names shall follow the registration process defined in Section 9.2.

10.1 *DICOM default transfer syntax*

DICOM defines a default Transfer Syntax, the DICOM Implicit VR Little Endian Transfer Syntax (UID = "1.2.840.10008.1.2"), that shall be supported by every conformant DICOM Implementation. This implies that:

a) If an Application Entity issues an A-ASSOCIATE request, it shall offer the DICOM Implicit VR Little Endian Transfer Syntax in at least one of the Presentation Contexts associated with each offered Abstract Syntax.

NOTE— Offering Abstract Syntax (AS1) in two Presentation Contexts with Transfer Syntaxes (TS1) and (TS2) is not valid, but offering AS1-TS1, AS1-TS2 and AS1-TSD is valid because the DICOM Default Little Endian Transfer Syntax (TSD) is present in at least one of the Presentation Contexts which are based on Abstract Syntax (AS1).

b) If an Application Entity receives an A-ASSOCIATE indication corresponding to a request which follows the requirements specified in Section 10.1 a), every Presentation Context related to a given Abstract Syntax cannot be rejected in an A-ASSOCIATE response for the reason that none of the Transfer Syntaxes are supported.

Both of these requirements, a) and b), are waived when the Application Entity sending the pixel data has only access to the pixel data in lossy compressed form.

10.2 *Transfer syntax for a DICOM default of lossless JPEG compression*

DICOM defines a default for lossless JPEG Image Compression, which uses a subset of coding Process 14 with a first-order prediction (Selection Value 1). It is identified by Transfer Syntax UID = "1.2.840.10008.1.2.4.70" and shall be supported by every DICOM implementation that chooses to support one or more of the lossless JPEG compression processes. This implies that:

a) If an Application Entity issues an A-ASSOCIATE request where any offered Abstract Syntaxes is associated in one or more Presentation Context with a JPEG lossless compression Transfer Syntax, at least one of the Presentation Contexts which include this

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Abstract Syntax, shall include the DICOM Default Lossless JPEG Compression Transfer Syntax and the DICOM Default Transfer Syntax (uncompressed).

NOTE— Offering Abstract Syntax (AS1) in two Presentation Contexts with Transfer Syntaxes JPEG lossless (JL1) and (JL2) is not valid, but offering AS1-JL1, AS1-JL2, AS1-TSD, and AS1-JLD is valid because the DICOM Default JPEG Lossless Transfer Syntax (JLD) and the DICOM Default Transfer Syntax (TSD) are present in at least one of the Presentation Contexts which are based on Abstract Syntax (AS1).

b) If an Application Entity that supports one or more lossless JPEG Transfer Syntax receives an A-ASSOCIATE indication corresponding to a request which follows the requirements specified in Section 10.2 a), every Presentation Context related to a given Abstract Syntax cannot be rejected in an A-ASSOCIATE response for the reason that the DICOM Default lossless JPEG Transfer Syntax is not supported.

10.3 *Transfer syntaxes for a DICOM defaults of lossy JPEG compression*

DICOM defines defaults for Lossy JPEG Image Compression, one for 8-bit images and the other for 12-bit images. JPEG coding Process 1 (identified by Transfer Syntax UID = "1.2.840.10008.1.2.4.50") is used for 8-bit images. JPEG coding Process 4 (identified by Transfer Syntax UID = "1.2.840.10008.1.2.4.51") is used for 12-bit images. This implies that:

a) If an Application Entity issues an A-ASSOCIATE request where any offered Abstract Syntaxes is associated in one or more Presentation Context(s) with a JPEG lossy compression Transfer Syntax, at least one of the Presentation Contexts which include this Abstract Syntax, shall include the appropriate DICOM Default Lossy JPEG Compression Transfer Syntax.

NOTES— 1. Offering Abstract Syntax (AS1) in two Presentation Contexts with Transfer Syntaxes JPEG lossy (JL1) and (JL2) is not valid, but offering AS1-JL1, AS1-JL2 and AS1-JLD is valid because the DICOM Default JPEG Lossy Transfer Syntax (JLD) is present in at least one of the Presentation Contexts which are based on Abstract Syntax (AS1).
2. The DICOM Default Transfer Syntax (uncompressed) may be offered if the sender has access to the original pixel data in an uncompressed or lossless compressed form.

b) If an Application Entity that supports one or more Lossy JPEG Transfer Syntaxes receives an A-ASSOCIATE indication corresponding to a request which follows the requirements specified in Section 10.3 a), every Presentation Context related to a given Abstract Syntax cannot be rejected in an A-ASSOCIATE response for the reason that the DICOM Default lossy JPEG Transfer Syntax is not supported.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Annex A
(Normative)
Transfer syntax specifications

A.1 DICOM implicit VR little endian transfer syntax

This Transfer Syntax applies to the encoding of the entire DICOM Data Set. This implies that when a DICOM Data Set is being encoded with the DICOM Implicit VR Little Endian Transfer Syntax the following requirements shall be met:

a) The Data Elements contained in the Data Set structure shall be encoded with Implicit VR (without a VR Field) as specified in Section 7.1.3.

b) The encoding of the overall Data Set structure (Data Element Tags, Value Length, and Value) shall be in Little Endian as specified in Section 7.3 of the DICOM Standard.

c) The encoding of the Data Elements of the Data Set shall be as follows according to their Value Representations:

— For all Value Representations defined in this part, except for the Value Representations OB and OW, the encoding shall be in Little Endian as specified in Section 7.3.

