

DICOM Correction Proposal

STATUS	Letter Ballot
Date of Last Update	2024-05-29
Person Assigned	Clunie
Submitter Name	Jörg Riesmeier <dicom@jriesmeier.com>
Submission Date	2023-08-25

Correction Number	CP-2337
Log Summary: Implementation Version Name will contain version information	
Name of Standard PS3.7	
Rationale for Correction: DICOM PS3.10 makes clear that the optional Implementation Version Name (0002,0013) Attribute “Identifies a version for an Implementation Class UID (0002,0012)” and references PS3.7 for details. However, PS3.7 is missing a clear statement that version information shall be present in the Implementation Version Name, i.e. the same Implementation Version Name shall not be used for different values of Implementation Class UID. The reverse is allowed (see first Note in PS3.7 Section D.3.3.2). The term “Version” in Implementation Version Name already suggests that version information is expected to be present, but some vendors seem to only encode the name of the product (i.e. without version information) in Implementation Version Name, and use it for different implementations (i.e. values of Implementation Class UID). Therefore, a clarification is needed.	
Correction Wording:	

Change PS3.7 Section D.3.3.2

D.3.3.2 Implementation Identification Notification

The implementation identification notification allows implementations of communicating AEs to identify each other at Association establishment time. It is intended to provide respective (each network node knows the other's implementation identity) and non-ambiguous identification in the event of communication problems encountered between two nodes. This negotiation is required.

Implementation identification relies on two pieces of information:

- Implementation Class UID (required)
- Implementation Version Name (optional)

The Implementation Class UID identifies in a unique manner a specific class of implementation. Each node claiming **e**Conformance to this Standard shall be assigned an Implementation Class UID to distinguish its implementation environment from others. Such Implementation Class UIDs shall be registered by the implementing organization per the policies defined in PS3.5. This Standard does not specify the policies associated with assigning such a UID.

Different equipment of the same type or product line (but having different serial numbers) shall use the same Implementation Class UID if they share the same implementation environment (i.e., software).

The notification by Association requestors and acceptors of their respective Implementation Class UID is required for all implementations conforming to this Standard. Figure D.3-3 illustrates the Implementation Class UID notification.

[diagram removed from this document]

Figure D.3-3. Implementation Class UID Notification

In addition to the Implementation Class UID, an option is provided to convey an Implementation Version Name of up to 16 characters, **which identifies a version of an implementation for an Implementation Class UID**. Figure D.3-4 illustrates the Implementation Version Name notification. This Standard does not specify the structure and policies associated with such an Implementation Version Name. The absence of the Implementation Version Name requires that the use of the same Implementation Class UID by two nodes guarantees that these use the same version of implementation.

Note

1. As the UID shall not be parsed (their structure is not intended to convey any semantic significance beyond uniqueness), this optional Implementation Version Name provides an adequate mechanism to distinguish two versions of the same implementation (same Implementation Class UID).
2. **It is expected that if the Implementation Version Name is present, it will not have the same value when there are different values of the Implementation Class UID. For example, version information may be included in the Implementation Version Name in addition to the name of the implementation.**

For reference see PS3.10 Section 7.1

7.1 DICOM File Meta Information

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

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

Note

1. If the File Preamble is not used by an Application Profile or a specific implementation, all 128 bytes shall be set to 00H. This is intended to facilitate the recognition that the Preamble is used when all 128 bytes are not set as specified above.
2. The File Preamble may for example contain information enabling a multi-media application to randomly access images stored in a DICOM Data Set. The same file can be accessed in two ways: by a multi-media application using the preamble and by a DICOM Application that ignores the preamble.

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

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

Table 7.1-1. DICOM File Meta Information

Attribute Name	Tag	Type	Attribute Description
[...]			
Implementation Class UID	(0002,0012)	1	Uniquely identifies the implementation that wrote this file and its content. It provides an unambiguous identification of the type of implementation that last wrote the file in the event of interchange problems. It follows the same policies as defined by PS3.7 (association negotiation).

Attribute Name	Tag	Type	Attribute Description
Implementation Version Name	(0002,0013)	3	Identifies a version for an Implementation Class UID (0002,0012) using up to 16 characters of the ISO 646:1990 (basic G0 set) repertoire. It follows the same policies as defined by PS3.7 (association negotiation).
[...]			