— For the Value Representations OB and OW, the encoding shall meet the following specification depending on the Data Element Tag:

— Data Element (7FE0,0010) Pixel Data has the Value Representation OW and shall be encoded in Little Endian.

— Data Element (60xx,3000) Overlay Data has the Value Representation OW and shall be encoded in Little Endian.

— Data Element (50xx,3000) Curve Data has the Value Representation OB with its component points (n-tuples) having the Value Representation specified in Data Value Representation (50xx,0103). The component points shall be encoded in Little Endian.

This DICOM Implicit VR Little Endian Transfer Syntax shall be identified by a UID of Value "1.2.840.10008.1.2."

A.2 DICOM little endian transfer syntax (explicit VR)

This Transfer Syntax applies to the encoding of the entire DICOM Data Set. This implies that when a DICOM Data Set is being encoded with the DICOM Little Endian Transfer Syntax the following requirements shall be met:

a) The Data Elements contained in the Data Set structure shall be encoded with Explicit VR (with a VR Field) as specified in Section 7.1.2.

b) The encoding of the overall Data Set structure (Data Element Tags, Value Length, and Value) shall be in Little Endian as specified in Section 7.3.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

c) The encoding of the Data Elements of the Data Set shall be as follows according to their Value Representations:

— For all Value Representations defined in this part, except for the Value Representations OB and OW, the encoding shall be in Little Endian as specified in Section 7.3.

— For the Value Representations OB and OW, the encoding shall meet the following specification depending on the Data Element Tag:

— Data Element (7FE0,0010) Pixel Data has the Value Representation OW and shall be encoded in Little Endian.

— Data Element (60xx,3000) Overlay Data has the Value Representation OW and shall be encoded in Little Endian.

— Data Element (50xx,3000) Curve Data has the Value Representation specified in its Explicit VR Field. See the specification of the Curve Data Module in PS 3.3 for the enumerated list of allowable VRs. The component points shall be encoded in Little Endian.

This DICOM Explicit VR Little Endian Transfer Syntax shall be identified by a UID of Value "1.2.840.10008.1.2.1."

A.3 DICOM big endian transfer syntax (explicit VR)

This Transfer Syntax applies to the encoding of the entire DICOM Data Set. This implies that when a DICOM Data Set is being encoded with the DICOM Big Endian Transfer Syntax the following requirements shall be met:

a) The Data Elements contained in the Data Set structure shall be encoded with Explicit VR (with a VR Field) as specified in Section 7.1.2.

b) The encoding of the overall Data Set structure (Data Element Tags, Value Length, and Value) shall be in Big Endian as specified in Section 7.3 of the DICOM Standard.

c) The encoding of the Data Elements of the Data Set shall be as follows according to their Value Representations:

— For all Value Representations defined in this part, except for the Value Representations OB and OW, the encoding shall be in Big Endian as specified in Section 7.3.

— For the Value Representations OB and OW, the encoding shall meet the following specification depending on the Data Element Tag:

— Data Element (7FE0,0010) Pixel Data has the Value Representation OW and shall be encoded in Big Endian.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

— Data Element (60xx,3000) Overlay Data has the Value Representation OW and shall be encoded in Big Endian.

— Data Element (50xx,3000) Curve Data has the Value Representation specified in its Explicit VR Field. See the specification of the Curve Data Module in PS 3.3 for the enumerated list of allowable VRs. The component points shall be encoded in Big Endian.

This DICOM Explicit VR Big Endian Transfer Syntax shall be identified by a UID of Value "1.2.840.10008.1.2.2."

A.4 Transfer syntaxes for encapsulation of encoded pixel data

These Transfer Syntaxes apply to the encoding of the entire DICOM Data Set, even though the image Pixel Data (7FE0,0010) portion of the DICOM Data Set is the only portion that is encoded by an encapsulated format. This implies that when a DICOM Message is being encoded according to an encapsulation Transfer Syntax the following requirements shall be met:

a) The Data Elements contained in the Data Set structure shall be encoded with Explicit VR (with a VR Field) as specified in Section 7.1.2.

b) The encoding of the overall Data Set structure (Data Element Tags, Value Length, etc.) shall be in Little Endian as specified in Section 7.3 of this part.

c) The encoding of the Data Elements of the Data Set shall be as follows according to their Value Representations:

— For all Value Representations defined in this part of the DICOM Standard, except for the Value Representations OB and OW, the encoding shall be in Little Endian as specified in Section 7.3.

— For the Value Representations OB and OW, the encoding shall meet the following specification depending on the Data Element Tag:

— Data Element (7FE0,0010) Pixel Data has the Value Representation OB and is a sequence of bytes resulting from one of the encoding processes. It contains the encoded pixel data stream fragmented into one or more Item(s). This Pixel Data Stream may represent a Single or Multi-frame Image. See Tables A.4-1 and A.4-2:

— The Length of the Data Element (7FE0,0010) shall be set to the Value for Undefined Length (FFFFFFFFH).

— Each Data Stream Fragment encoded according to the specific encoding process shall be encapsulated as a DICOM Item with a specific Data Element Tag of Value (FFFE,E000). The Item Tag is followed by a 4 byte Item Length field encoding the explicit number of bytes of the Item.

— All items containing an encoded fragment shall be made of an even number of bytes greater or equal to two. The last fragment of the DICOM message may be padded, if necessary, to meet fragment format requirements of the DICOM Standard.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

— The first Item in the Sequence of Items before the encoded Pixel Data Stream shall be a Basic Offset Table item. The Basic Offset Table Item Value, however, is not required to be present:

  — When the Item Value is not present, the Item Length shall be zero (00000000H) (see Table A.4-1).

  — When the Item Value is present, the Basic Offset Table Item Value shall contain concatenated 32-bit unsigned integer values that are byte offsets to the first byte of the Item Tag of the first fragment for each frame in the Sequence of Items. These offsets are measured from the first byte of the first Item Tag following the Basic Offset Table item (See Table A.4-2).

  NOTE— For a Multi-Frame Image containing only one frame or a Single Frame Image, the Basic Offset Table Item Value may be present or not. If present it will contain a single 00000000H value.

— This Sequence of Items is terminated by a Sequence Delimiter Item with the Tag (FFFE,E0DD) and an Item Length Field of Value (00000000H) (i.e., no Value Field shall be present).

— Data Element (60xx,3000) pertaining to Overlay Data has the Value Representation OW and shall be encoded in Little Endian.

— Data Element (50xx,3000) for Curve Data has the Value Representation specified in its Explicit VR Field. See the specification of the Curve Data Module in PS 3.3 for the enumerated list of allowable VRs. The component points shall be encoded in Little Endian.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Provided by IHS
No reproduction or networking permitted without license from IHS

Table A.4-1—Example for elements of an encoded single-frame image defined as a sequence of three fragments without basic offset table item value

Pixel Data Element Tag	Data Element Length	Data Element													
		Basic Offset Table with NO Item Value		First Fragment (Single-Frame) of Pixel Data			Second Fragment (Single-Frame) of Pixel Data			Third Fragment (Single-Frame) of Pixel Data			Sequence Delimiter Item	
(7FE0, 0010) with VR of OB	FFFF FFFFH un-defined length	Item Tag	Item Length	Item Tag	Item Length	Item Value	Item Tag	Item Length	Item Value	Item Tag	Item Length	Item Value	Sequence Delim. Tag	Item Length
		(FFFE, E000)	0000 0000H	(FFFE, E000)	0000 04C6H	com-pressed frag-ment	(FFFE, E000)	0000 024AH	com-pressed frag-ment	(FFFE, E000)	0000 0628H	com-pressed frag-ment	(FFFE, E0DD)	0000 0000H
4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	04C6H bytes	4 bytes	4 bytes	024AH bytes	4 bytes	4 bytes	0628H bytes	4 bytes	4 bytes

Table A.4-2—Examples of elements for an encoded two-frame image defined as a sequence of three fragments without basic table item values

Pixel Data Ele-ment Tag	Data Ele-ment Length	Data Element													
		Basic Offset Table with Item Value			First Fragment (Frame1) of Pixel Data			Second Fragment (Frame 2) of Pixel Data			Third Fragment (Frame 3) of Pixel Data			Sequence Delimiter Item	
(7FE0, 0010) with VR of OB	FFFF FFFFH un-defined length	Item Tag	Item Length	Item Value	Item Tag	Item Length	Item Value	Item Tag	Item Length	Item Value	Item Tag	Item Length	Item Value	Sequence Delim. Tag	Item Length
		(FFFE, E000)	0000 0008H	0000 0000H 0000 0646H	(FFFE, E000)	0000 02C8H	com-pressed frag-ment	(FFFE, E000)	0000 036EH	com-pressed frag-ment	(FFFE E000)	0000 0BC8H	com-pressed frag-ment	(FFFE E0DD)	0000 0000H
4 bytes	4 bytes	4 bytes	8 bytes	0008H bytes	4 bytes	4 bytes	02C8H bytes	4 bytes	4 bytes	036EH bytes	4 bytes	4 bytes	0BC8H bytes	4 bytes	4 bytes

Not for Resale

A.4.1 *JPEG image compression*

The International Standards Organization ISO/IEC JTC1 has developed an International Standard, ISO/IS-10918-1 (JPEG Part 1) and an International Draft Standard, ISO/IS-10918-2 (JPEG Part 2), known as the JPEG Standard, for digital compression and coding of continuous-tone still images. (See Annex F for further details.)

A DICOM Transfer Syntax for JPEG Image Compression shall be identified by a UID value, appropriate to its JPEG coding process, chosen from Table A.4-3.

Table A.4-3—DICOM transfer syntax UIDs for JPEG

DICOM Transfer Syntax UID	JPEG coding process	JPEG description
1.2.840.10008.1.2.4.50	1	baseline
1.2.840.10008.1.2.4.51	2(8-bit),4(12-bit)	extended
1.2.840.10008.1.2.4.52	3(8-bit),5(12-bit)	extended
1.2.840.10008.1.2.4.53	6(8-bit),8(12-bit)	spectral selection, non-hierarchical
1.2.840.10008.1.2.4.54	7(8-bit),9(12-bit)	spectral selection, non-hierarchical
1.2.840.10008.1.2.4.55	10(8-bit),12(12-bit)	full progression, non-hierarchical
1.2.840.10008.1.2.4.56	11(8-bit),13(12-bit)	full progression, non-hierarchical
1.2.840.10008.1.2.4.57	14	lossless, non-hierarchical
1.2.840.10008.1.2.4.58	15	lossless, non-hierarchical
1.2.840.10008.1.2.4.59	16(8-bit),18(12-bit)	extended, hierarchical
1.2.840.10008.1.2.4.60	17(8-bit),19(12-bit)	extended, hierarchical
1.2.840.10008.1.2.4.61	20(8-bit),22(12-bit)	spectral selection, hierarchical
1.2.840.10008.1.2.4.62	21(8-bit),23(12-bit)	spectral selection, hierarchical
1.2.840.10008.1.2.4.63	24(8-bit),26(12-bit)	full progression, hierarchical
1.2.840.10008.1.2.4.64	25(8-bit),27(12-bit)	full progression, hierarchical
1.2.840.10008.1.2.4.65	28	lossless, hierarchical
1.2.840.10008.1.2.4.66	29	lossless, hierarchical
1.2.840.10008.1.2.4.70	14 (Selection Value 1)	lossless, hierarchical, first-order prediction

NOTE— DICOM identifies, to increase the likelihood of successful association, three Transfer Syntaxes for Default JPEG Compression Image processes (see Sections 8.2.1 and 10).

If the object allows multi-frame images in the pixel data field, then each frame shall be encoded separately. Each fragment shall contain encoded data from a single-frame image.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

For single-frame images or for the first frame of a multi-frame image, the JPEG Interchange Format shall be used (the table specification shall be included). For any frame of a multi-frame image, other than the first frame, the JPEG Abbreviated Format for compressed image data may be used (the table specifications are not included).

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

© NEMA PS 3.5-1993

Annex B
(Informative)
Creating a privately defined unique identifier

Privately defined Unique Identifiers (UIDs) are used in DICOM to uniquely identify items such as Specialized or Private SOP Classes, Image SOP Instances, Study SOP Instances, etc.

A UID is formed using a registered root (see Annex C) and an organization specific suffix. The manner in which the suffix of a Privately Defined UID is defined is not constrained by the DICOM Standard. Only the guarantee of its uniqueness by the defining organization is required by DICOM. This example presents a particular choice made by a specific organization in defining its suffix to guarantee uniqueness. A variant is discussed.

"1.2.840.xxxxx.3.152.235.2.12.187636473"

root. suffix

In this example, the root is:

1	Identifies ISO
2	Identifies ANSI Member Body
840	Country code of a specific Member Body (U.S. for ANSI)
xxxxx	Identifies a specific Organization.(provided by ANSI)

In this example the first two components of the suffix relate to the identification of the device:

3	Manufacturer or user defined device type
152	Manufacturer or user defined serial number

The remaining four components of the suffix relate to the identification of the image:

235	Study number
2	Series number
12	Image number
187636473	Encoded date and time stamp of image acquisition

In this example, the organization has chosen these components to guarantee uniqueness. Other organizations may choose an entirely different series of components to uniquely identify its images. In this example, the organization has chosen these components to guarantee uniqueness. Other organizations may choose an entirely different series of components to uniquely identify its images. For example it may have been perfectly valid to omit the Study Number, Series Number and Image Number if the time stamp had a sufficient precision to ensure that no two images might have the same date and time stamp.

--`,,`,,`-`-`,,`,,`,`,,`---

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Because of the flexibility allowed by the DICOM Standard in creating Privately Defined UIDs, implementations should not depend on any assumed structure of UIDs and attempt to parse UIDs to extract the semantics of some of its components.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

Annex C
(Informative)
DICOM unique identifier registration process

This registration process applies to a number of unique identifiers that share the same properties, structure and registration process. It applies to the following identifiers:

— The Values assigned to DICOM Data Elements of Value Representation VR = UID (See Table 6.2-1). Such Data Elements are defined in PS 3.3, PS 3.4, PS 3.6, and PS 3.7.

— The DICOM Abstract Syntaxes Names. Abstract Syntax Names are defined in PS 3.4.

— The DICOM Transfer Syntax Names. Transfer Syntax Names are defined in Annex A.

— The DICOM Application Context Names. Application Context Names are defined in PS 3.7

UID structure is based on the numeric form of the OSI Object Identifier as defined by ISO 8824. Values shall be registered as defined by ISO 9834-3 to ensure global uniqueness.

The DICOM Standard assigns Values to a number of such unique identifiers. The organization responsible for their registration is NEMA which guarantees uniqueness.

For privately registered identifiers, NEMA will not act as registration authority. Related organizations shall obtain their proper registration as defined for OSI Object Identifiers by ISO 9834-3 to ensure global uniqueness. National Standards Organizations representing a number of countries (e.g., UK, France, Japan, USA, etc.) for the International Standards Organization act as a registration authority by delegation from ISO, as defined by ISO 9834-3.

NOTES— 1. For example, in the USA ANSI assigns, for a fee, Organization Identifiers to any requesting organization. Such an identifier may be used by the identified organization as a root to which it may add a suffix made of one or more components. The identified organization accepts the responsibility to properly register these suffixes to ensure uniqueness.

2. Following are two typical examples of obtaining a UID <org root>. These examples are not intended to illustrate all the possible methods for obtaining a UID <org root>, see ISO 8824 and ISO 9843-3 for complete specifications. Organization identifiers may be obtained from various ISO member bodies (e.g., IBN in Belgium, ANSI in the United States, AFNOR in France, BSI in Great Britain, DIN in Germany, COSIRA in Canada).

   The first example shows the case of an <org root> issued by an ISO Member Body (ANSI in the USA in this example). The <org root> is composed of an identifier for ISO, a member body branch identifier, a country code and an organization ID.

   NOTE— There is no requirement that an implementation using an ANSI issued <org root> be made or located in the USA.

The <org root> is made up of the following components: 1.2.840.xxxxx

— 1 identifies ISO

— 2 identifies the ISO member body branch

— 840 identifies the country code of a specific ISO member body (U.S. for ANSI)

— xxxxx identifies a specific organization as registered by the ISO member body ANSI.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

The second example shows the case of an <org root> issued by ISO (is delegated to BSI) to an international organization. It is composed of an identifier for ISO, an international organization branch identifier, and an International Code Designator. The value of the<org root> is assigned by an international registration authority that may be used by many different UID's defined by the same international organization.

The <org root> is made up of the following components: 1.3.yyyy

— 1 identifies ISO

— 3 identifies the international organization branch

— yyyy identifies a specific organization as registered by an International Code Designator registration authority (see ISO 6523).

3. Example components of a <suffix> for unique identification of an image may include:

— product

— system identifier

— study, series and image numbers

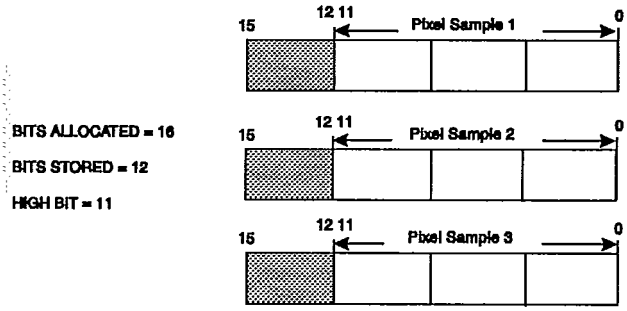
— study, series and image date & times.

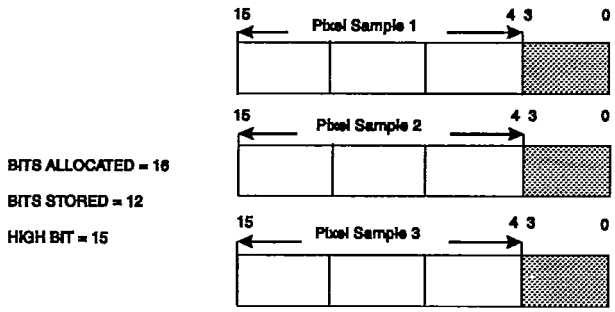
Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

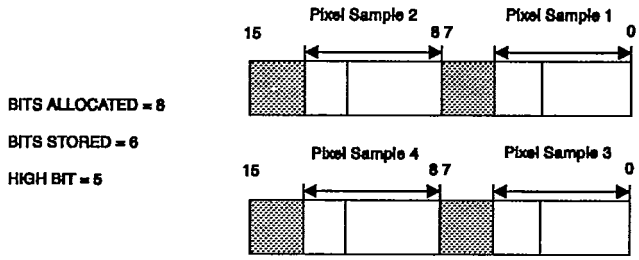
© NEMA PS 3.5-1993

Annex D
(Informative)
Examples of various pixel data and overlay encoding schemes

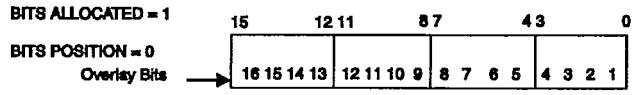
The following examples further illustrate the use of the data elements for Bits Allocated (0028,0100), Bits Stored (0028,0101), and High Bit (0028,0102).



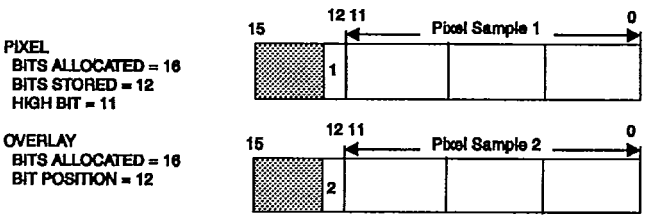




An Example of an encoded Overlay



An example of encoded pixel data with an embedded overlay



Provided by IHS
No reproduction or networking permitted without license from IHS
Not for Resale

© NEMA PS 3.5-1993

Annex E
(Normative)
DICOM default character repertoire

The default repertoire for character strings in DICOM is the Basic G0 Set of the International Reference Version of ISO 646:1990 (ISO IR-6). In addition, the three control characters LF, FF, and CR are supported. These control characters are a subset of the C0 set defined in ISO 646:1990 and ISO 6429:1990.

The byte encoding of the default character repertoire is pictured in Table E-1. This table can be used to derive both ISO column/row byte values and hex values for encoded representations (see Section 6.1.1).

Table E-1—DICOM Default character repertoire encoding

				b_8	0	0	0	0	0	0	0	0
				b_7	0	0	0	0	1	1	1	1
				b_6	0	0	1	1	0	0	1	1
				b_5	0	1	0	1	0	1	0	1
b_4	b_3	b_2	b_1		00	01	02	03	04	05	06	07
0	0	0	0	00			SP	0	@	P	`	p
0	0	0	1	01			!	1	A	Q	a	q
0	0	1	0	02			"	2	B	R	b	r
0	0	1	1	03			#	3	C	S	c	s
0	1	0	0	04			$	4	D	T	d	t
0	1	0	1	05			%	5	E	U	e	u
0	1	1	0	06			&	6	F	V	f	v
0	1	1	1	07			'	7	G	W	g	w
1	0	0	0	08			(	8	H	X	h	x
1	0	0	1	09			)	9	I	Y	i	y
1	0	1	0	10	LF		*	:	J	Z	j	z
1	0	1	1	11			+	;	K	[	k	{
1	1	0	0	12	FF		,	<	L	\	l	|
1	1	0	1	13	CR		-	=	M	]	m	}
1	1	1	0	14			.	>	N	^	n	~
1	1	1	1	15			/	?	O	_	o	

Provided by IHS
No reproduction or networking permitted without license from IHS
Not for Resale

Annex F
(Informative)
Encapsulated images as part of a DICOM message

The following remarks generally apply to communicating an encoded image within a message structure according to the DICOM Standard:

a) In the course of including an encoded image in a DICOM message, the encoding is not changed. The encoded data stream is merely segmented and encapsulated according to the protocols of the DICOM Standard. After unpacking the DICOM message, the encoded data stream can be fully reconstructed at the receiving node.

b) The object definition of the DICOM Standard is always determining format and other choices that a specific encoding implementation may offer. The encoded image must be consistent with the definition of the object of which the encoded image is part. For example:

   1) If the object is defined to contain 10-bit pixel data, it is assumed that the encoding process is one that accepts at least 10-bit data. Hence, there is no need for defining separate Transfer Syntaxes, e.g. for 8-bit or 12-bit implementations. Any 12-bit implementation is assumed to operate in an 8-bit process if the object is defined to contain 8-bit data.

   2) If the image of an object is interleaved, the encoding process must reproduce the interleaving.

c) Specifications in the encoding file header must be consistent with the DICOM Message header, e.g. regarding the number of rows and columns.

d) The byte order specification of an encoded file is not altered in the course of encapsulating it in a DICOM message.

F.1 Encapsulated JPEG encoded images

The International Standards Organization (ISO/IEC JTC1/SC2/WG10) has prepared an International Standard, ISO/IS-10918-1 (JPEG Part 1) and International Draft Standard ISO/IS-10918-2 (JPEG Part 2), for the digital compression and coding of continuous-tone still images. This standard is collectively known as the JPEG Standard.

Part 1 of the JPEG Standard sets out requirements and implementation guidelines for the coded representation of compressed image data to be interchanged between applications. The processes and representations are intended to be generic in order to support the broad range of applications for color and grayscale still images for the purpose of communications and storage within computer systems. Part 2 of the JPEG Standard defines tests for determining whether implementations comply with the requirements of the various encoding and decoding processes specified in Part 1 of the JPEG Standard.

The JPEG Standard specifies lossy and lossless code processes. The lossy coding is based on the discrete cosine transform (DCT), permitting data compression with an adjustable compression ratio. The lossless coding employs differential pulse code modulation (DPCM).

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

The JPEG Standard permits a variety of coding processes for the coder and decoder. These processes differ in coding schemes for the quantified data and in sample precision. The coding processes are consecutively numbered as defined in the International Draft Standard ISO/IS-10918-2 (JPEG Part 2), and are summarized in Table F.1-1. The simplest DCT-based coding process is referred to as Baseline Sequential with Huffman Coding for 8-bit Samples.

Table F.1-1—JPEG modes of image coding

No.	Description	Lossy LY Lossless LL	Non-Hierarchy NH Hierarchical H	Sequential S Progressive P	Transform	Coding	Accepted Bits
1	Baseline	LY	NH	S	DCT	Huffman	8
2	Extended	LY	NH	S	DCT	Huffman	8
3	Extended	LY	NH	S	DCT	Arithmetic	8
4	Extended	LY	NH	S	DCT	Huffman	12
5	Extended	LY	NH	S	DCT	Arithmetic	12
6	Spectral selection only	LY	NH	P	DCT	Huffman	8
7	Spectral selection only	LY	NH	P	DCT	Arithmetic	8
8	Spectral selection only	LY	NH	P	DCT	Huffman	12
9	Spectral selection only	LY	NH	P	DCT	Arithmetic	12
10	Full progression	LY	NH	P	DCT	Huffman	8
11	Full progression	LY	NH	P	DCT	Arithmetic	8
12	Full progression	LY	NH	P	DCT	Huffman	12
13	Full progression	LY	NH	P	DCT	Arithmetic	12
14	Lossless	LL	NH	S	DPCM	Huffman	2-16
15	Lossless	LL	NH	S	DPCM	Arithmetic	2-16
16	Extended sequential	LY	H	S	DCT	Huffman	8
	Extended sequential	LY	H	S	DCT	Arithmetic	8
17	Extended sequential	LY	H	S	DCT	Huffman	12
18	Extended sequential	LY	H	S	DCT	Arithmetic	12
19							
20	Spectral selection only	LY	H	P	DCT	Huffman	8
21	Spectral selection only	LY	H	P	DCT	Arithmetic	8
22	Spectral selection only	LY	H	P	DCT	Huffman	12
23	Spectral selection only	LY	H	P	DCT	Arithmetic	12
24	Full progression	LY	H	P	DCT	Huffman	8
25	Full progression	LY	H		DCT	Arithmetic	8
26	Full progression	LY	H	P	DCT	Huffman	12
27	Full progression	LY	H	P	DCT	Arithmetic	12
				P			
28	Lossless	LL	H	S	DPCM	Huffman	2-16
29	Lossless	LL	H	S	DPCM	Arithmetic	2-16

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

The different coding processes specified in the JPEG Standard are closely related. By extending the capability of an implementation, increasingly more 'lower level' processes can also be executed by the implementation. This is shown in Tables F.1-2 and F.1-3 for Huffman and Arithmetic Coding, respectively. Table F.1-4 presents the capabilities for lossless implementations.

It is worth recognizing that implementations using arithmetic coding have the capability of performing Huffman coded operations with two Huffman tables since they must be able to execute the baseline process. Hence, by increasing the capability of operating with two additional Huffman tables, the odd-numbered coding processes with arithmetic coding (Table F.1-1) can also execute all corresponding even-numbered processes with Huffman coding (Table F.1-5).

Inclusion of a JPEG-coded image in a DICOM message is facilitated by the use of specific Transfer Syntaxes which are defined in Annex A. Independent of the JPEG coding processes, the same syntax applies. The only distinction for different processes in the syntax is the UID value. Table F.1-5 lists the UID values in the Transfer Syntax for the various JPEG coding processes for reference.

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

© NEMA PS 3.5-1993

Table F.1-2—Relationship between the lossy JPEG huffman coding processes

* Coding process of column can execute coding process of row

Process	1	2	4	6	8	10	12	16	18	20	22	24	26
1	*	*	*	*	*	*	*	*	*	*	*	*	*
2		*	*	*	*	*	*	*	*	*	*	*	*
4			*		*		*		*		*		*
6				*	*	*	*			*	*	*	*
8					*		*				*		*
10						*	*					*	*
12							*						*
16								*	*	*	*	*	*
18									*		*		*
20										*	*	*	*
22											*		*
24												*	*
26													*

Provided by IHS
No reproduction or networking permitted without license from IHS
Not for Resale

© NEMA PS 3.5-1993

Table F.1-3—Relationship between the lossy JPEG arithmetic coding processes

* Coding process of column can execute coding process of row

Process	1	3	5	7	9	11	13	17	19	21	23	25	27
1+	*	*	*	*	*	*	*	*	*	*	*	*	*
3		*	*	*	*	*	*	*	*	*	*	*	*
5			*		*		*		*		*		*
7				*	*	*	*			*	*	*	*
9					*		*				*		*
11						*	*					*	*
13							*						*
17								*	*	*	*	*	*
19									*		*		*
21										*	*	*	*
23											*		*
25												*	*
27													*

\+ The Baseline Coding Process, which uses Huffman encoding, is required by all lossy coding processes.

Table F.1-4—Relationship between the lossless JPEG processes

* Coding process of column can execute coding process of row

Process	14	15	28	29
14	*		*	
15		*		*
28			*	
29				*

Provided by IHS
No reproduction or networking permitted without license from IHS
Not for Resale

Table F.1-5—Identification of JPEG coding processes in DICOM

DICOM Transfer Syntax UID	JPEG process	JPEG description	capable of performing	
				lossy arithmetic coding w/opt. 4 Huffman tables
1.2.840.10008.1.2.4.50	1	baseline	1	
1.2.840.10008.1.2.4.51	2,4	extended	1,2,4	
1.2.840.10008.1.2.4.52	3,5	extended	1,3,5	ALL ≤ 5
1.2.840.10008.1.2.4.53	6,8	spect. select. NH	1,2,4,6,8	
1.2.840.10008.1.2.4.54	7,9	spect. select. NH	1,3,5,7,9	ALL ≤ 9
1.2.840.10008.1.2.4.55	10,12	full progression NH	1,2,4,6,8,10,12	
1.2.840.10008.1.2.4.56	11,13	full progression NH	1,3,5,7,9,11,13	ALL ≤ 13
1.2.840.10008.1.2.4.57	14	lossless NH	14	
1.2.840.10008.1.2.4.58	15	lossless NH	15	
1.2.840.10008.1.2.4.59	16,18	extended H	1,2,4,16,18	
1.2.840.10008.1.2.4.60	17,19	extended H	1,3,5,17,19	1,2,3,4,5,16,17,18,19
1.2.840.10008.1.2.4.61	20,22	spect. select.H	1,2,4,6,8,16,18,20,22	
1.2.840.10008.1.2.4.62	21,23	spect. select. H	1,3,5,7,9,17,19,21,23	1,2,3,4,5,6,7,8,9,16,17,18,19,20,21,22,23
1.2.840.10008.1.2.4.63	24,26	full progression H	1,2,4,6,8,10,12,16,18, 20,22,24,26	
1.2.840.10008.1.2.4.64	25,27	full progression H	1,3,5,7,9,11,13,17,19, 21,23,25,27	ALL ≤ 27, EXCEPT 14,15
1.2.840.10008.1.2.4.65	28	lossless H	14,28	
1.2.840.10008.1.2.4.66	29	losses H	15,29	
1.2.840.10008.1.2.3.70	14 Selection Value 1	lossless NH, first-order prediction		

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

© NEMA PS 3.5-1993

Annex G
(Informative)
Index to Data Element Tags

Nomenclature **Document Pages**

(0000,eeee) 21
(0001,eeee) 21
(0002,eeee) 21
(0003,eeee) 21
(0004,eeee) 21
(0005,eeee) 21
(0006,eeee) 21
(0007,eeee) 21
(0008,0005) 13, 17-19
(0018,0020) 27
(0018,0082) 27
(0018,00FF) 16
(0028,0100) 32, 33, 49
(0028,0101) 32, 49
(0028,0102) 32, 49
(0028,0103) 32
(0028,0106) 32
(0028,0107) 32
(4008,0212) 15
(5000-501E,eeee) 30
(50xx,0103) 38
(50xx,3000) 38-41
(50xx,eeee) 9, 30
(6000-601E,eeee) 30
(60xx,3000) 38-41
(60xx,eeee) 9, 30
(7FE0,0010) 32, 33, 38-40
(FFFE,E000) 27, 40
(FFFE,E00D) 27, 28
(FFFE,E0DD) 27, 28, 41
(FFFF,eeee) 21, 28
(gggg,0000) 25
(gggg,0010) 31
(gggg,0010-00FF) 31
(gggg,0011) 31
(gggg,00FF) 31
(gggg,1000-10FF) 31
(gggg,10FF-7FFF) 31
(gggg,1100-11FF) 31
(gggg,eeee) 11
(gggg,FF00-FFFF) 31

Provided by IHS
No reproduction or networking permitted without license from IHS
Not for Resale

Annex H
(Informative)
Index to Transfer Syntax UIDs

Nomenclature	Document Pages
1.2.840.10008.1.2	36-40, 43, 56
1.2.840.10008.1.2.1	39
1.2.840.10008.1.2.2	40
1.2.840.10008.1.2.4.50	37, 43, 56
1.2.840.10008.1.2.4.51	43, 56
1.2.840.10008.1.2.4.52	43, 56
1.2.840.10008.1.2.4.53	43, 56
1.2.840.10008.1.2.4.54	43, 56
1.2.840.10008.1.2.4.55	43, 56
1.2.840.10008.1.2.4.56	43, 56
1.2.840.10008.1.2.4.57	43, 56
1.2.840.10008.1.2.4.58	43, 56
1.2.840.10008.1.2.4.59	43, 56
1.2.840.10008.1.2.4.60	43, 56
1.2.840.10008.1.2.4.61	43, 56
1.2.840.10008.1.2.4.62	43, 56
1.2.840.10008.1.2.4.63	43, 56
1.2.840.10008.1.2.4.64	43, 56
1.2.840.10008.1.2.4.65	43, 56
1.2.840.10008.1.2.4.66	43, 56
1.2.840.10008.1.2.4.70	36, 43

Provided by IHS
No reproduction or networking permitted without license from IHS Not for Resale

NEMA STANDARDIZATION BACKGROUND

The purpose of NEMA standards, their classification, and status are set forth in certain clauses of the NEMA *Standardization Policies and Procedures* manual and are referenced below:

Purpose of Standards

National Electrical Manufacturers Association Standards are adopted in the public interest and are designed to eliminate misunderstandings between the manufacturer and the purchaser and to assist the purchaser in selecting and obtaining the proper product for their particular needs. Existence of a National Electrical Manufacturers Association Standard does not in any respect preclude any member or nonmember from manufacturing or selling products not conforming to the standard.

Definition of a Standard

A standard of the National Electrical Manufacturers Association defines a product, process, or procedure with reference to one or more of the following: nomenclature, composition, dimensions, tolerances, safety, operating characteristics, performance, rating, testing, and the service for which they are designed.

(Standardization Policies and Procedures, p. 1)

Dimensions

Where dimensions are given for interchangeability purposes, alternate dimensions satisfying the other provisions of the Standards Publication may be capable of otherwise equivalent performance.

(Standardization Policies and Procedures, p. 6)

Categories of Standards

National Electrical Manufacturers Association Standards are of three classes, which have received the affirmative vote of at least two-thirds of the Subdivision votes cast in the affirmative or negative:

1. NEMA Standard, which relates to a product, process, or procedure commercially standardized and subject to repetitive manufacture.
2. Suggested Standard for Future Design, which may not have been regularly applied to a commercial product, but which suggests a sound engineering approach to future development.
3. Adoptive Standard, which is adopted in whole or in part from the standards of another organization, either domestic, regional, or international.

(Standardization Policies and Procedures, pp. 5)

Authorized Engineering Information

Authorized Engineering Information consists of explanatory data and other engineering information of an informative character not falling within the classification of NEMA Standard or Suggested Standard for Future Design, which standard has received the affirmative vote of at least two-thirds of the Subdivision votes cast in the affirmative or negative.

(Standardization Policies and Procedures, pp. 5)

Identification of Status

Standards in NEMA Standards Publications are identified as "NEMA Standard," "Suggested Standard for Future Design," or "Adoptive Standard." These indicate the status of the standard. A statement incorporating the auxiliary verb "shall" indicates that compliance with a requirement is mandatory for compliance with the standard. These classes of standards are identified in the foreword or throughout the text.

The material identified as "Authorized Engineering Information" is designated similarly. Statements incorporating other auxiliary verbs such as "should," "may," etc. refer to the authorized engineering information and not to procedures required for compliance with the standard.

Provided by IHS
No reproduction or networking permitted without license from IHS
Not for Resale

MEDICAL PICTURE ARCHIVING & COMMUNICATION SYSTEM SECTION OF THE NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION

MEMBER COMPANIES

AGFA Division of Miles Inc
Orangeburg, NY 10962

Camtronics Ltd Medical Systems
Hartland, WI 53029

CEMAX Inc
Fremont, CA 94538

Dejarnette Research Systems
Towson, MD 21204

E I Dupont Denemours & Company Inc
Wilmington, DE 19880-0708

Eastman Kodak Company
Rochester, NY 14653-9014

Eigen
Nevada City, CA 95959

E-Systems Inc
Melpar Division
Falls Church, VA 22046-1571

GE Medical Systems
Milwaukee, WI 53201

InfiMed, Inc.
Liverpool, NY 13088

Merge Technologies Inc
Milwaukee, WI 53214-3151

Oldelft Corporation of America
Fairfax, VA 22031

Philips Medical Systems,
North America Company
Shelton, CT 06484

Picker International, Inc.
Highland Heights, OH 44143

Polaroid Corporation
Cambridge, MA 02139

Siemens Medical Systems, Inc
Iselin, NJ 08830

Siemens Gammasonics, Inc
Hoffman Estates, IL 60195-7372

SMS
Malvern, PA 19355

Sony Medical Systems Division
Park Ridge, NJ 07656

3M
St. Paul, MN 55144

Toshiba America Medical System Inc
Tustiin, CA 92680-2068

Vortech Data, Inc
Richardson, TX 75080

Provided by IHS
No reproduction or networking permitted without license from IHS
Not for Resale